



Silk Test 19.5

Silk Test Workbench ヘルプ

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

Copyright © Micro Focus 1992-2018. All rights reserved.

MICRO FOCUS, Micro Focus ロゴ及び Silk Test は Micro Focus IP Development Limited
またはその米国、英国、その他の国に存在する子会社・関連会社の商標または登録商標です。

その他、記載の各名称は、各所有社の知的所有財産です。

2018-10-22

目次

Silk Test Workbench 19.5 へようこそ	23
ライセンス情報	25
はじめに	26
ログオン	26
ログオン パスワードの変更	26
Silk Test Workbench の概要	27
Silk Test Workbench を使用する利点	27
特殊な状況下での自動化 (周辺機器が無い)	28
Silk Test 製品スイート	29
Silk Test Workbench の機能	30
Silk Test Workbench の UI	31
Silk Test Workbench ソフトウェア コンポーネント	32
自動テストについて	32
テスト戦略	32
別の場所にあるデータベースの利用について	33
Silk Test Workbench ドキュメント	34
Silk Test Workbench の新機能	34
UI オートメーション サポート	34
Silk Test Workbench での資産の比較	34
ユーザビリティの改善	35
使用技術の更新	35
プロジェクトの管理	36
プロジェクトの表示	37
ショートカット キー	37
チュートリアル	39
Silk Test Workbench ビジュアル テスト チュートリアル	40
Silk Test Workbench スクリプト チュートリアル	69
ヘルプについてのヘルプ	77
ヘルプを表示する方法	77
ヘルプ トピックの印刷	78
使用状況データの収集の有効化/無効化	78
Micro Focus へのお問い合わせ	78
Micro Focus SupportLine が必要とする情報	79
テストの作成	80
ビジュアル テストの作成	80
ビジュアル テストの記録	80
開始画面からのビジュアル テストの記録	82
複数のテスト アプリケーションをテストするビジュアル テストの記録	83
ビジュアル テストでのロケータからオブジェクト マップ エントリへの移動	84
Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するビジュアル テ	85
新しいブラウザ ウィンドウで開くビジュアル テストの構成	86
ビジュアル テスト内での特定のステップの表示	87
テスト アプリケーションでのビジュアル テスト用コントロールの識別	87
画面プレビューでのビジュアル テスト用コントロールの識別	88
オブジェクトの識別ダイアログを使用したビジュアル テスト用コントロールの識別	88
特定のテスト ステップでの追加操作の記録	89
テスト アプリケーションからビジュアル テストへの画面の挿入	89

ビジュアルテストのオブジェクト マップ項目またはロケーターの手動による記録	90
既存のビジュアルテストを開く	91
ビジュアルテストの保存	91
ビジュアルテストの概要	91
プロパティ	100
スクリプトの作成	110
スクリプトを使用する利点	111
テスト スクリプト作成のベスト プラクティス	111
スクリプトの構文	112
ドライバ スクリプト	113
手動によるスクリプト作成	114
スクリプトの記録	115
複数のテスト アプリケーションをテストするスクリプトの記録	117
Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するスクリプトの	117
新しいブラウザ ウィンドウで開くスクリプトの構成	118
テスト アプリケーションでのコントロールの識別 (.NET スクリプト)	119
スクリプト内からのアプリケーションの起動	120
スクリプトのオブジェクト マップ項目またはロケーターの手動による記録	121
既存のテストへの追加操作の記録	122
既存のスクリプトを開く	122
スクリプトの編集	122
スクリプトの保存	123
スクリプトでの要素の定義への移動	123
アプリケーション構成	123
アプリケーション構成の編集	124
アプリケーション構成の変更	125
アプリケーション構成の削除	125
[アプリケーションの選択] ダイアログ ボックス	126
リモート ロケーションの編集	126
ユーザー アカウント制御	127
特定のテクノロジ ドメインの無効化	127
アプリケーション構成エラー	128
アプリケーション構成のトラブルシューティング	128
アプリケーション構成での環境変数	129
基本状態	129
基本状態のオン/オフの切り替え	129
基本状態の実行	130
ユーザー インターフェイスからの基本状態の変更	130
スクリプトでの基本状態の変更	132
記録の概要	134
記録中のオブジェクトのハイライト	134
記録中および再生中に除外される文字	135
記録中に利用可能なアクション	135
テストの拡張	137
アクティブ データ	137
アクティブ データ テストの利点	138
アクティブ データ資産	138
アクティブ データ テスト資産の作成	138
アクティブ データ テストのためのテスト ファイルの準備	139
アクティブ データ テスト用の新しいデータ ファイルの作成	140
アクティブ データ テスト ファイルに対する読み取りオプションの設定	141
アクティブ データ テストのためのアクティブ データ ファイルの編集	142
アクティブ データに使用するデータの指定	142

ビジュアルテストでのアクティブ データの使用	144
スクリプトでのアクティブ データの使用	151
ビジュアル テストの拡張	154
ビジュアル テスト ステップで実行される操作の変更	154
ビジュアル テストへのコメントの挿入	154
フラグ	155
テスト ロジックの追加	157
自動ステップの操作	187
ビジュアル テストでの項目識別子の使用	192
ビジュアル テストでのテスト データの再利用	192
テスト フローの管理	208
ビジュアル テストの比較	217
スクリプトの拡張	217
スクリプト内からのスクリプトの参照	217
記録中のスクリプトへの検証の追加	218
コーディングによるスクリプトへの検証の追加	218
スクリプトでのテスト データの再利用	220
スクリプトの管理	235
Windows DLL の呼び出し	240
スクリプトからの Windows DLL の呼び出し	241
DLL 関数の宣言構文	241
DLL 呼び出しの例	241
DLL 関数への引数の受け渡し	242
DLL 関数で変更できる引数の受け渡し	243
DLL 関数への文字列引数の受け渡し	244
DLL 名のエイリアス設定	244
DLL 関数呼び出しの表記規則	244
Microsoft ユーザー補助を使用したオブジェクト解決の向上	245
ユーザー補助の使用	245
ユーザー補助の有効化	246
UI オートメーション	246
UI オートメーション メソッドの動的呼び出し	247
UI オートメーションを使ってコントロールを識別するためのロケーター属性	249
UI オートメーション コントロールでのスクロール	250
UI オートメーション使用時の制限事項	251
UI オートメーション サポートを使ったテスト時のトラブルシューティング	251
Silk Test Workbench での Unicode コンテンツのサポートの概要	251
テキスト解決のサポート	252
カスタム コントロール	254
動的呼び出し	254
テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする	255
Apache Flex カスタム コントロールのテスト	257
カスタム コントロールの管理	258
実行時間の計測	262
テスト実行の遅延	262
テストのデバッグ	263
ビジュアル テストのデバッグ	263
選択したポイントでのビジュアル テスト再生の停止	263
選択したポイントからのビジュアル テスト再生の段階的実行	263
ビジュアル テストの再生におけるステップ実行の制御	264
デバッグ中のビジュアル テストの編集	264
デバッグ モードでのビジュアル テストの段階的実行	265
ビジュアル テスト内でのエラー処理による確実な再生	265
ビジュアル テストの印刷	267

スクリプトのデバッグ	268
スクリプトの信頼性	269
デバッグ モードでのスクリプトの段階的実行	269
選択したポイントからのスクリプト再生の段階的実行	270
スクリプト デバッグ中の行の実行の制御	270
スクリプト例 - エラー処理	270
スクリプトへのコメントの挿入	271
選択したポイントでのスクリプト再生の停止	271
スクリプトの印刷	272
テストの再生と結果の分析	273
ビジュアル テストの再生	273
ビジュアル テストの再生	273
ビジュアル テスト内でのビジュアル テストの実行	274
[ビジュアル テスト変数] ダイアログ ボックス	275
スクリプトの再生	276
スクリプトの再生	276
再生中のオートメーション操作の表示	277
テストの並列実行	277
ビジュアル テスト再生の構成	278
ビジュアル テスト内での再生設定値の設定	279
ビジュアル テスト内での再生設定値の取得	279
挿入されたビジュアル テストまたは VB .NET スクリプトの再生の構成	280
Silk Test Workbench がテストを同期する方法	280
結果の分析	282
[結果] ウィンドウ	283
結果の管理	288
結果フィルタ	294
[条件] ダイアログ ボックス	297
再生ステータス ダイアログ ボックスの有効化	297
[再生完了] ダイアログ ボックス	298
[再生エラー] ダイアログ ボックス	298
資産の管理	299
資産の種類	299
資産のバージョンをページ	300
資産のバージョンをページ	300
ページのコマンド ライン パラメータ	301
アセット ブラウザを使用した資産の作成	304
スクリプトから資産を開く	304
資産を開く	304
資産の比較	304
資産の複製	305
資産名の変更	305
資産の削除	306
資産リストのフィルタ処理	306
資産の検索	307
資産の並べ替え	307
資産保存時のデフォルト動作の設定	307
資産のバージョンの最大数の設定	307
インクルードする資産の表示	308
複数バージョンの資産の表示	308
資産の命名規則	309
テスト資産のインポートとエクスポート	309
資産のインポート	309
資産のエクスポート	311
資産のインポートとエクスポートの権限	312

オブジェクト解決	313
ロケータの基本概念	313
オブジェクトタイプと検索範囲	313
属性を使用したオブジェクトの識別	314
ロケータの構文	314
スクリプトでのロケータの使用	316
Find メソッドの使用	317
ロケータを使用したオブジェクトの存在確認	317
1つのロケータで複数のオブジェクトを識別する	318
ロケータのカスタマイズ	318
安定した識別子	318
カスタム属性	321
XPathのパフォーマンス問題のトラブルシューティング	323
Silk Test Open Agent	325
Silk Test Open Agent の起動	325
テスト実行後に Open Agent を停止する	325
エージェント オプション	325
Silk Test Workbench コンポーネント間の接続設定	334
Open Agent に接続するポートの構成	335
Information Service に接続するポートの構成	336
Silk Test Information Service プロパティの編集	337
Information Service との HTTPS 接続に使用する証明書の置換	338
Open Agent を使用したリモートテスト	339
リモート Open Agent を使用したテスト	339
オブジェクトマップ	340
オブジェクトマップを使用する利点	341
オブジェクトマップのオン/オフの切り替え	341
オブジェクトマップのスコープ	341
操作の記録中でのオブジェクトマップのマージ	342
Web アプリケーションでのオブジェクトマップの使用	343
オブジェクトマップ項目名の変更	343
オブジェクトマップの変更	345
オブジェクトマップのロケータの変更	346
オブジェクトマップの比較	347
テストアプリケーションからのオブジェクトマップの更新	348
オブジェクトマップ項目のコピー	349
オブジェクトマップ項目の追加	350
スクリプトからオブジェクトマップを開く	352
テストアプリケーションでのオブジェクトマップ項目の検索	352
オブジェクトマップのエラーの検出	353
オブジェクトマップ項目の削除	354
オブジェクトマップを最初書き出す	355
オブジェクトマップの要素のグループ化	355
オブジェクトマップ：よくある質問	356
複数のオブジェクトマップを単一のオブジェクトマップのマージする方法	356
テストスクリプトを削除したときにオブジェクトマップで起こること	356
テスト対象アプリケーションのオブジェクトマップを手動で作成する方法	356
イメージ解決のサポート	357
イメージクリックの記録	357
イメージ解決メソッド	357
イメージ資産	358
Creating an Image Asset	358
同じイメージ資産に複数のイメージを追加する	359
スクリプトから資産を開く	360

イメージ クリックの検証資産名での変数の使用	360
イメージ検証	360
イメージ検証の作成	360
ビジュアル テストにイメージ検証を追加する	361
記録中にイメージ検証を追加する	362
イメージ検証が失敗した理由の確認	362
イメージ検証の検証資産名での変数の使用	362
イメージ検証の相違点ビュー	363
特定の環境のテスト	364
Apache Flex のサポート	364
Adobe Flash Player で実行するための Flex アプリケーションの構成	364
Component Explorer の起動	365
Apache Flex アプリケーションのテスト	365
Apache Flex カスタム コントロールのテスト	365
Apache Flex スクリプトのカスタマイズ	375
同一 Web ページ上の複数の Flex アプリケーションのテスト	375
Adobe AIR のサポート	376
名前またはインデックスを使用する Flex の Select メソッドの概要	376
FlexDataGrid コントロールでの項目の選択	377
Flex アプリケーションのテストの有効化	377
Apache Flex アプリケーションのスタイル	389
Adobe Flash Player のセキュリティ制約に対応するための Flex アプリケーションの構成	390
Apache Flex アプリケーションの属性	390
Silk Test Workbench が Apache Flex コントロールを認識できない理由	391
Java AWT/Swing のサポート	391
Java AWT/Swing アプリケーションの属性	392
Java メソッドの動的な呼び出し	392
Java AWT/Swing テクノロジ ドメインでの priorLabel の判別	393
Oracle Forms のサポート	393
Java SWT と Eclipse RCP のサポート	394
Java SWT クラス リファレンス	395
Java SWT カスタム属性	395
Java SWT アプリケーションの属性	395
Java メソッドの動的な呼び出し	396
Java SWT と Eclipse アプリケーションのトラブルシューティング	397
モバイル アプリケーションのテスト	397
Android	398
iOS	404
インストール済みアプリのテスト	417
モバイル アプリケーションの記録	417
テストを再生するモバイル デバイスの選択	417
Mobile Center デバイスの使用	418
Sauce Labs デバイスの使用	419
モバイル デバイスの接続文字列	419
モバイル デバイスの操作	423
モバイル デバイスの開放	423
Silk Central によって管理されているデバイスの使用	424
モバイル アプリケーションのテスト時のトラブルシューティング	424
モバイル Web アプリケーションのテストにおける制限事項	430
ネイティブ モバイル アプリケーションのテストにおける制限事項	431
ネイティブ モバイル アプリに対するメソッドの動的呼び出し	433
モバイル Web サイトでのオブジェクトのクリック	434
既存のモバイル Web テストの使用方法	435
.NET のサポート	435

Windows Forms のサポート	436
Windows Presentation Foundation (WPF) のサポート	441
Silverlight アプリケーションのサポート	449
Visual COBOL のサポート	454
Rumba のサポート	454
Rumba クラス リファレンス	455
Rumba の有効化と無効化	455
Rumba コントロールを識別するためのロケーター属性	455
Rumba での画面検証の使用	456
Unix ディスプレイのテスト	456
SAP のサポート	456
SAP クラス リファレンス	457
SAP アプリケーションの属性	457
SAP メソッドの動的な呼び出し	457
SAP コントロールの動的呼び出し	459
SAP の自動セキュリティ設定の構成	459
SAP eCATT の操作	459
Windows API ベースのアプリケーションのサポート	471
Win32 クラス リファレンス	471
Windows API ベースのクライアント/サーバー アプリケーションの属性	471
Win32 テクノロジ ドメインにおける priorLabel の決定方法	472
埋め込み Chrome アプリケーションのテスト	472
MFC (Microsoft Foundation Class) のサポート	473
クロス ブラウザ テスト	473
テストを再生するブラウザの選択	474
xBrowser でのテスト オブジェクト	476
xBrowser オブジェクトに対するオブジェクト解決	477
xBrowser のページ同期	478
xBrowser における API 再生とネイティブ再生の比較	479
マウス移動の詳細設定	480
xBrowser のブラウザ構成の設定	480
リモート デスクトップ ブラウザーの接続文字列	482
リモート Windows マシン上でのブラウザのテスト	482
Mac 上の Google Chrome または Mozilla Firefox のテスト	483
WebDriver ベースのブラウザのケイパビリティの設定	483
Mac 上の Apple Safari を使用したテスト	485
Google Chrome を使用したテスト	489
Mozilla Firefox を使用したテスト	493
Microsoft Edge を使用したテスト	497
レスポンス Web デザインのテスト	499
ビジュアル ブレークポイントの検出	500
追加のブラウザのバージョンでのテスト	501
クロス ブラウザ テスト:よくある質問	502
スクリプトからのブラウザの起動	506
非表示入力フィールドの検索	507
Web アプリケーションの属性	507
Web アプリケーションのカスタム属性	507
xBrowser クラス リファレンス	508
Microsoft Windows 10 上のテストの制限事項	508
サポートする属性の種類	508
Apache Flex アプリケーションの属性	509
Java AWT/Swing アプリケーションの属性	509
Java SWT アプリケーションの属性	509
SAP アプリケーションの属性	510
Silverlight コントロールを識別するためのロケーター属性	510

UI オートメーションを使ってコントロールを識別するためのロケータ属性	511
Rumba コントロールを識別するためのロケータ属性	512
Web アプリケーションの属性	513
Windows Forms アプリケーションの属性	513
Windows Presentation Foundation (WPF) アプリケーションの属性	513
Windows API ベースのクライアント/サーバー アプリケーションの属性	515
動的ロケータ属性	515
キーワード駆動テスト	517
キーワード駆動テストの利点	517
キーワード	518
Silk Test Workbench でキーワード駆動テストを作成する	519
Silk Test Workbench でのキーワード駆動テストの記録	519
Silk Test Workbench でのキーワード駆動テストの基本状態の設定	521
Silk Test Workbench でのキーワードの実装	521
Silk Test Workbench でのキーワードの記録	522
ビジュアル テストをキーワードとして指定	522
スクリプトのテスト メソッドをキーワードとして指定	523
キーワード間でのデータの受け渡し	523
キーワード駆動テストの編集	524
Silk Central でテストのキーワードを管理する	524
Silk Test Workbench のキーワード レコメンド機能	527
キーワードでのパラメータの使用	527
例：パラメータを取るキーワード	528
キーワードのキーワード シーケンスへの結合	530
変数を指定したキーワード駆動テストの再生	530
Silk Test Workbench と Silk Central の統合	531
Silk Central からキーワードを開く	531
Silk Test Workbench での Silk Central キーワードの実装	532
Silk Central へのキーワード ライブラリのアップロード	533
コマンド ラインから Silk Central へのキーワード ライブラリの更新	533
キーワードの検索	535
キーワードのフィルタリング	535
キーワードのグループ化	535
コマンド ライン	537
SilkTest.Exe コマンド ライン	537
STW.EXE コマンド ライン	537
STW.EXE のパラメータ	537
入力ファイル	542
出力ファイル	543
エラー数	543
Silk Central から STW.EXE を実行するための前提条件	547
STW.EXE の使用例	547
コマンド ラインからの資産のインポートとエクスポート	549
資産のインポートのコマンド ライン パラメータ	549
資産のエクスポートのコマンド ライン パラメータ	552
ログ	555
Silk Test Workbench のオプション	557
グローバル オプション	557
eCATT	557
全般オプション	558
全般オプションの変更	559
開始画面オプション	560
開始画面オプションの変更	561
記録オプション	561

イメージ資産記録設定の設定	562
オブジェクト マップの詳細設定の設定	562
マウス移動の詳細設定	563
無視するクラスの設定	564
ロケーターの詳細設定の設定	564
UI オートメーション オプションの設定	565
ブラウザの記録オプションの設定	566
カスタム属性の設定	567
記録/再生の対象とする WPF クラスの設定	568
記録の出力オプションの設定	568
記録のホットキー オプションの設定	571
記録オプション プロファイルの作成	571
保存済み記録オプションの適用	572
再生オプション	572
全般再生オプションの設定	573
xBrowser 同期オプションの設定	574
再生のタイミング オプションの設定	575
UI オートメーション オプションの設定	576
閉じるオプションの設定	577
再生結果オプションの設定	577
再生ホットキーの変更	580
再生オプション プロファイルの作成	580
保存済み再生オプションの適用	580
スクリプト作成オプション	581
詳細オプションの設定	581
アクティブ データ詳細設定オプション	582
Silk Test Workbench の管理	584
Silk Test Workbench の実行に必要な管理者権限	584
Silk Test Workbench データベースの構成	584
SQL Server データベースの構成	584
Oracle データベースの構成	587
ドメインなしでの SQL Server または Oracle データベースの設定	593
データ ソース名の作成	594
データベースへ接続する	596
データベース メンテナンス	598
Silk Test Workbench データベースを開く	599
データベース バージョンの更新	600
データベース レコードのコピー	601
データベースのレコードのロック解除	601
Access データベースのコンパクト化	602
データベース メンテナンス設定の構成	602
接続ユーザーがいるデータベースのメンテナンスの実行	603
データベース整合性チェックの実行	603
データベースの膨張の抑制	603
プロジェクトの管理	604
プロジェクトの追加	604
プロジェクトの参照	605
プロジェクト参照の表示	605
グローバル プロジェクトの定義	605
Silk Test Workbench がプロジェクトの循環参照を回避する方法	606
プロジェクト名の変更	607
プロジェクトの複製	608
プロジェクトの削除	608
ユーザーの管理	609
データベースに接続中のユーザーの表示	609

ユーザーの追加	609
ユーザーの削除	610
ユーザー プロファイルの編集	610
プロジェクトに対するユーザーのアクセス権の変更	610
データベースからのユーザーのログアウト	610
グループの管理	611
グループの追加	611
グループの削除	612
グループの変更	612
UI トピック	613
開始画面	613
Silk Test Workbench のメニュー	614
[ファイル] メニュー	614
[編集] メニュー	615
[操作] メニュー	616
[表示] メニュー	616
[挿入] メニュー	618
[デバッグ] メニュー	619
[ツール] メニュー	620
[ウィンドウ] メニュー	622
[ヘルプ] メニュー	622
ツールバー	622
ビジュアルナビゲータ	625
[ビジュアルナビゲータ]: [テスト ステップ] ペイン	627
[ビジュアルナビゲータ]: [画面プレビュー]	631
[ビジュアルナビゲータ]: [プロパティ]	632
[ビジュアルナビゲータ]: [ストーリーボード]	633
ビジュアルナビゲータのレイアウトのカスタマイズ	634
[結果] ウィンドウ	634
アセットブラウザ	635
[オブジェクトの識別] ダイアログ ボックス	636
[コード] ウィンドウ	638
スクリプトの [プロパティ] ペインの表示	638
出カウインドウ	639
既知の問題	640
全般的な問題	640
モバイル Web アプリケーション	642
Web アプリケーション	642
Google Chrome	642
Internet Explorer	643
Microsoft Edge	644
Mozilla Firefox	645
SAP アプリケーション	645
Oracle Forms	647
Silk Test Workbench	647
Silk Test Workbench 言語リファレンス	650
共通クラス リファレンス	650
BaseGuiTestObject クラス	650
CharSet 列挙	654
CheckBox クラス	654
CheckBoxToolItem クラス	658
ComboBox クラス	662
Control クラス	666
Desktop クラス	671

Dialog クラス	671
DropDownToolItem クラス	676
FileInfo クラス	679
FileSizeUnit 列挙	680
Group クラス	680
GuiTestObject クラス	684
HorizontalScrollBar クラス	688
IClickable インターフェイス	692
IFocusable インターフェイス	693
IKeyable インターフェイス	693
IMoveable インターフェイス	693
INativeWindow インターフェイス	694
IScrollable インターフェイス	694
Item クラス	695
Label クラス	699
Link クラス	703
ListBox クラス	706
Menu クラス	711
MenuItem クラス	714
PushButton クラス	718
PushToolItem クラス	722
RadioList クラス	725
RadioListToolItem クラス	729
RegistryCategory 列挙	733
RegistryView 列挙	734
Scale クラス	734
ScrollBar クラス	738
SeparatorItem クラス	742
TabControl クラス	746
Table クラス	750
TableColumn クラス	754
TableRow クラス	758
TestObject クラス	762
TextField クラス	764
ToggleButton クラス	768
ToolBar クラス	772
ToolItem クラス	776
Tree クラス	779
VerticalScrollBar クラス	784
Window クラス	788
ExecutionMode 列挙	793
ExecutionResult クラス	793
FileHandle クラス	794
FileOpenMode 列挙	794
FilePointerMode 列挙	794
FileShareMode 列挙	795
SystemFunctions クラス	795
コア クラス リファレンス	798
ActiveData クラス	798
ActiveDataRow クラス	803
Agent クラス	808
BaseState クラス	814
BrowserBaseState クラス	818
ConsoleWindow クラス	819
DllCall クラス	823

IBaseState インターフェイス	824
Timer クラス	825
Workbench クラス	829
データ型	836
ClickType 列挙	836
Color クラス	836
ItemIdentifier データ型	837
ItemPath データ型	838
ModifierKeys 列挙	838
MouseButton 列挙	839
Point クラス	839
Range クラス	839
Rectangle クラス	840
TextPosition クラス	840
TextRange クラス	841
TreeContent クラス	841
TreeNode クラス	842
Flex クラス リファレンス	842
FlexAccordion クラス	842
FlexAdvancedDataGrid クラス	849
FlexAlert クラス	858
FlexApplication クラス	865
FlexAreaChart クラス	872
FlexAreaSeries クラス	879
FlexAxisRenderer クラス	885
FlexBarChart クラス	891
FlexBarSeries クラス	898
FlexBox クラス	904
FlexBubbleChart クラス	910
FlexBubbleSeries クラス	917
FlexButton クラス	923
FlexButtonBar クラス	929
FlexCandlestickChart クラス	936
FlexCandlestickSeries クラス	943
FlexCanvas クラス	949
FlexCartesianChart クラス	956
FlexChart クラス	963
FlexChartLegend クラス	970
FlexChartSeries クラス	975
FlexCheckBox クラス	981
FlexColorPicker クラス	988
FlexColumnChart クラス	994
FlexColumnSeries クラス	1001
FlexComboBase クラス	1007
FlexComboBox クラス	1014
FlexContainer クラス	1021
FlexContainerMovieClip クラス	1027
FlexDataGrid クラス	1032
FlexDateChooser クラス	1041
FlexDateField クラス	1047
FlexDisplayObject クラス	1054
FlexDividedBox クラス	1058
FlexForm クラス	1065
FlexFormItem クラス	1072
FlexHLOCChart クラス	1078

FlexHLOCSeries クラス	1085
FlexHLOCSeriesBase クラス	1091
FlexImage クラス	1097
FlexLabel クラス	1103
FlexLineChart クラス	1108
FlexLineSeries クラス	1115
FlexLinkBar クラス	1121
FlexList クラス	1128
FlexListBase クラス	1136
FlexListLabel クラス	1143
FlexLoader クラス	1147
FlexMenu クラス	1152
FlexMenuBar クラス	1158
FlexNavigationBar クラス	1164
FlexNumericStepper クラス	1171
FlexObject クラス	1177
FlexOLAPDataGrid クラス	1182
FlexPanel クラス	1192
FlexPieChart クラス	1198
FlexPieSeries クラス	1205
FlexPlotChart クラス	1211
FlexPlotSeries クラス	1218
FlexPopUpButton クラス	1224
FlexProgressBar クラス	1231
FlexRadioButton クラス	1237
FlexRepeater クラス	1243
FlexRule クラス	1247
FlexScrollBar クラス	1252
FlexScrollBase クラス	1257
FlexSlider クラス	1263
FlexStandalonePlayer クラス	1269
FlexTabNavigator クラス	1273
FlexTextArea クラス	1280
FlexTitleWindow クラス	1286
FlexToggleButtonBar クラス	1293
FlexTree クラス	1300
FlexUIMovieClip クラス	1308
FlexVideoDisplay クラス	1313
FlexViewStack クラス	1318
FlexWindow クラス	1325
FlexWindowedApplication クラス	1332
SparkAirHTML クラス	1340
SparkApplication クラス	1345
SparkBorderContainer クラス	1351
SparkButton クラス	1357
SparkButtonBar クラス	1363
SparkButtonBarButton クラス	1369
SparkButtonBase クラス	1375
SparkCheckBox クラス	1380
SparkComboBox クラス	1386
SparkComplexDisplay クラス	1392
SparkDataGrid クラス	1399
SparkDataGridLabel クラス	1405
SparkDataGroup クラス	1410
SparkDataRenderer クラス	1417

SparkDropDownList クラス	1424
SparkDropDownListBase クラス	1430
SparkForm クラス	1437
SparkFormItem クラス	1443
SparkGroup クラス	1449
SparkGroupBase クラス	1456
SparkImage クラス	1462
SparkLabel クラス	1468
SparkList クラス	1473
SparkListBase クラス	1480
SparkListLabel クラス	1486
SparkMuteButton クラス	1493
SparkNavigatorContent クラス	1498
SparkNumericStepper クラス	1504
SparkObject クラス	1510
SparkPanel クラス	1515
SparkPopUpAnchor クラス	1521
SparkRadioButton クラス	1527
SparkRange クラス	1532
SparkRichEditableText クラス	1538
SparkRichText クラス	1544
SparkScrollBar クラス	1550
SparkSkinnableContainer クラス	1555
SparkSkinnableContainerBase クラス	1561
SparkSkinnableDataContainer クラス	1567
SparkSkinnablePopUpContainer クラス	1572
SparkSkinnableTextBase クラス	1578
SparkSlider クラス	1584
SparkSpinner クラス	1590
SparkTabBar クラス	1595
SparkTextArea クラス	1601
SparkTextBase クラス	1608
SparkTextInput クラス	1613
SparkTileGroup クラス	1620
SparkTitleWindow クラス	1627
SparkToggleButton クラス	1633
SparkToggleButtonBase クラス	1638
SparkTrackBase クラス	1644
SparkVideoDisplay クラス	1649
SparkVideoPlayer クラス	1655
SparkVolumeBar クラス	1660
SparkWindow クラス	1666
SparkWindowedApplication クラス	1673
Java SWT クラス リファレンス	1680
CBanner クラス	1680
CoolBar クラス	1684
CoolItem クラス	1688
CTabFolder クラス	1692
CTabItem クラス	1696
ExpandBar クラス	1700
ExpandItem クラス	1704
HorizontalSash クラス	1707
Sash クラス	1711
SashForm クラス	1715
ScrollableControl クラス	1719

ScrolledComposite クラス	1723
Shell クラス	1727
Spinner クラス	1731
StyledText クラス	1735
SWTBrowser クラス	1740
SWTDateTime クラス	1743
SWCTabControl クラス	1747
SWCTabItem クラス	1751
SWCTable クラス	1755
SWCTableColumn クラス	1760
SWCTableRow クラス	1763
SWCTree クラス	1767
SWCTreeColumn クラス	1772
VerticalSash クラス	1776
ViewForm クラス	1780
キーワード駆動テスト クラス リファレンス	1784
ArgumentAttribute クラス	1784
KeywordAttribute クラス	1785
KeywordGroupAttribute クラス	1785
Mobile クラス リファレンス	1786
IMobileClickable インターフェイス	1786
IMobileGestures インターフェイス	1786
IMobileKeyable インターフェイス	1787
MobileButton クラス	1787
MobileDevice クラス	1791
MobileObject クラス	1796
MobileTextField クラス	1800
MobileWindow クラス	1804
Rumba クラス リファレンス	1807
Rumba キー	1808
RumbaCharacterAttribute データ型	1810
RumbaField クラス	1811
RumbaLabel クラス	1815
RumbaObject クラス	1819
RumbaScreen クラス	1823
RumbaTextField クラス	1827
SAP クラス リファレンス	1832
ISapContextMenuable インターフェイス	1832
SapBarChart クラス	1832
SapBox クラス	1836
SapButton クラス	1840
SapCalendar クラス	1844
SapChart クラス	1848
SapCheckBox クラス	1852
SapColorSelector クラス	1856
SapComboBox クラス	1860
SapComponent クラス	1864
SapContainer クラス	1868
SapContainerShell クラス	1872
SapContextMenu クラス	1876
SapCustomControl クラス	1879
SapDockShell クラス	1883
SapGridView クラス	1887
SapHorizontalScrollBar クラス	1894
SapHTMLViewer クラス	1897

SapLabel クラス	1901
SapMenu クラス	1905
SapMenubar クラス	1909
SapNetPlan クラス	1913
SapOfficeIntegration クラス	1917
SapOkCodeField クラス	1921
SapPicture クラス	1925
SapRadioButton クラス	1929
SapScrollbar クラス	1933
SapScrollContainer クラス	1936
SapShell クラス	1940
SapSimpleContainer クラス	1944
SapSplitterContainer クラス	1947
SapStatusbar クラス	1951
SapTab クラス	1955
SapTable クラス	1959
SapTabStrip クラス	1963
SapTextEdit クラス	1967
SapTextField クラス	1971
SapTitlebar クラス	1975
SapToolbar クラス	1979
SapToolbarControl クラス	1983
SapTree クラス	1987
SapUserArea クラス	1994
SapVerticalScrollBar クラス	1998
SapWindow クラス	2001
Silverlight クラス リファレンス	2006
SLApplication クラス	2006
SLAutoCompleteBox クラス	2009
SLBase クラス	2014
SLButton クラス	2018
SLCalendar クラス	2022
SLCalendarButton クラス	2026
SLCalendarDayButton クラス	2030
SLCheckBox クラス	2034
SLComboBox クラス	2038
SLComboBoxItem クラス	2042
SLDataGrid クラス	2046
SLDataGridCell クラス	2050
SLDataGridDetails クラス	2054
SLDataGridRow クラス	2058
SLDataPager クラス	2062
SLDatePicker クラス	2066
SLDescriptionViewer クラス	2069
SLFrame クラス	2073
SLGridSplitter クラス	2077
SLGroup クラス	2080
SLHeader クラス	2084
SLHeaderItem クラス	2088
SLHorizontalScrollBar クラス	2091
SLHyperlinkButton クラス	2095
SLImage クラス	2099
SLListBox クラス	2103
SLListItem クラス	2107
SLMediaElement クラス	2111

SLMenu クラス	2115
SLMenuBar クラス	2118
SLMenuItem クラス	2122
SLMultiScaleImage クラス	2126
SLPane クラス	2129
SLPasswordBox クラス	2133
SLPopup クラス	2137
SLProgressBar クラス	2141
SLRadioButton クラス	2145
SLRepeatButton クラス	2148
SLRichTextBox クラス	2152
SLSeparator クラス	2156
SLSlider クラス	2160
SLSpinner クラス	2164
SLSplitButton クラス	2167
SLStatusBar クラス	2171
SLTabControl クラス	2175
SLTabItem クラス	2179
SLTable クラス	2182
SLTextBlock クラス	2186
SLTextBox クラス	2190
SLThumb クラス	2194
SLTitleBar クラス	2197
SLToggleButton クラス	2201
SLToolBar クラス	2205
SLToolTip クラス	2209
SLTreeView クラス	2212
SLTreeViewItem クラス	2217
SLValidationSummary クラス	2221
SLVerticalScrollBar クラス	2224
SLWindow クラス	2228
Java AWT および Swing クラス リファレンス	2232
AbstractButton クラス	2232
Applet クラス	2236
AppletContainer クラス	2240
AWTButton クラス	2244
AWTCanvas クラス	2248
AWTCheckbox クラス	2252
AWTCheckboxMenuItem クラス	2256
AWTChoice クラス	2260
AWTComponent クラス	2264
AWTContainer クラス	2268
AWTDialog クラス	2272
AWTFrame クラス	2277
AWTHorizontalScrollbar クラス	2282
AWTLabel クラス	2286
AWTList クラス	2290
AWTMenu クラス	2295
AWTMenuComponent クラス	2298
AWTMenuItem クラス	2301
AWTRadioButton クラス	2305
AWTScrollbar クラス	2309
AWTScrollPane クラス	2313
AWTTextArea クラス	2318
AWTTextComponent クラス	2323

AWTTextField クラス	2327
AWTVerticalScrollbar クラス	2332
AWTWindow クラス	2336
BasicArrowButton クラス	2341
IAWTScrollable インターフェイス	2345
IAWTScroller インターフェイス	2345
IBaseScrollable インターフェイス	2346
IOracleFormsMenuBase インターフェイス	2346
IOracleFormsScrollable インターフェイス	2347
IOracleFormsScroller インターフェイス	2348
JButton クラス	2348
JCheckBox クラス	2352
JCheckBoxMenuItem クラス	2357
JColorChooser クラス	2361
JComboBox クラス	2365
JComponent クラス	2369
JDesktopPane クラス	2374
JDialog クラス	2378
JEditorPane クラス	2382
JFrame クラス	2387
JHorizontalScrollBar クラス	2392
JLabel クラス	2396
JLayeredPane クラス	2400
JList クラス	2404
JMenu クラス	2409
JMenuBar クラス	2413
JMenuItem クラス	2417
JPanel クラス	2421
JPasswordField クラス	2425
JPopupMenu クラス	2430
JProgressBar クラス	2434
JRadioButton クラス	2438
JRadioButtonMenuItem クラス	2442
JRootPane クラス	2446
JScrollBar クラス	2451
JScrollPane クラス	2455
JSlider クラス	2459
JSpinner クラス	2463
JSplitPane クラス	2468
JTabbedPane クラス	2472
JTable クラス	2476
JTableHeader クラス	2481
JTextArea クラス	2485
JTextComponent クラス	2490
JTextField クラス	2494
JTextPane クラス	2499
JToggleButton クラス	2503
JToolBar クラス	2508
JTree クラス	2512
JVerticalScrollBar クラス	2516
JViewport クラス	2521
JWindow クラス	2525
OracleFormsApplication クラス	2529
OracleFormsButton クラス	2533
OracleFormsCheckbox クラス	2537

OracleFormsChoice クラス	2541
OracleFormsComboBox クラス	2546
OracleFormsContainer クラス	2550
OracleFormsHorizontalScrollbar クラス	2554
OracleFormsLabel クラス	2559
OracleFormsListBox クラス	2563
OracleFormsListView クラス	2568
OracleFormsMenu クラス	2574
OracleFormsMenuItem クラス	2578
OracleFormsPopList クラス	2582
OracleFormsPopupMenu クラス	2586
OracleFormsRadioButton クラス	2590
OracleFormsScrollbar クラス	2595
OracleFormsStatusArea クラス	2599
OracleFormsStatusBar クラス	2603
OracleFormsStatusBarItem クラス	2607
OracleFormsStatusIndicator クラス	2611
OracleFormsTabBar クラス	2615
OracleFormsTabBarItem クラス	2619
OracleFormsTabPanel クラス	2623
OracleFormsTextField クラス	2627
OracleFormsTitleBar クラス	2633
OracleFormsToolBar クラス	2637
OracleFormsToolBarItem クラス	2641
OracleFormsTree クラス	2645
OracleFormsVerticalScrollbar クラス	2650
SplitPaneDivider クラス	2654
Win32 クラス リファレンス	2658
AccessibleControl クラス	2658
Header クラス	2662
ListView クラス	2666
MonthCalendar クラス	2671
Pager クラス	2675
ProgressBar クラス	2679
StatusBar クラス	2683
UpDown クラス	2687
Windows Forms クラス リファレンス	2691
CheckedListBox クラス	2691
DataGrid クラス	2695
DataGridColumn クラス	2699
DataGridItem クラス	2703
DataGridRow クラス	2707
DomainUpDown クラス	2711
ElementHost クラス	2715
FormsHost クラス	2719
FormsWindow クラス	2723
MenuStrip クラス	2727
NumericUpDown クラス	2731
xBrowser クラス リファレンス	2735
BrowserApplication クラス	2735
BrowserObject クラス	2741
BrowserWindow クラス	2744
DomButton クラス	2748
DomCheckBox クラス	2753
DomElement クラス	2757

DomEmbeddedElement クラス	2762
DomForm クラス	2766
DomLink クラス	2771
DomListBox クラス	2775
DomRadioButton クラス	2780
DomTable クラス	2784
DomTableRow クラス	2789
DomTextField クラス	2793

Silk Test Workbench 19.5 へようこそ



Silk Test Workbench 19.5 へようこそ

はじめに
ドキュメント
製品スイート



新機能

Silk Test Workbench 19.5 の新機能
リリース ノート



主なセクション

テストの作成
テストの再生と結果の分析
テストのデバッグ
オブジェクト解決



チュートリアルとデモ

ビジュアル テスト チュートリアル
スクリプト チュートリアル
Training Store の無料の Web ベース トレーニング



コード サンプル

言語リファレンス



オンライン リソース

Micro Focus リソース ページ
Micro Focus のホーム ページ
YouTube の *Micro Focus* チャンネル
Silk Test ドキュメント
オンライン ドキュメント
Micro Focus SupportLine
Micro Focus 製品アップデート
Silk Test Knowledge Base
Silk Test Forum

Web ベースのトレーニング




フィードバックをお寄せください

[Micro Focus へのお問い合わせ \(78 ページ\)](#)



[このヘルプに関するフィードバックを電子メールでお送りください](#)
[機能を提案する](#)

ライセンス情報

評価版を使用しているのではない限り、Silk Test はライセンスを必要とします。

 **注:** Silk Test ライセンスは、Silk Test の特定のバージョンに固定されています。たとえば、Silk Test 19.5 には Silk Test 19.5 のライセンスが必要です。

ライセンス モデルは、使用しているクライアントとテストすることができるアプリケーションに基づきま
す。利用可能なライセンス モードに応じて、次のアプリケーションの種類がサポートされます。

ライセンス モード	アプリケーションの種類
モバイル ネイティブ	<ul style="list-style-type: none">モバイル Web アプリケーション<ul style="list-style-type: none">AndroidiOSネイティブ モバイル アプリケーション<ul style="list-style-type: none">AndroidiOS
フル	<ul style="list-style-type: none">Web アプリケーション (以下を含む)<ul style="list-style-type: none">Apache FlexJava アプレットモバイル Web アプリケーション<ul style="list-style-type: none">AndroidiOSApache FlexJava AWT/Swing (Oracle Forms を含む)Java SWT と Eclipse RCP.NET (Windows Forms および Windows Presentation Foundation (WPF) を含む)RumbaWindows API ベース <p> 注: ライセンスをフル ライセンスにアップグレードする場合は、http://www.microfocus.co.jp に移動します。</p>
プレミアム	フル ライセンスでサポートされるすべてのアプリケーションの種類 + SAP アプリケーション <p> 注: ライセンスをプレミアム ライセンスにアップグレードする場合は、http://www.microfocus.co.jp に移動します。</p>
モバイル ネイティブ アドオン	フル ライセンスとプレミアム ライセンスでサポートされるテクノロジーに加えて、モバイル ネイティブ アドオン ライセンスによって、Android と iOS 上でのネイティブ モバイル アプリケーション テストのサポートを提供します。

はじめに

Silk Test Workbench は、自分たちのテストでの使いやすさを重視するビジネス アナリスト、ドメイン エキスパート、オートメーション エキスパートにとって理想的な Silk Test クライアントです。Silk Test Workbench にはビジュアル テスト オプションがあり、コードを作成せずにテストを視覚的に作成できる、直感的にわかりやすい強力な方法を提供します。スクリプト作成オプションでは、さらに強力で柔軟性の高いテストを実行でき、ビジュアル テストと組み合わせて VB.NET スクリプトを拡張できます。Silk Test Workbench を使用すると、アプリケーションとのユーザー セッションを記録してテストを作成し、検証やテスト ロジックを追加してテストを拡張し、テストを再生してアプリケーションが期待どおりに動作することを確認することができます。

このセクションでは、Silk Test Workbench の概要と Silk Test Workbench インターフェイスの概要について説明します。

ログオン

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > クライアント > Silk Test Workbench**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Workbench** をクリックします。Silk Test Workbench が起動され、**SilkTest Workbench にログイン** ダイアログ ボックスが表示されます。

2. **認証** リストボックスから、**Silk Test Workbench** 認証、または **Windows** 認証のいずれかを使用する方を選択します。



注: Windows 認証を使用できるユーザーは、**ツール > 管理** を使用して Silk Test Workbench ユーザーに追加されたユーザーだけです。

3. **Silk Test Workbench** 認証を使用している場合、**ユーザー名** フィールドにユーザー名を入力し、**パスワード** フィールドにパスワードを入力します。**Windows** 認証を使用している場合は、ユーザー名とパスワードを入力する必要はありません。



注: Silk Test Workbench をはじめて使用する場合は、管理者権限を持つユーザー名とパスワードが必要になる場合があります。デフォルトのユーザー名は Admin で、デフォルトのパスワードは admin です。不正アクセスを防ぐため、このパスワードはログオンのあとで変更してください。ログオンパスワードはいつでも変更できます。

4. **データベース** リストから、使用するデータベースを選択します。

データベースが **データベース** リストに表示されるようにするには、まずデータベース接続を使用できるように構成する必要があります。データベース接続を構成するには、**ログオン** ダイアログ ボックスで **構成** ボタンをクリックします。

5. **OK** をクリックします。

Silk Test Workbench データベースに初めてログインすると、*Project1* という名前のデフォルトプロジェクトがデータベースに作成されます。デフォルトでは、作成する資産は、このプロジェクトに保存されます。プロジェクト名は、**プロジェクトの編集** をクリックすると変更できます。

開始画面 が開きます。Silk Test Workbench に始めてログインすると、Silk Test Workbench は *Project1* または最初に利用可能な非グローバル オブジェクトをアクティブ オブジェクトとして選択します。

ログオン パスワードの変更

ログオン パスワードを変更するには、以下の手順に従います。

1. **ツール > パスワードの変更** をクリックします。**パスワードの変更** ダイアログ ボックスが開きます。現在のユーザーのログオン名が、ダイアログ ボックスのタイトル バーに表示されます。

2. パスワードを変更し、**OK** をクリックします。このユーザー ID を使って次回ログオンしたとき、変更が有効になります。

Silk Test Workbench の概要

Silk Test Workbench は、Microsoft、Java、Web、その他多くの分散テクノロジーで開発された複雑なアプリケーションの機能テストを高速化する自動テストツールです。Silk Test Workbench を使用すると、アプリケーションとのユーザー セッションを記録してテストを作成し、検証やテスト ロジックを追加してテストを拡張し、テストを再生してアプリケーションが期待どおりに動作することを確認することができます。

Silk Test Workbench には、テストの自動化を開発するための 2 つのオプションがあります。以下を使用できます。

- ビジュアル テスト
- VB ベースの .NET スクリプト

Silk Test Workbench によりユーザー操作を確実に記録し、強力なビジュアル テストをすばやく作成できます。記録したビジュアル テストはそれぞれ、わかりやすい簡潔なステップの一連の操作として表示されます。これは初心者からエキスパートまであらゆるテスト担当者が容易に理解できるものです。テスト担当者は、注文の処理や顧客情報の入力など、すべてのユーザー操作について Silk Test Workbench の記録方法を広範に制御できます。テスト担当者は、記録中または記録終了後いつでもテストを編集して、テストプロジェクトの最も困難な要求にも合わせるすることができます。

Silk Test Workbench は、さまざまな開発ツールで開発されたアプリケーションのテストをサポートします。テスト ロジック、実際のデータ、検証、およびエラー処理をテストに追加して、さらに柔軟性と信頼性を上げることが可能です。

Silk Test Workbench は、テスト、開発、品質保証、その他のプロジェクト関係者間でのテスト ステータスの効率的な伝達をサポートするプロジェクト協業機能により、チームの生産性を向上させます。アドホッククエリと詳細なレポートは、チームがテストの実行結果を解釈して、情報に基づきテストプロジェクトに関する意思決定を行うことを可能にします。

Silk Test Workbench を使用する利点

Silk Test Workbench は、アプリケーション ソフトウェア、Web アプリケーション、およびモバイル アプリケーションのテスト手順の計画、開発、実行とバグの検出を支援します。Silk Test Workbench では、アプリケーションのセッションを記録し、実際の結果が予想した結果と一致するかどうかを調べる検証機能を追加したうえで、いつでもセッションを再生してアプリケーションが期待どおりに機能するかを確認できます。

Silk Test Workbench は、開発チームに以下のような利点を提供します。

- | | |
|----------------------|---|
| 開発/テストサイクルの短縮 | Silk Test Workbench では、マシンの使用時間をスケジュールしやすい夜間や週末に無人でテストを再生できるため、開発とテストのスケジュールを早めてアプリケーションをより早く市場に出すことができます。 |
| 一貫性 | Silk Test の自動テストは毎回同じように実行されるため、再現可能な一貫した結果が生成されます。 |
| 生産性の向上 | Silk Test Workbench は手動操作なしでテストを再生できるため、プログラマーやテスト担当者は他の作業に時間を使うことができます。 |
| 迅速なテスト開発 | アプリケーションをナビゲートしてテストしながらユーザーの操作を記録することにより、有用なテストをすばやく簡単に作成できます。テンプレートを設定して、テストの標準フレームワークを作成することもできます。自動テストは、手動によるテストの数分の 1 の時間で実行できます。 |

テスト結果の確認 テストスクリプトとビジュアル テスト再生の詳細は結果に記録され、いつでも確認できます。

特殊な状況下での自動化 (周辺機器が無い)

製品の基本的な位置付け

Silk Test Workbench は GUI テスト製品で、自動化された状況下での有意なテスト結果を得るために、人間のように振舞います。Silk Test Workbench が実行したテストは、人間が実行するよりもすばやく実行しますが、同等の価値のあるものです。このことは、人間が同じテストを実行するために必要なテスト環境とできる限り同等なテスト環境を Silk Test Workbench が必要とすることを意味します。

物理的な周辺機器

実際のアプリケーション UI の手動テストでは、キーボード、マウス、ディスプレイなどの入出力デバイスが必要です。Silk Test Workbench では、テストの再生時に物理的な入力デバイスを必要としません。Silk Test Workbench に必要なものは、キーストロークやマウス クリックを実行するオペレーティング システムの機能です。大抵の場合、入力デバイスが接続されていなくても、Silk Test Workbench の再生は期待通り動作します。ただし、デバイス ドライバによっては、物理的な入力デバイスが利用可能でないと、Silk Test Workbench の再生機構をブロックする場合があります。

同じことが物理的な出力デバイスについても言えます。物理的なディスプレイが接続されている必要はありませんが、機能するビデオ デバイス ドライバがインストールされ、オペレーティング システムが画面にレンダリングできる状態になければなりません。たとえば、スクリーン セーバー モードやセッションがロックされている状況では、レンダリングできません。レンダリングできない場合、低レベルの再生は機能せず、高レベルの再生もテスト対象アプリケーション (AUT) で使用するテクノロジーに依存しますが、期待通り機能しない可能性があります。

仮想マシン

Silk Test Workbench は仮想化ベンダーを直接サポートしませんが、仮想ゲスト マシンが物理マシンと同等に動作する限り、任意の仮想化手法のもとで動作可能です。標準的な周辺機器は、通常は仮想デバイスとして提供されており、仮想マシンを実行するマシンで使用されている物理デバイスとは無関係です。

クラウド インスタンス

自動化の観点からは、クラウド インスタンスは仮想マシンと変わりありません。ただし、クラウド インスタンスでは、ビデオ レンダリングに特殊な最適化が行われている場合があります、ハードウェア リソースの消費を抑えるために、画面のレンダリングが一時的にオフになる状況があります。これは、ディスプレイを表示しているアクティブなクライアントが無いと、クラウド インスタンスが検知した場合に発生する場合があります。このような場合、回避策として VNC ウィンドウを開くことができます。

特殊な状況

ウィンドウが無く起動されるアプリケーション (ヘッドレス) このようなアプリケーションは、Silk Test Workbench を使ってテストできません。Silk Test Workbench は、対象のアプリケーション プロセスにフックして、対話操作する必要があります。ウィンドウが表示されないプロセスをフックすることはできません。このような場合は、システム コマンドの実行のみ可能です。

リモート デスクトップ、ターミナル サービス、リモート アプリケーション (すべてのベンダー) Silk Test Workbench がリモート デスクトップ セッション側に存在し、操作する場合、完全に期待通りの操作が行われます。



注: フル ユーザー セッションが必要で、リモート表示ウィンドウは最大化されている必要があります。リモート表示ウィンドウが何らかの理由で表示されていない場合 (ネットワーク上の問題など)、Silk Test Workbench は再生を続けますが、使用されているリモート表示技術によっては予期しない結果を生じ

る可能性があります。たとえば、リモートデスクトップセッションが失われると、ビデオレンダリングに悪影響を与えますが、他のリモート表示手法では、一度表示されたウィンドウが失われても、問題なく表示されるものもあります。

Silk Test Workbench がリモートデスクトップ、リモートビュー、リモート アプリ ウィンドウなどとの対話操作に使用される場合は、Silk Test Workbench が見ることができるのはリモート マシンのスクリーンショットだけであるため、低レベルな技術だけが使用できます。リモート表示技術によっては、セキュリティ上の制約により、低レベル操作でさえできないものもあります。たとえば、リモート アプリケーション ウィンドウにキーストロークを送信できない場合があります。

既知の自動化の障壁

Silk Test Workbench では、ログオンした対話的なフル ユーザー セッションが必要です。スクリーンセーバー、休止状態、スリープ モードなどのセッションをロックするものは無効化してください。組織の方針などで、これができない場合は、キーボード アライブ 操作 (定期的にあるいは各テストケースの終わりにマウスを動かすなど) を追加することによって、このような問題を回避できます。



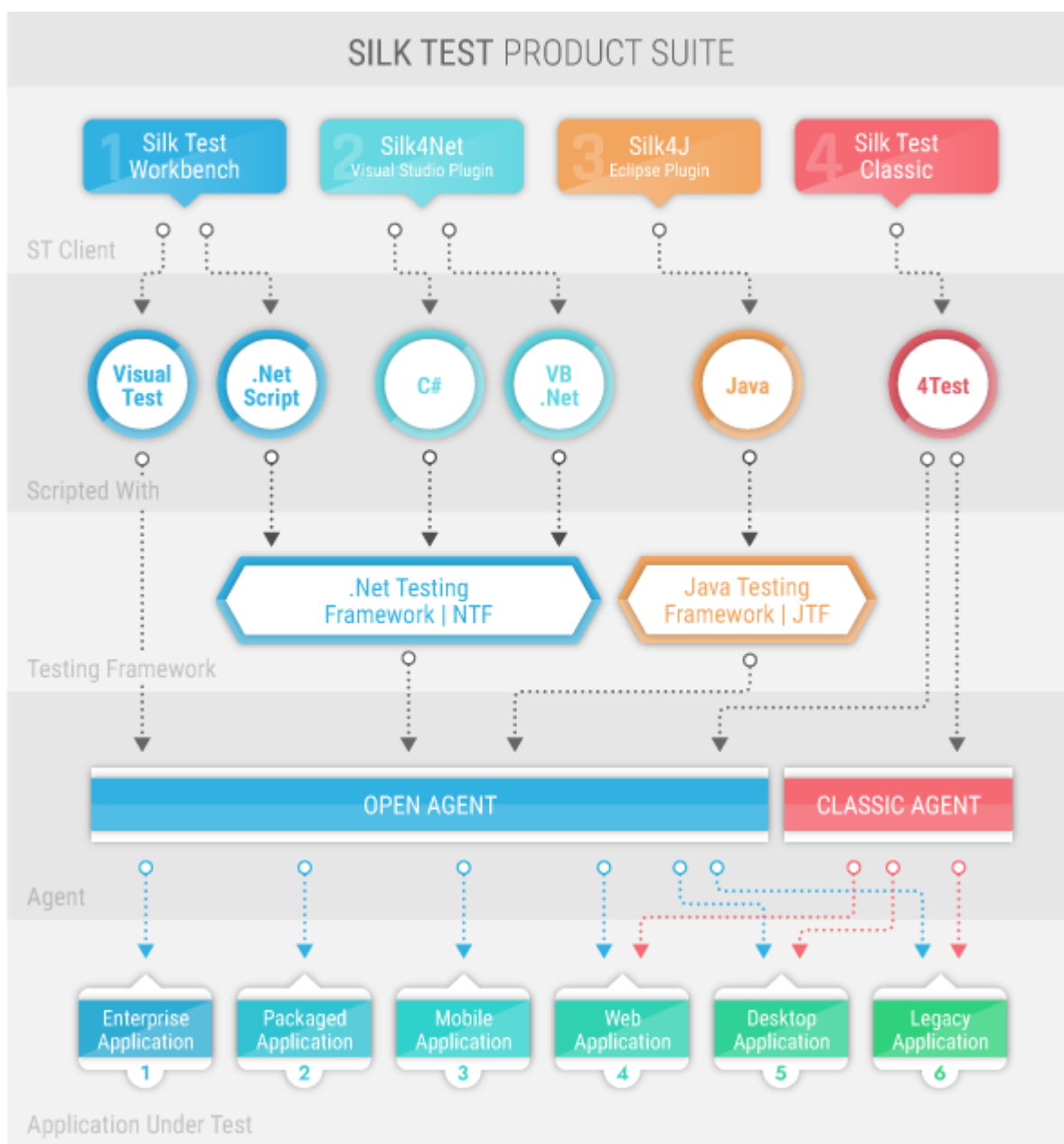
注: 実際のテスト環境の構成や AUT、仮想化、ターミナル サービスで使用される技術によっては、テストの自動化プロセスにおいて、さらなる問題や制約に直面する可能性があります。

Silk Test 製品スイート

Silk Test は、高速で信頼性の高い機能テストと回帰テストを行うための自動テスト ツールです。Silk Test は、高品質のソフトウェアをすばやくリリースするために、開発チーム、品質管理チーム、ビジネス アナリストを支援します。Silk Test を使用すると、アプリケーションが意図したとおりに動作することを確実にするために、複数のプラットフォームとデバイス上でテストを記録/再生することができます。

Silk Test 製品スイートには、以下のコンポーネントが含まれています。

- Silk Test Workbench : Silk Test Workbench は、品質テスト環境です。上級者用の .NET スクリプトと、より幅広い利用者がテストを行えるようにする使いやすいビジュアルテストが提供されます。
- Silk4NET : Silk4NET Visual Studio プラグインを使用すると、Visual Studio で直接 VB.NET または C# のテスト スクリプトを作成できます。
- Silk4J : Silk4J Eclipse プラグインを使用すると、Eclipse 環境で直接 Java ベースのテスト スクリプトを作成できます。
- Silk Test Classic : Silk Test Classic は、4Test を使用したスクリプトを作成できる Silk Test クライアントです。
- Silk Test Agent : Silk Test Agent は、テストのコマンドを GUI 固有のコマンドに変換するソフトウェア プロセスです。つまり、テストするアプリケーションをエージェントが動かし、監視しています。ホストマシン上で 1 つのエージェントをローカルに実行できます。ネットワーク環境では、任意の数のエージェントをリモート マシン上で実行できます。



上の画像の個々のボックスのサイズは、記述上の理由で異なっているだけで、含まれる機能を反映しているわけではありません。

インストールする製品スイートによって、使用できるコンポーネントが決まります。すべてのコンポーネントをインストールするには、完全インストール オプションを選択します。Silk Test Classic を除くすべてのコンポーネントをインストールするには、標準インストール オプションを選択します。

Silk Test Workbench の機能

Silk Test Workbench はテスト担当者の動作を模倣します。キー入力やマウス操作、メニューやリストからの項目の選択など、テスト担当者と同じようにアプリケーションを操作します。ただし、キー入力やマウス操作は、ハードウェアではなくソフトウェアによって自動的に生成されます。Silk Test Workbench は、アプリケーションが入力に対してどのように反応するか、画面を監視することによって、テスト操作を決定します。

予想結果と実際の結果が一致しないテスト条件にフラグを付けることができます。Silk Test Workbench のテストは柔軟です。スクリプトは、前の操作の結果に基づいて、実行すべき操作を決定できます。Silk Test Workbench は、システム クロックやハード ディスクなど、PC の内部リソースにもアクセスできるため、時間を考慮したり、ファイルの読み書きを行ったりもできます。

開始画面 と **ビジュアル ナビゲータ** は、テスト ソリューションを迅速に提供できるように設計されています。**開始画面** でテスト プロジェクトを管理し、**ビジュアル ナビゲータ** を使用して個々のテスト ステップを制御できます。

一元的マルチユーザー資産データベースは、テスト プロセスを管理するうえで効果的です。テスト実行に対するレポート作成機能も提供されます。

Silk Test Workbench は、動的オブジェクト解決を使用してテスト アプリケーションと対話します。動的オブジェクト解決により、オブジェクトを検索し識別する Silk Test Workbench クエリを使用した、テストの作成が可能になります。Silk Test Workbench は、ダイアログ ボックスやコントロールそれぞれに、固有に識別するためのロケーター名を割り当てます。Silk Test Workbench は、ロケーター名を使用してアプリケーション内のウィンドウ、ダイアログ ボックス、およびコントロールを検索します。同様に、Web ページにボタンなど多数のコントロールが含まれる場合があり、Silk Test Workbench はそれらにも固有のロケーター名を割り当てます。

Silk Test Workbench では、以下をテストできます。

- Silk Test とは異なるマシンで実行されるアプリケーション
- 2 台以上のマシンで同時に実行されるアプリケーション
- 複数の異なるアプリケーション (同時に)
- テスト アプリケーションのデータベース処理
- モバイル アプリケーション

Silk Test Workbench の UI

Silk Test Workbench のデスクトップは、テストにかかわるすべての操作の開始点になります。メニューバーを除き、デスクトップに表示されるウィンドウやペインをはじめとする各種機能の表示と外観を制御することができます。

Silk Test Workbench UI の主要な構成要素を以下に示します。

メニューバー Silk Test Workbench で利用できるすべてのメニューを含みます。利用可能なメニューとメニュー コマンドについての詳細は、「[メニュー](#)」を参照してください。

ツールバー ツールバーから、よく使用する操作にワンクリックでアクセスすることができます。ツールバーで利用可能なアクションについての詳細は、「[ツールバー](#)」を参照してください。

開始画面 **開始画面** は、Silk Test Workbench の機能を使い始める拠点であり、アプリケーションのテスト ソリューションをすばやく作成し始めることができます。詳細については、「[開始画面](#)」を参照してください。

アセットブラウザ **アセットブラウザ** を使用してテスト資産を管理します。**アセットブラウザ** では、データベース内の各種類の資産を一元的に作成、管理、表示できます。詳細については、「[アセットブラウザ](#)」を参照してください。

ビジュアル ナビゲータ **ビジュアル ナビゲータ** は、ビジュアル テストの要素をグラフィカルに表すものであり、ポイント アンド クリック インターフェイスを通して各要素を操作できます。**ビジュアル ナビゲータ** は、ビジュアル テストを作成、または開いたときに表示されます。詳細については、「[ビジュアル ナビゲータ](#)」を参照してください。

コード ウィンドウ **コード ウィンドウ** では、スクリプトの記録、設計、および変更を行います。**コード ウィンドウ** は、VB .NET スクリプトを作成、または開いたときに表示されます。詳細については、「[コード ウィンドウ](#)」を参照してください。

Silk Test Workbench ソフトウェア コンポーネント

Silk Test Workbench は、別々のプロセスとして実行される 2 種類のソフトウェア コンポーネントから構成されます。

- *Silk Test Workbench* ホスト ソフトウェア は、スクリプトとビジュアル テストの開発、編集、コンパイル、実行、およびデバッグに使用するコンポーネントです。このプログラムを実行するマシンは、ホスト マシンと呼ばれます。
- *Silk Test Agent* は Silk Test Workbench のコンポーネントで、アプリケーションの GUI と対話します。Agent はテスト内のコマンドを GUI 固有のコマンドに翻訳し、テストしているアプリケーションを実行、監視します。Agent は、ホストが実行されているのと同じマシン上でローカルに実行できます。また、ネットワーク環境で任意の数のエージェントをリモート マシン上で実行することもできます。Agent を実行するマシンは、リモート マシンと呼ばれます。

自動テストについて

ソフトウェアのテストは、ソフトウェア開発プロセスに欠かせない要素です。ソフトウェアをリリースするには、その前に、ソフトウェアが期待どおりに動作することを、さまざまな種類のコンピュータやオペレーティング システムで確認する必要があります。その結果、ソフトウェアのテストはますます複雑化し、その機能をさまざまなコンピュータで繰り返しテストすることが必要になっています。しかし、ソフトウェアのテストが複雑になる一方で、テストに割り当てられる時間は、リリース スケジュールを守るように短縮されているのが実態です。テストを自動化すると、ソフトウェアにバグやパフォーマンス上の問題がないかどうかを効果的かつ効率的に調べることができます。そのため、自動テストツールは、高品質ソフトウェアを開発するうえで欠かせないツールとなっています。

アプリケーションを手動でテストするには、一連のテストケースを作成して、特定の操作（一連のキーを押す、メニュー オプションを選択する、マウスを使用するなど）を実行したときのテスト アプリケーションの必要な反応を指定します。手動によるテスト サイクルは、通常、以下のようなパターンに従います。

- テスト アプリケーションをテスト開始点まで実行する
- テスト アプリケーションに入力を提供する一連の操作を実行する
- アプリケーションと対話しながら結果を監視する
- 対話の結果と予想を比較する
- 予想と結果の違いを記録する
- 必要に応じてアプリケーションを修正する
- アプリケーションを再テストするか、次のテストに進む

Silk Test Workbench では、このテスト パターンの各ステップを模倣できます。つまり、キーボード入力を行い、マウスの移動やマウス ボタンのクリックを行い、コンピュータに表示される内容を予想と比較し、テストの結果を分析できるようファイルに保存することができます。テストをリンクしておくと、1 つのテストが正しく終了したときに次のテストが開始されます。操作と検証が含まれるため、自動テストは手動によるテストより高速で、信頼性も高いものになります。

テスト戦略

ビジュアル テストやスクリプトなどの Silk Test Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロフィールの作成
- 注文の発注
- オンライン システムのログオフ

1 つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが 10 通りある場合、それらのシナリオを処理するために 10 個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが 1 つ追加された場合、新しいフィールドへのデータ入力を処理するために 10 個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテスト シナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

ドライバ スクリプトを実行すると、各テストが連続して呼び出されて実行されます。

モジュール式アプローチでは、在庫確認のテストを行う追加のテストを 1 つ作成し、それが適切な順序で呼び出されるようにドライバ スクリプトを更新できます。以下の例は、別々のスクリプトを呼び出してオンライン発注システムをテストするドライバ スクリプトを示しています。

```
Workbench.RunScript ("OnlineLogOn_Script")
Workbench.RunScript ("CreateProfile_Script")
Workbench.RunScript ("ValidateStock_Script")
Workbench.RunScript ("PlaceOrders_Script")
Workbench.RunScript ("OnlineLogOff_Script")
```

このようなモジュール式のテスト戦略を実施すると、テスト フローの変更も容易です。

たとえば、注文発注前の在庫確認など、オンライン発注システムのテストに追加のトランザクション単位が必要になった場合、10 個の異なるテスト シナリオがあれば、在庫確認を処理する 10 個のスクリプトを変更しなければなりません。

このようなモジュール式アプローチは、再利用性は向上しますが、毎回異なるデータを使用して一連のトランザクション単位を複数回繰り返すようなテスト戦略を要する、異なるテスト「シナリオ」に対応するには十分とは言えません。アクティブ データを使用すると、外部データ ソースを使用して、アプリケーション テストでのモジュール式テスト スクリプトの再利用を促進できます。

また、ソリューションをテストするためにドライバ スクリプトから関数を呼び出すこともできます。関数は、他のアプリケーション テストで再利用することができます。

別の場所にあるデータベースの利用について

分散環境で Silk Test Workbench を使用する場合、つまり、広域ネットワーク上で Silk Test Workbench クライアントから Silk Test Workbench の中央データベースにアクセスする場合、次の問題が発生する可能性があります。

- Silk Test Workbench クライアントとデータベース サーバー間での高レイテンシ。
- データベースへの多くのリクエストによるネットワーク トラフィックの増大。

このような分散環境では、Micro Focus では次のようなベスト プラクティスの適用をお勧めします。

- プロジェクト サイズを小さく保つ。これにより、**アセットブラウザ** の使用時などに、Silk Test Workbench クライアントとデータベース サーバー間でデータ転送量が増大するのを防ぎます。
- リモートでのテスト結果の表示を避ける。
- 可能な限り、中央データベースからプロジェクトをエクスポートしてオフラインで作業する。これを行うには、Silk Test Workbench ユーザーは、どの資産を読み取り専用として扱うか、どの資産を誰が更新するか、など調整する必要があります。さらに、中央データベースの資産をリモートのクライアントにインポートする必要があります。

- 仮想マシンやリモート デスクトップ接続を使用して、中央データベースにアクセスする。

Silk Test Workbench ドキュメント

Silk Test Workbench をマシンにインストールすると、(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > ドキュメント**、または (Microsoft Windows 10) **スタート > Silk** をクリックしてドキュメントにアクセスできるようになります。

Silk Test Workbench ドキュメント セットには、以下のものが含まれています。

Silk Test インストールガイド	Silk Test Workbench のシステム要件とインストール手順について説明しています。さらに、基本的なセットアップと Silk Test Workbench データベースの作成、構成、および接続方法についても詳しく説明しています。 <i>Silk Test</i> インストールガイドには、 Silk Test ドキュメント からオンラインでアクセスすることもできます。
Silk Test リリースノート	新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『 リリースノート 』を参照してください。
Silk Test Workbench ヘルプ	Silk Test Workbench 製品の使用方法に関する情報と Silk Test Workbench の言語リファレンスを提供します。
Silk Test チュートリアル	これらのチュートリアルには、特定のアプリケーション タイプを Silk Test でテストする方法や、Silk Test の特定の機能を使用する方法が記載されています。ローカルのインストール フォルダにあるチュートリアルのほか、最新のチュートリアルを Silk Test ドキュメント からダウンロードすることもできます。

Silk Test Workbench の新機能

Silk Test Workbench では、以下の新しい機能がサポートされています。

UI オートメーション サポート

Silk Test 19.5 では、新しいフォールバック サポートとして Microsoft UI オートメーション (UI オートメーション) のサポートを提供します。

UI オートメーション プロバイダー インターフェイスを実装する Windows ベースのアプリケーションを記録する場合、UI オートメーション サポートを有効にすると、コントロールをよりうまく解決できるようになります。UI オートメーション サポートは、Microsoft Windows 8 以降のマシン上でのみ使用できます。この機能は、次のテクノロジーを利用したアプリケーションに対するフォールバック サポートとして利用できます。

- WPF
- WinForms
- Oracle JavaFX*
- QT
- PowerBuilder
- Delphi
- Microsoft Office

* JavaFX アプリケーションは、自動的に検出されます。UI オートメーション サポートを有効化する必要はありません。

Silk Test Workbench での資産の比較

Silk Test Workbench 19.5 を使用すると、同じ種類の 2 つの異なる資産や、同じ資産の 2 つのバージョンを比較することができ、これらの間の相違点を簡単に調べられます。

次の資産の種類に対して、資産や資産のバージョンを比較できます。

- ビジュアル テスト
- .NET スクリプト
- オブジェクト マップ

ユーザビリティの改善

このセクションでは、Silk Test19.5 に対して行われたユーザビリティの改善点の一覧を提供します。

WebDriver ベースのブラウザーでのテキスト認識

Silk Test Workbench は、WebDriver ベースのブラウザーに対してテキスト認識をサポートするようになりました。

テキスト解決は次のメソッドを含みます。

- TextCapture
- TextClick
- TextExists
- TextRectangle

ネイティブ モバイル アプリ テスト時のメソッドの動的呼び出し

Silk Test クライアントの API を介して公開されていない Appium WebDriver のメソッドをサポートするために、Silk Test はモバイル デバイスに対するメソッドを動的に呼び出すことができるようになりました。

アクティブ データ ファイルのデータベースへの格納

Silk Test Workbench データベースにアクティブ データ ファイルを格納できるようになりました。

オブジェクト マップの利用状況の表示

Silk Test Workbench では、資産がどのオブジェクト マップを利用しているか、[インクルード] ビューに表示されるようになったため、オブジェクト マップの名前の変更時や編集時に、資産を更新しやすくなりました。

テスト実行の一時停止

Silk Test Workbench 上で、テストの再生中に **Alt + F11** を押して、テスト実行の現在実行中の操作を一時停止できるようになりました。

使用技術の更新

このセクションでは、Silk Test 19.5 に対して行われた重要な使用技術の更新をリストします。

Microsoft Windows の新しいバージョン


既存のテスト セットを利用して、Windows 10 October 2018 Update および Windows Server 2019 上で機能テスト、回帰テスト、ローカライズ テストを実行できるようになりました。

Mozilla Firefox の新しいバージョン

Silk Test の前のバージョンでテストされた Mozilla Firefox のバージョンに加えて、Silk Test は Mozilla Firefox の次の新しいバージョンでテストされました。

- Mozilla Firefox 61


- Mozilla Firefox 62
- Mozilla Firefox 63

 **注:** このリストには、Silk Test 19.5 のリリースまでに Silk Test 19.5 でテストされた Mozilla Firefox の新しいバージョンが含まれます。Mozilla Firefox の新しいバージョンが、Silk Test 19.5 のリリースより後にリリースされた場合でも、ブラウザ側で互換性が保たれていれば、Silk Test 19.5 は新しいバージョンを使用してテストできます。

Google Chrome の新しいバージョン

Silk Test の前のバージョンでテストされた Google Chrome のバージョンに加えて、Silk Test は Google Chrome の次のバージョンでテストされました。

- Google Chrome 68
- Google Chrome 69
- Google Chrome 70

 **注:** このリストには、Silk Test 19.5 のリリースまでに Silk Test 19.5 でテストされた Google Chrome のバージョンが含まれます。Google Chrome の新しいバージョンが Silk Test 19.5 のリリースより後にリリースされた場合でも、ブラウザ側で互換性が保たれていれば、Silk Test 19.5 は新しいバージョンを使用してテストできます。

Microsoft Edge の新しいバージョン

Silk Test の前のバージョンでテストされた Microsoft Edge のバージョンに加えて、Silk Test は、Microsoft Edge 44.17763 (Windows 10 October 2018 Update の Microsoft Edge) を使用して、記録と再生がテストされました。

Apple Safari の新しいバージョン

Silk Test の前のバージョンでテストされた Apple Safari のバージョンに加えて、Silk Test は Apple Safari 12 でもテストされました。

Android の新しいバージョン

Silk Test の前のバージョンでテストされた Android のバージョンに加えて、Silk Test は Android 9 でもテストされました。

iOS の新しいバージョン

Silk Test の前のバージョンでテストされた iOS のバージョンに加えて、Silk Test は iOS 12 でもテストされました。

macOS の新しいバージョン

既存のテストセットを利用して、macOS Mojave (10.14) 上の Apple Safari や iOS に対して機能テスト、回帰テスト、ローカライズテストを実行できるようになりました。

Java 11 のサポート

Silk Test Workbench は、Java 11 ベースのアプリケーションのテストをサポートするようになりました。

Java SWT の新しいバージョン

Silk Test は、SWT 4.8 と SWT 4.9 ベースのアプリケーションのテストをサポートするようになりました。


プロジェクトの管理

プロジェクトは資産のコレクションであり、各プロジェクトではアクセス権限を各ユーザーに割り当てます。Silk Test Workbench 環境の管理者が行うタスクの 1 つに、プロジェクトリストの管理があります。

資産を論理的に関連付けたプロジェクトとしてまとめて保管します。たとえば、特定のアプリケーションまたはビルドに関連付けられた資産をすべて一緒に、1つのプロジェクトに保管できます。

管理者は、プロジェクトを作成して、Silk Test Workbench データベースに追加できます。作成した新しいプロジェクトのアクセス権は、各ユーザーに設定された [デフォルトの権限] が割り当てられます。資産を別々のプロジェクトに保管すると、セキュリティのレベルを変えることができます。たとえば、プロジェクト 1 はフル アクセス権、プロジェクト 2 は読み取り専用、プロジェクト 3 はアクセス不可、のように設定できます。

現在のユーザーには、アクセス権を割り当てられているプロジェクトの資産だけが表示されます。資産は、**アセットブラウザ** を使用して表示および管理します。


 **注:** Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

Common プロジェクト

デフォルトのプロジェクトは Common プロジェクトです。これは、複数のプロジェクトで使用される資産の中央リポジトリとして機能します。Common プロジェクトの資産は、他のすべてのプロジェクトで使用できます。Common プロジェクトはデフォルトでアクティブ プロジェクトになっています。

アクティブ プロジェクト

新しく作成される資産は、アクティブ プロジェクトとして設定されているプロジェクトに追加されます。アクティブに追加された資産は、そのプロジェクト内と、アクティブ プロジェクトを参照しているすべてのプロジェクト内でのみ使用可能になります。

 **注:** プロジェクトの資産は、同じプロジェクトまたは参照プロジェクトの他の資産にのみアクセスできます。

プロジェクトの表示

アセットブラウザ では、**表示** ダイアログ ボックスで選択したプロジェクトとアクティブ プロジェクトに属するテスト資産がすべて表示されます。アクセスできるプロジェクトのテスト資産のみを表示できません。

デフォルトでは、新しい資産は常にアクティブ プロジェクトに追加され、そのプロジェクトでだけ使用できます。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **タスク** に移動して、**アセットブラウザ** をクリックします。**アセットブラウザ** が開きます。
2. アクティブ プロジェクトを変更するには、**アセットブラウザ** ツールバーの **表示** をクリックします。**プロジェクトの資産の表示** ダイアログ ボックスが開き、現在のユーザーがアクセスできるプロジェクトのリストが表示されます。
3. 表示する資産を持つプロジェクトのチェック ボックスをオンにし、**OK** をクリックします。**アセットブラウザ** には、選択したすべてのプロジェクトに含まれているテスト資産が表示されます。アクティブ プロジェクトの資産は常に表示されます。

ショートカット キー

標準の Windows メニュー コマンド以外に、多くのコマンドにキーの組み合わせが関連付けられており、キー入力によってコマンドを実行できます。通常、Ctrl または Alt キーを押しながら、該当するコマンドメニューの下線付きの文字を押します。

コマンドのメニュー ショートカット キー


ショートカット キー	実行される操作
Alt+F9	選択したオブジェクトを使用して、記録中にビジュアル テストまたはスクリプトに検証を挿入します。
Alt+F10	記録を開始または停止します。
Alt+Enter	ビジュアル テストまたはスクリプトから 要約情報 ダイアログ ボックスを表示します。
Ctrl+Alt	ビジュアル テスト (記録または再生中) またはスクリプト (記録中) に検証を挿入します。
Ctrl+F8	現在選択されているテスト ステップまでビジュアル テストを再生するか、または現在選択されているステートメントまでスクリプトを再生します。
Ctrl+F9	次のステートメントまたはポインタまで再生を実行します。デバッグ モードの場合に有効になります。
Ctrl+K	スクリプトで現在選択されている行 (複数) をコメントアウトして無効化します。コードブロック全体をコメントアウトするために、このショートカットを使用できます。また、 Ctrl+F11 を使用することもできます。
Ctrl+N	新規作成 ウィンドウを表示し、任意の種類 of 資産を作成することができます。
Ctrl+O	アセットブラウザ を開き、資産を開くことができます。前回作業した資産の種類が強調表示されます。
Ctrl+P	[印刷] ダイアログ ボックスを表示して、選択した資産を印刷できます。
Ctrl+S	<ul style="list-style-type: none"> 資産の現在の状態を資産の新しいバージョンとして保存し、増分の数値識別子をそのバージョンに割り当てます。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。 資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
Ctrl+Y	テスト スクリプトの直前の 元に戻す 操作を無効にします。ビジュアル テストでは利用できません。
Ctrl+Z	テスト スクリプトで直前に加えた編集内容を元に戻します。ビジュアル テストでは利用できません。
Ctrl+Alt+A	アセットブラウザ を開きます。
Ctrl+Alt+I	選択したオブジェクトのロケーターを記録できる オブジェクトの識別 ダイアログ ボックスを開きます。
Ctrl+Alt+S	開始画面 を表示します。
Ctrl+Shift+F2	前回の位置を表示します。
Ctrl+Shift+F8	別のビジュアル テストから再生しているビジュアル テスト内の残りのステップをすべて実行したあと、元のビジュアル テスト内の次のステップで再生が一時停止します。スクリプトの場合は、プロシージャ内の残りすべてのコードを 1 つのステートメントのように実行し、そのプロシージャを最初に呼び出したプロシージャ内の次のステートメントに移ります。
Ctrl+Shift+F9	すべての開いているビジュアル テストと VB .NET スクリプトのすべてのブレークポイントをクリアします。
Ctrl+Shift+K	スクリプトで現在選択されている行 (複数) のコメントを解除して有効化します。

ショートカット キー	実行される操作
Ctrl+Shift+L	<ul style="list-style-type: none"> 開いているすべての資産の現在の状態を資産の新しいバージョンとして保存し、各資産のバージョン識別子をインクリメントします。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。 開いているすべての資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
Ctrl+Tab	ユーザー インターフェイスで開いているウィンドウを切り替えます。
Ctrl+Space	単語を補完します。
Del	選択した資産を削除します。
F1	特定の項目のヘルプを表示します。
F2	プロパティ ペインで選択したパラメータの種類に応じて、 スクリプト入力パラメータの編集 または スクリプト出力パラメータの編集 ダイアログ ボックスを表示します。
F4	プロパティを表示します。
F5	アクティブなビジュアル テストまたはスクリプトを再生します。
F7	スクリプトの構文がコンパイルのすべての条件を満たしているかをチェックし、コードをコンパイルし、エラーが発生した場合は表示します。
F8	ビジュアル ステップの場合、テスト ステップを一度に 1 つずつ再生します。スクリプトの場合、ステートメントを一度に 1 つずつ再生します。
F9	選択したステップまたはコード行にブレークポイントを設定するか、または選択したステップまたはコード行に設定されているブレークポイントをクリアします。
Shift+F2	定義を表示します。
Shift+F8	埋め込まれているビジュアル テストまたはスクリプト全体を再生し、元のビジュアル テストまたはスクリプト内の次のステップで停止してデバッグ モードになります。VB .NET スクリプトで押下された場合は、関数呼び出しのステップ オーバーも行います。

チュートリアル

Silk Test Workbench チュートリアルでは、Silk Test Workbench を使用してアプリケーションをテストする基本的な方法をご自身のペースで学習することができます。また、チュートリアルでは、テストアプリケーションのさまざまな条件を処理するパワフルで柔軟なテストを作成したり、再生中に発生した詳細を正確にレポートしたりするための高度な機能もいくつか紹介しています。

チュートリアルではハンズオン デモが提供されており、Silk Test Workbench と直接対話しながら現実の世界のシナリオを使ってサンプル アプリケーションをテストします。

 **注:** Silk Test Workbench チュートリアルで使用されているサンプル アプリケーションは、Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行しないでください。

Silk Test Workbench ビジュアル テスト チュートリアル

Silk Test Workbench ビジュアル テスト チュートリアルによろこそ。ご自身のペースで学習できるこのチュートリアルでは、Silk Test Workbench のビジュアルでストーリーボードベースのインターフェイスを使用して、パワフルで柔軟性の高い機能テストを作成する方法について説明します。このチュートリアルでは、ビジュアル テストの作成、ビジュアル テストの再生、および再生結果の分析に必要な基本ステップについて学習します。また、多数の機能を使用して、記録したビジュアル テストをすばやく更新および拡張する方法についても学習します。

このチュートリアルでは、Silk Test Workbench を使用して繰り返し可能なテストを作成する練習をするための Silk Test サンプル Web アプリケーション <http://demo.borland.com/InsuranceWebExtJS/> を使用します。

各レッスンは前のレッスンの出力に基づいて進められるため、チュートリアルのレッスンは順番に完了してください。

GUI の概要

このセクションでは、メイン画面、開始画面、およびビジュアル ナビゲータを含めた GUI について説明します。このセクションはオプションです。すでに開発環境の基本要素をご存知の場合は、次のレッスンに進んでください。

Silk Test Workbench のメイン画面

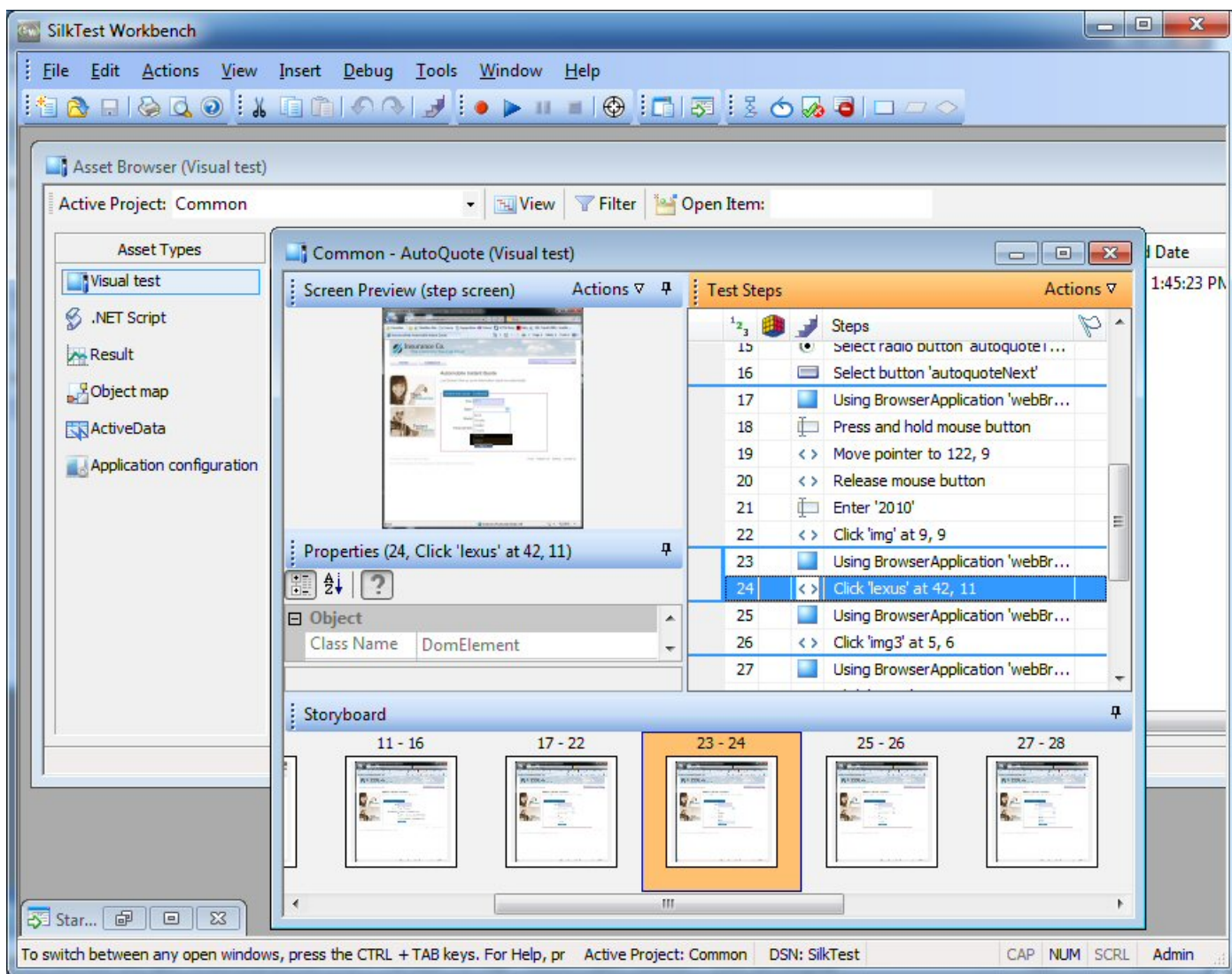
Silk Test Workbench のメイン画面は、テストにかかわるすべての操作の開始点になります。メニューバーを除き、デスクトップに表示される項目はすべてユーザーが制御できます。

開始画面、ビジュアル ナビゲータ、アセット ブラウザなど、Silk Test Workbench のその他すべてのウィンドウは、Silk Test Workbench のメイン画面の子ウィンドウとして表示されます。

Silk Test Workbench のメイン画面には以下の要素が含まれます。

- メニューバー
- ツールバー
- メインのテスト資産表示領域
- ステータスバー

以下の図は、Silk Test Workbench のメイン画面のこれらの要素を示しています。メインのテスト資産表示領域では、ビジュアル ナビゲータにビジュアル テストが表示され、背景にアセット ブラウザが表示され、開始画面が最小化されて表示されています。



開始画面

開始画面 は、Silk Test Workbench の機能を使い始める拠点であり、アプリケーションのテスト ソリューションをすばやく作成し始めることができます。

アプリケーションのテストをすばやく、最低限の手順で開始できるようにするため、**開始画面** には次の領域が含まれています。

領域 説明

開始 ビジュアルテスト、.NET スクリプト、キーワード駆動テストをワンクリックで記録したり、開いたりできます。

資産 最近使ったビジュアルテスト、.NET スクリプト、キーワード駆動テストをワンクリックで開くことができます。最近使った資産のリストは、リアルタイムで更新されるので、変更や更新により、現在の資産がリストの先頭に移動します。資産の右側にあるアイコンをクリックすることによって、資産を再生したり、資産の最後の実行結果を表示したりすることができます。



注: **開始画面** 設定 (**ツール > オプション > 開始画面**) では、最近使った資産のリストに表示する項目の数を制御できます。

領域 説明

新機能 Silk Test Workbench の現在のバージョンに含まれている新しい機能や機能強化を一覧します。Silk Test Workbench の更新が利用可能な場合も、ここに表示されます。



注: Silk Test Workbench の起動時に開始画面を表示しないように選択している場合、**ヘルプ > 製品更新の確認** をクリックして利用可能な更新を確認できます。

入門ガイド 生産性向上のため、広く使われるタスクや有用なヘルプ トピックへの直接リンクを提供します。

リソース サポート技術情報や Silk Test コミュニティなどの Silk Test Workbench に関連した情報へのリンクを提供します。

お知らせとコミュニティ Silk Test コミュニティの最新のフォーラム記事とブログ投稿へのリンクを提供します。

フラグ ビジュアル テストのテスト ステップ、およびフラグが付けられてフォローアップ対象として割り当てられた結果が表示されます。**フラグ** ペインを使用すると、テスト スイートやテスト プロジェクトについての情報を交換でき、データベースのテスト プロジェクトのほかのテスト担当者やユーザーと共同作業が行えます。

デフォルトでは、フラグは作成日および表示される資産で分類されます。フラグの情報をすばやく表示するには、マウスのポインタをフラグの上に移動します。フラグの説明、更新データ、プロジェクト、フラグが割り当てられているユーザーの ID を示すツール ヒントが表示されます。**開始画面のフラグ** オプション (**ツール > オプション > 開始画面 > フラグ**) を変更して、フラグの表示方法を変更できます。

フラグに関連付けられているビジュアル テストまたは結果を開くには、**フラグ** ペインでフラグをダブルクリックします。



注: **フラグ** ペインは、少なくとも 1 つのフラグがログインしているユーザーに割り当てられているか、少なくとも 1 つのフラグがフィルタで表示されるように設定されている場合にのみ、**開始画面** に表示されます。

開始画面 の右上隅にあるウィンドウの **閉じる** をクリックすると、**開始画面** を閉じることができます。



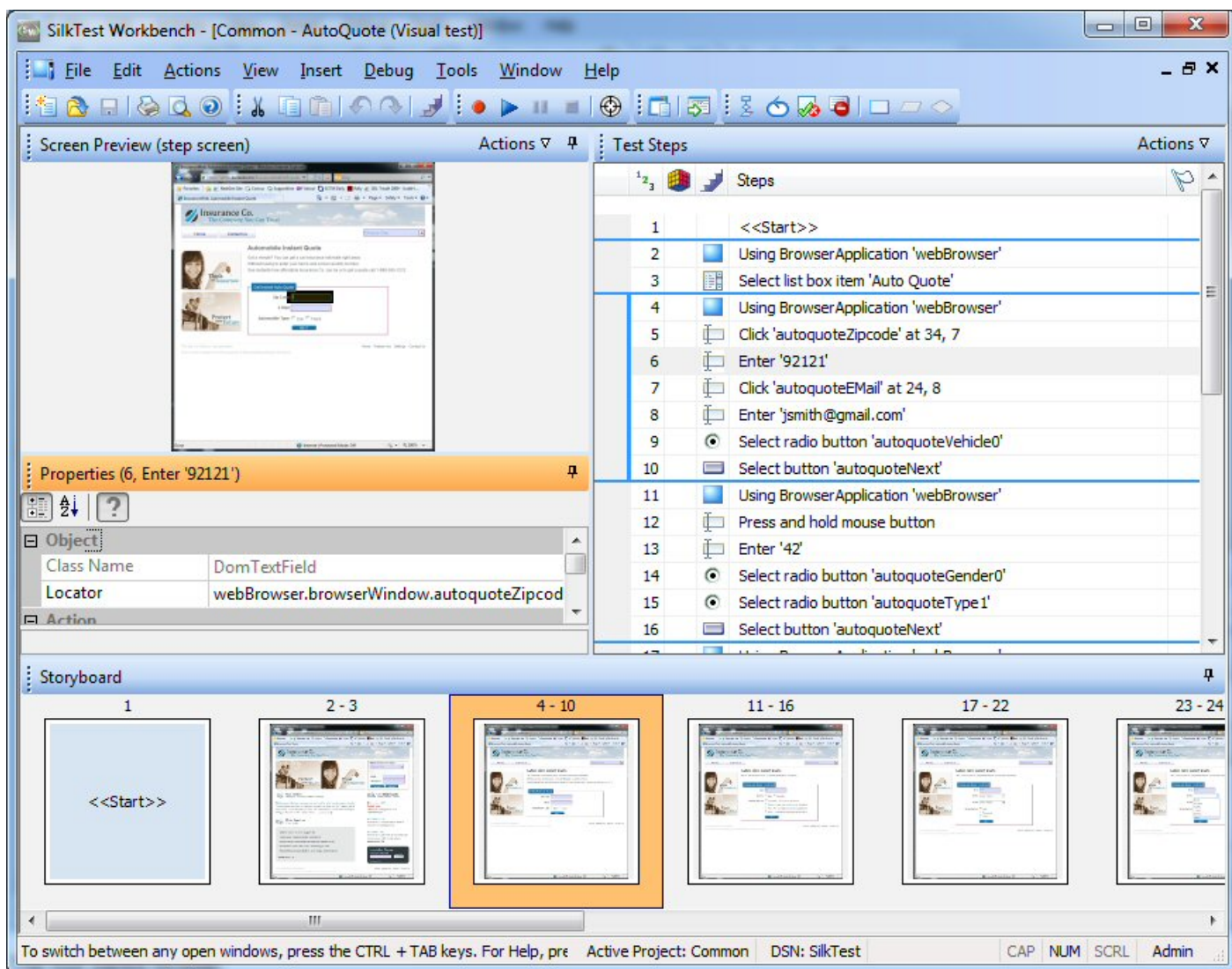
ヒント: ツール バーの **開始画面** をクリックすることで、メイン画面から **開始画面** にいつでもアクセスできます。また、いつでも Ctrl+Alt+S を押すか、または **表示 > 開始画面** を選択して、開始画面を表示できます。



注: ユーザー インターフェイスは、Windows 標準の大きいフォントや小さいフォントを使用します。カスタムのフォント サイズを使用すると、ユーザー インターフェイスのテキスト表示が不正確になる可能性があります。

ビジュアル ナビゲータ

ビジュアル ナビゲータ は、ビジュアル テストの要素をグラフィカルに表すものであり、ポイント アンド クリック インターフェイスを通して各要素を操作できます。**ビジュアル ナビゲータ** では、ビジュアル テストに関する情報が 4 つのペインに表示されます。4 つのペイン全体でビジュアル テストの各ステップを総合的に把握できます。



4つのペインは以下のとおりです。

テストステップ ビジュアルテストの各ステップが技術的ではない明確な言語で示されます。

画面プレビュー ビジュアルテストの再生中に実行されるステップに応じてテスト対象アプリケーションのスナップショットが表示されます。


プロパティ ビジュアルテストのステップのプロパティが表示されます。

ストーリーボード サムネイルイメージを使用してビジュアルテストのフローが表示されます。サムネイルは、ビジュアルテストのステップの論理グループを表します。

 **注:**

Silk Test Workbench では、スナップショットは以下のような状況のときに取得されます。

- 記録中のすべての自動テストステップの前。

 **注:** SAP アプリケーションの場合、スナップショットはすべての自動テストステップの前ではなく、画面が変わったときに発生します。

- ビジュアルテストで Using ステップを実行した結果のスナップショット。

- 再生エラーが発生した場合。


画面プレビュー、**ストーリーボード**、および**プロパティ** ペインは **テスト ステップ** ペインと同期され、**テスト ステップ** ペインで選択したステップに関する情報が表示されます。このように、**テスト ステップ** ペインでステップを選択し、他のペインでそのステップに関する情報を確認することで、ステップのすべての側面を簡単に把握できます。

ビジュアル ナビゲータ では、ビジュアル テストを表示するだけでなく、**画面プレビュー** ペインと**プロパティ** ペインを使用して既存のビジュアル テストを拡張または更新することもできます。たとえば、ビジュアル テストを記録したあと、**プロパティ** ペインで、記録したプロパティのリテラル値を変数に置き換えることができます。また、テスト対象アプリケーションに変更が発生したときにビジュアル テストをすばやく更新する場合は、**画面プレビュー** の **画面の更新** 機能を使用して、以前にキャプチャした画面を更新できます。

ビジュアル ナビゲータ では、ビジュアル テストに使用すると同じペインを使用して、ビジュアル テストの再生結果も表示します。結果の場合、これらのペインには機能が付加され、**結果** ウィンドウ内に表示されます。ウィンドウにはツールバー オプションと複数のタブが表示されます。各タブには結果のさまざまな内容が表示されます。結果に固有の追加機能として、たとえば、**テスト ステップ** ペインに各ステップのステータス（合格または失敗）を表示する機能があります。さらに、**画面プレビュー** では、記録中にキャプチャされた画面と再生中にキャプチャされた画面の違いを比較し、テスト アプリケーションにアクセスすることなく、既存のビジュアル テストを更新できます。

ビジュアル テストの記録開始画面

サンプル Web アプリケーションで保険の見積もりを作成するための操作を行うと、Silk Test Workbench はそれらの操作をステップとして記録します。ステップはビジュアル テストの基礎となります。テストに必要な操作の記録を完了すると、ビジュアル ナビゲータに記録されたテストが表示されます。ステップは、ビジュアル ナビゲータの [テスト ステップ] ペインに表示されます。

 **注:** このチュートリアルで使用されているサンプル アプリケーションは、Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Micro Focus は Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行することをお勧めしません。


 **注:** Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザ アドオンを無効にします。Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

サンプル Web アプリケーションの起動

このチュートリアルでは、Silk Test サンプル Web アプリケーションを使用します。この Web アプリケーションは、デモ用に提供されています。

Internet Explorer で、Silk Test サンプル Web アプリケーションを使用します。チュートリアルのレッスンと同じユーザー体験を得るには、Internet Explorer 以外のサポート対象ブラウザでサンプル Web アプリケーションを実行しないでください。

1. テストの速度と信頼性を高めるために DOM 関数を記録するには、以下のステップを実行します。
 - a) **ツール > オプション** をクリックします。
 - b) **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
 - c) **xBrowser** をクリックします。
 - d) **ネイティブなユーザー入力を記録する** リスト ボックスから、**いいえ** を選択します。
 - e) **OK** をクリックします。

 **注:** 通常、Web アプリケーションをテストする場合、DOM 関数ではなく、ユーザーの入力そのものを使用します。Flash および Java アプレットなどのプラグインや、AJAX を使用するアプリケ

ーションは、ネイティブなユーザー入力ではサポートされますが、高レベルの API 記録ではサポートされません。

2. Web アプリケーションの記録または再生を行う前に、ブラウザのアドオンをすべて無効にする必要があります。ブラウザのすべてのアドオンを確実に無効にするには、以下のステップを実行します。
 - a) Internet Explorer で **ツール > インターネット オプション** を選択します。 **インターネット オプション** ダイアログ ボックスが開きます。
 - b) **プログラム** タブをクリックし、 **アドオンの管理** をクリックします。 **アドオンの管理** ダイアログ ボックスが開きます。
 - c) アドオンのリストの **状態** 列で、各アドオンの状態が **無効** になっていることを確認します。
状態 列に **有効** と表示されている場合は、アドオンを選択してから **無効にする** をクリックします。
 - d) **閉じる** をクリックし、 **OK** をクリックします。
3. サンプル アプリケーションにリモートでアクセスするには、 <http://demo.borland.com/InsuranceWebExtJS> をクリックします。サンプル アプリケーションの Web ページが開きます。

サンプル Web アプリケーションに対するビジュアル テストの記録

記録中は、記録を停止するまで、テスト アプリケーションでのすべての操作 (SilkTest Workbench の操作は除く) が Silk Test Workbench によって記録されます。記録が完了したら、生成したビジュアル テストを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. このチュートリアルの後半で記録したコントロールを編集できるようにするために、ビジュアル テストを記録する前にブラウザ コントロールのキャプチャを有効化します。
 - a) Silk Test Workbench メニューで、 **ツール > オプション** を選択します。 **オプション** ダイアログ ボックスが開きます。
 - b) ツリーから **記録 > 出力 > ビジュアル テスト** を選択します。
 - c) **ブラウザ コントロールのキャプチャ** の値を **はい** に設定します。



注: Web アプリケーションの記録中にコントロールのキャプチャを有効化すると、ビジュアル テストのパフォーマンスが低下する恐れがあります。Micro Focus では、このチュートリアルで述べるタスクを完了したら、コントロールのキャプチャを無効化することをお勧めします。

2. **ファイル > 新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。
3. 資産の種類のリストから **ビジュアル テスト** を選択します。
4. 記録をすぐに開始するために、 **記録の開始** チェック ボックスをオンにします。
5. **OK** をクリックして、ビジュアル テストを資産として保存し、記録を開始します。 **アプリケーションの選択** ダイアログ ボックスが開きます。
6. **Web** タブを選択します。
7. リストから **Internet Explorer** を選択します。
8. **移動する URL の入力** テキスト ボックスに、 demo.borland.com/InsuranceWebExtJS/ を入力します。
9. **OK** をクリックします。Silk Test Workbench が最小化され、 **記録中** ダイアログ ボックスが開きます。
10. Insurance Company Web サイトでは、次のステップのいずれかを行います：
 - a) **Select a Service or login** リスト ボックスから **Auto Quote** を選択します。 **Automobile Instant Quote** ページが開きます。
 - b) 郵便番号と電子メール アドレスを適切なテキスト ボックスに入力し、自動車タイプをクリックして、 **Next** をクリックします。
たとえば、郵便番号に 92121、電子メール アドレスに jsmith@gmail.com をそれぞれ入力し、自動車タイプとして Car を指定します。
 - c) 年齢を指定し、性別と運転履歴タイプをクリックして、 **Next** をクリックします。
たとえば、年齢に 42 を入力し、性別と運転履歴タイプに Male および Good をそれぞれ指定します。
 - d) 製造年、車種、モデルを指定し、財務情報タイプをクリックして、 **Next** をクリックします。

たとえば、製造年に 2010 と入力し、車種とモデルに Lexus および RX400 をそれぞれ指定し、財務情報タイプとして Lease を指定します。
指定した情報の概要が現れます。

e) **Purchase** をクリックします。

Purchase A Quote ページが開きます。

f) ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。

11 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバーアイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、ビジュアル テストは **テスト ステップ** ペインに表示されます。

ビジュアル テストの保存と名前の指定

テストを記録すると、**記録完了** ダイアログ ボックスが開き、以下の操作を実行できます。

- 記録したビジュアル テストを再生する
- 記録した操作をビジュアル ナビゲータで確認する
- ビジュアル テストを保存し、記録したテストをビジュアル ナビゲータで確認する

ビジュアル テストをはじめて記録したばかりなので、テストを再生したり、記録した操作を確認したりする前にテストを保存し、名前を付けます。

1. **記録完了** ダイアログ ボックスで、**保存** をクリックします。



ヒント: 名前を指定せずに資産を作成すると、Silk Test Workbench により「無題_」に連番の付いた一時的な名前が割り当てられます。

テストにまだ名前を付けていないため、**名前を付けて保存** ダイアログ ボックスが開きます。

2. **名前** テキスト ボックスで、名前を AutoQuote に変更します。

3. **説明** テキスト ボックスに Visual test tutorial と入力します。

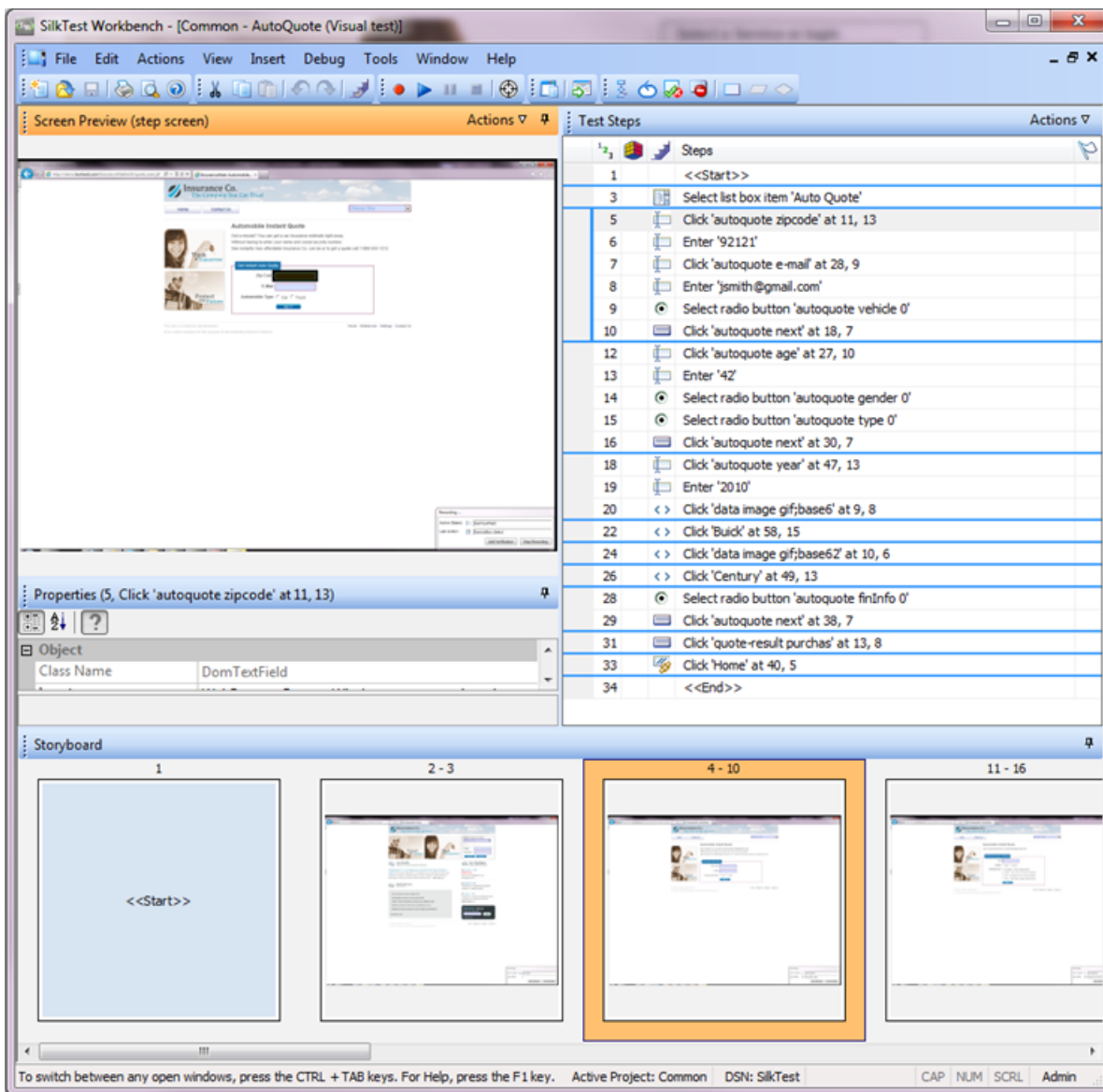
4. **OK** をクリックします。ビジュアル ナビゲータにビジュアル テストが表示されます。

記録されたテスト ステップの確認





ビジュアル テストを記録し、保存すると、ビジュアル テストがビジュアル ナビゲータに表示されます。ビジュアル ナビゲータには 4 つのペインがあり、ビジュアル テストの包括的なビューが提供されます。


[テスト ステップ] ペインのステップは、Web アプリケーションで見積もりを完了するときにアクセスした画面と実行した操作を表します。


記録されたステップは、以下の図に示すステップと同じようになるはずですが。



[テスト ステップ] ペインには以下の列が表示されます。

列	説明
 (番号) *	ステップが記録され、再生される順番を表します。デフォルトでは、ステップはこの順番で表示されます。 ステップにデバッグのためのブレークポイントが追加されている場合は、この列の隣の列にブレークポイントが表示されます。 デフォルトのビュー（ステップと画面）から画面のみのビューまたはステップのみのビューにビューを変更した場合、番号付けの方法では、記録の順序で間隔が示されるようにステップまたは画面は順番にスキップされます。
 (ロジック) *	ステップにロジックが含まれている場合、ステップのロジックのタイプを表すアイコンを表示します。
 (ステップ タイプ) *	ステップで実行される操作のタイプを表すアイコンを表示します。
ステップ	ステップで実行される操作について説明します。
 (フラグ) *	割り当てられたフラグ のアイコンを表示します。

列	説明
 (ステップの説明) *	<p>割り当てられたフラグ のアイコンは、関連付けられたステップが割り当て先のユーザーの テストステップ ペインおよびオプションで 開始画面 ペインに表示されるように、フラグが付けられていることを示します。さらに、ポインタをアイコンの上に移動すると、フラグの説明、更新日、および割り当て日を示すツールヒントを表示できます。</p> <p>ユーザー定義のステップの説明を表示します。この列はデフォルトのビューには表示されません。テストステップ ペインのタイトルバーから 操作 をクリックし、次に 表示 > ステップの説明 をクリックすると表示できます。</p> <p>ステップの説明を作成するには、ステップを選択します。プロパティ ペインで、ステップの説明 プロパティを更新します。ステップの説明 アイコンがこの列に表示され、ステップに説明があることが示されます。ポインタをアイコンの上に移動して説明を含むツールヒントを表示するか、ステップを選択してプロパティ ペインの説明を読みます。</p>

 **ヒント:** ビジュアルナビゲータの他のペインは、**テストステップ** ペインと同期しています。上の図では、記録したステップのうち、リストボックスから **Auto Quote** を選択するステップが **テストステップ** ペインで選択されています。結果は以下のようになります。

- **画面プレビュー** には、**Auto Quote** を選択する前のアプリケーションの状態が示されます。
- **プロパティ** ペインには、選択されたステップのプロパティが示されます。
- ストーリーボードでは、**Auto Quote** リスト項目の選択に関連するステップのグループを表すサムネイルが強調表示されます。

ビジュアルテストのステップをスクロールし、さまざまなステップを選択すると、他のペインに更新した情報が表示されます。

記録されたビジュアルテストの再生

ビジュアルテストを記録し、保存したら、再生して、ビジュアルテストが適切に動作することを確認できます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログボックスが開きます。このダイアログボックスで、結果をどのように保存するかを指定できます。

2. **結果の説明** テキストボックスに Initial test results for the recorded test と入力します。

3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

各結果は、一意のテスト実行番号によって識別されます。

Silk Test Workbench が最小化され、ビジュアルテストが再生されます。再生中、サンプルアプリケーションアプリケーションに対して、ビジュアルテストの記録中に行った操作が画面上で再生されます。再生が正常に完了すると、**再生完了** ダイアログボックスが開きます。

結果の分析：概要

テストを再生すると、Silk Test Workbench によってテスト結果が生成されます。テスト結果にはビジュアルテストの再生に関する情報が含まれています。ビジュアルテストの名前、実行番号、各ステップが実行された日付と時刻、各ステップの合格/失敗のステータスなどの重要な情報が含まれます。

[結果] ウィンドウのタブの使用

テスト結果情報をすばやく参照するために、**結果** ウィンドウには、結果情報を種類別に分けて表示するフィルタの役目を果たす 5 つのタブがあります。

1. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。**要約** タブには、再生が成功したかどうか、最新の実行番号、合格または失敗となったチェックまたは検証の数、テストの開始時刻と終了時刻、テストの実行結果に関するその他の基本情報などを含むテスト実行の概要が示されます。

2. **詳細** タブをクリックします。

詳細 タブでは、ビジュアル ナビゲータの以下の 4 つのペインを使用して各ステップの結果が示されます。

テスト ステップ ビジュアル テストの各ステップの再生結果に関する情報がリストされます。

画面プレビュー 再生中にキャプチャされた Web アプリケーションの画面が表示されます。

プロパティ ステップのプロパティが表示されます。

ストーリーボード 結果のアウトラインと概要がグラフィカルに示されます。



注: 合格、失敗、フラグ タブでもビジュアル ナビゲータを使用して結果情報が示されます。違いは、**詳細** タブではすべてのステップが示されるのに対し、これらのタブでは特定の種類のステップが示されるという点だけです。

3. 任意のステップを選択します。

ステップを選択すると、Silk Test Workbench により他のペインの情報がそのステップの情報に更新されます。**画面プレビュー** では、再生中にキャプチャされた画面が、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較されます。**プロパティ** ペインには、選択したステップのプロパティが示されます。**ストーリーボード** では、ステップが属するグループが強調表示されます。



ヒント: ステップの名前全体を表示するには、**ステップ** 列を広げるか、ステップの上にマウス ポインタを置いて、名前全体を示すツール ヒントを表示する必要がある場合があります。

[結果] ウィンドウのツールバーの使用

結果 ウィンドウのツールバーには、結果の内容の表示と種類をカスタマイズするためのいくつかのオプションがあります。

1. 結果のツールバーの **詳細表示** をクリックします。Silk Test Workbench により、**テスト ステップ** ペインに追加の列が表示され、**プロパティ** ペインに追加のプロパティが表示されます。
2. **テスト ステップ** ペインで、右にスクロールして追加の列を表示します。
これらの列には、ステップの実行に要したミリ秒単位の時間、テストを実行した人のユーザー名など、各ステップに関する特定の情報が示されます。
3. **プロパティ** ペインの [結果] プロパティ カテゴリには、**テスト ステップ** ペインの各列に対応するプロパティが示されます。下にスクロールして、リスト全体を表示します。
4. **表示** ドロップダウン矢印をクリックし、**ステップのみ** を選択します。Silk Test Workbench によりすべての画面が除外されるので、ステップだけをすばやく参照できます。
5. **表示** ドロップダウン矢印をクリックし、**ステップと画面** を選択します。Silk Test Workbench により結果の画面とステップがすべて表示されます。

結果 ウィンドウのツールバーには、この他にも結果の内容の表示と種類をカスタマイズするためのオプションがあります。

[プロパティ] ペインの使用

プロパティ ペインには、ステップの基本特性を示すプロパティが表示されます。ステップの名前、実行ステータス、再生の詳細などの情報が示されます。

1. ステップ「'42' を入力します」をクリックします。

このステップは、**Age** テキスト ボックスに値「42」を入力する操作を実行します。

プロパティ ペインが更新され、選択したステップのプロパティが表示されます。

2. **カテゴリ** アイコンまだ選択されていない場合は、これをクリックします。

プロパティが以下のメイン カテゴリにグループ分けされます。

結果 **テスト ステップ** ペインの列に対応するプロパティが含まれます。ステップの名前、ステップ実行の日付と時刻、ステップ実行に要した時間などのプロパティです。

拡張プロパティ Silk Test Workbench が、入力 操作ステップによって入力されたテキストのコントロールまたは値を識別するために使用するロケーター名など、追加の再生の詳細が含まれます。拡張プロパティは、操作ステップで変数や式が使用されたときにその内容を表示するのに便利です。たとえば、入力 ステップで変数 *textVar* が値として使用された場合、[拡張プロパティ] カテゴリに *textVar* の内容が表示されます。

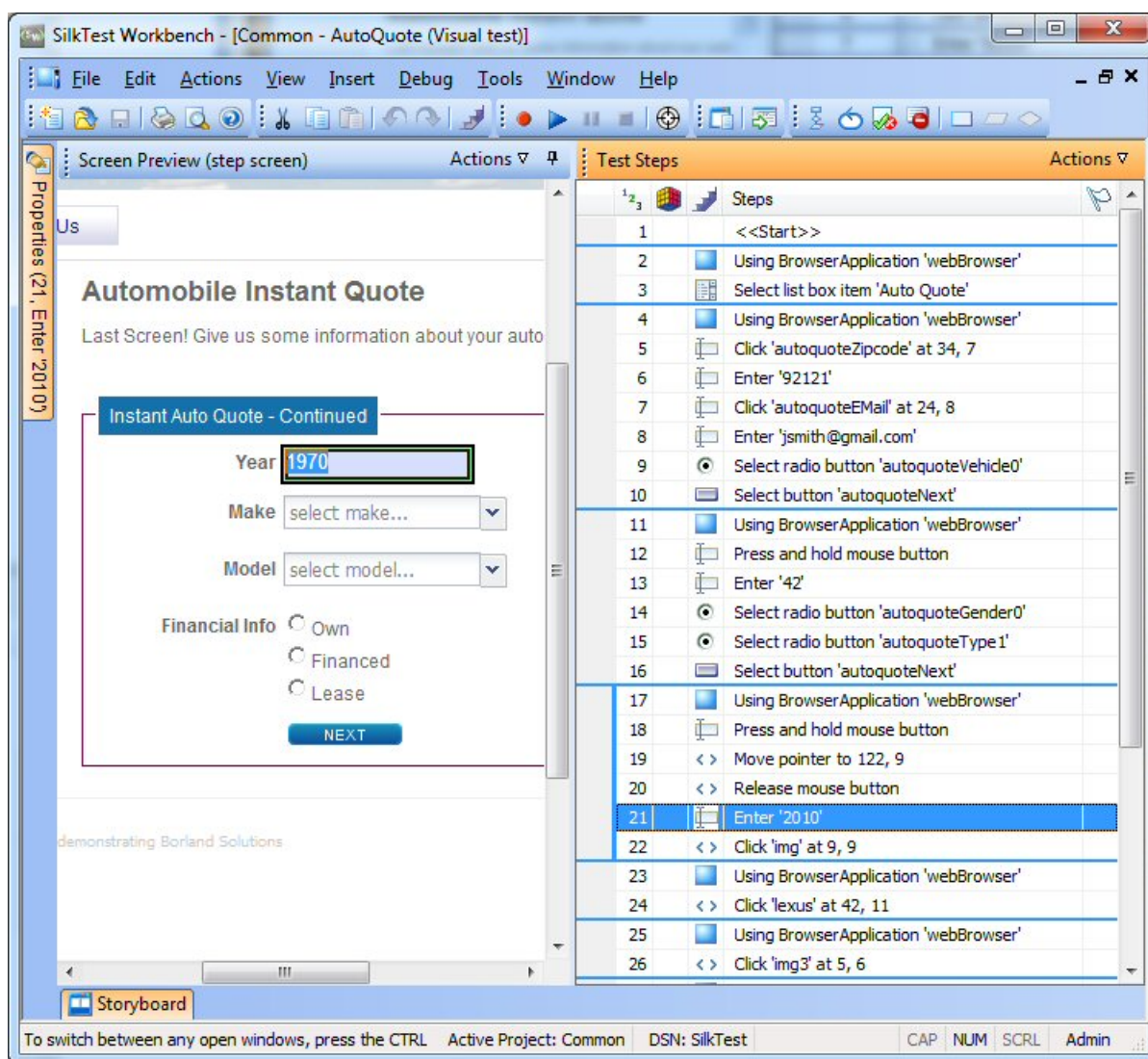
3. **拡張プロパティ** ノードを展開します。Text プロパティおよびロケーター プロパティがその値と共に示されます。Text プロパティの値は '42' です。これは、このステップで **Age** テキスト ボックスに入力された値です。

4. **再生前のビジュアル テストのステップ プロパティを表示/非表示** アイコンをクリックします。[ビジュアル テストの詳細]ノードが表示され、ビジュアル テストに存在するステップの元のプロパティがすべて表示されます。

画面プレビューの使用

画面プレビュー には、**テスト ステップ** ペインに表示された、コントロールと対話する各ステップについてキャプチャされたテスト アプリケーションのイメージが表示されます。キャプチャされたイメージとして、デスクトップ全体、アプリケーション ウィンドウ、またはアクティブなウィンドウのみを示すことができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

1. **テスト ステップ** ペインで、車の製造年を含むステップを選択します。「2010」と入力します。**画面プレビュー** に、再生中にキャプチャされた画面とビジュアル テストの記録中にキャプチャされた画面が並んで表示されます。このステップのコントロールであるブラウザ ウィンドウが黒のボックスで強調表示されます。



2. 画面プレビューで、操作 > 相違点の表示 > オフ をクリックします。ビジュアルテストの画面が消え、再生の画面が開きます。
3. 操作 > ズーム > 75% をクリックしたあと、スクロールバーを使用して製造年がはっきりと表示されるまで移動します。
4. ビジュアルテストに切り替えます。テストステップペインで、操作 > ビジュアルテストのウィンドウ をクリックします。

ビジュアルテストの拡張：概要

ビジュアルテストの拡張には、既存のビジュアルテストが新しいバージョンのテストアプリケーションで使用できるようにテストを更新することなどが含まれます。たとえば、テストアプリケーションのさまざまな状態を処理し、検証するために、テストロジックを挿入することができます。さらに、ビジュアルテストの内容をわかりやすくしたり、重要な点について自分自身や他のユーザーに注意を促すために、フラグ、メッセージボックス、またはステップの説明を挿入することができます。

これらは、Silk Test Workbench で既存のビジュアルテストを拡張し、よりパワフルで堅牢かつ柔軟なテストを作成する、ほんの数例です。

画面プレビューからの更新

Silk Test Workbench はビジュアルテストを記録するとき、テストアプリケーションの各画面の関連するコントロールだけでなく、画面もキャプチャします。Silk Test Workbench により、画面プレビューには

キャプチャされた各画面が表示され、ビジュアルテストのステップによって識別されるコントロールが強調表示されます。**画面プレビュー**で、キャプチャされた画面の別のコントロールを識別することにより、テストアプリケーションにアクセスすることなくステップを更新できます。


このレッスンでは、Silk Test Workbench の **画面プレビューからのコントロールの挿入** 機能を使用して、**画面プレビュー** のキャプチャされた画面から別のボタンを選択します。

1. AutoQuote ビジュアルテストの **テストステップ** ペインで、<<終了>> ステップの前にあるステップを選択します。

ステップテキストは、次のようになります：(40, 5) の場所で 'Home' をクリックします

画面プレビュー に **Purchase A Quote** ページが表示され、**Home** ボタンが強調表示されます。

2. **画面プレビュー** で **操作 > ズーム > 75%** をクリックし、スクロールバーを使用して **Contact Us** ボタンがはっきり表示されるまで移動します。
3. **プロパティ** ペインで、**ロケータ** テキスト ボックスをクリックします。ロケータの値領域に、ロケータ選択ボタンが表示されます。
4. **プロパティ** ペインで、**画面プレビューから識別** をクリックします。

 **注:** ボタンが無効の場合、ビジュアルテストの記録を行う前にブラウザコントロールのキャプチャを有効にしていなかったことが原因です。

ポインタが **画面プレビュー** に移動します。

5. ポインタを **Contact Us** ボタンの上に移動し、**Contact Us** ボタンをクリックします。

ボタンをクリックすると、ボタンの新しいロケータ名を表示するように **プロパティ** ペインの **ロケータ** プロパティが変わり、**テストステップ** ペインのステップテキストが以下のように変わります。
(40, 5) の場所で 'Contact Us' をクリックします

6. **保存** をクリックします。

次にこのビジュアルテストを再生すると、Silk Test Workbench により **Home** ボタンではなく **Contact Us** ボタンがクリックされます。

検証の挿入

検証とは、ユーザー定義の条件を評価し、合格/失敗のメッセージや、オプションとしてフラグをビジュアルテストの再生結果に送信するテストロジックです。

このレッスンでは、検証を挿入して、見積もりで正しい車両モデルが使用されていることを確認します。

1. **表示 > テストステップ > ステップと画面** をクリックして、ステップと画面の両方が表示されていることを確認します。
2. 次のステップを選択します：「(91,11) の場所で 'rx400' をクリックします」
3. **画面プレビュー** で **操作 > ズーム > 75%** をクリックし、スクロールバーを使用して **Model** がはっきり表示されるまで移動します。
4. 以下のいずれかのステップを実行します。
 - ツールバーの **検証タイプのロジックの作成** をクリックします。
 - **挿入 > テストロジック > 検証** を選択します。

テストロジックデザイナー ウィザードの **ようこそ** ページが開きます。

5. **次へ** をクリックします。**ロジックの種類**の選択 ページが開きます。
6. **コントロールのプロパティ** をクリックし、**次へ** をクリックします。**プロパティに基づく条件の定義** ページが開きます。
7. **画面プレビューから識別** をクリックします。ポインタが **画面プレビュー** に移動します。

8. **Model** コンボ ボックスを選択します。**プロパティに基づく条件の定義** ページに、選択したコントロールの名前が、コントロールのプロパティと共に表示されます。コントロール名は、以下のように表示されます。

```
webBrowser.browserWindow.modelCombo
```

9. **プロパティの選択** グリッドで、**テキスト** にスクロールして選択します。

10 **条件の選択** が **等しい** に設定されていることを確認します。

11 **プロパティの選択** から Text プロパティを選択して、**期待値** テキスト ボックスに **RX400** が入力されていることを確認します。

12 **次へ** をクリックします。**検証の構築** ページが開きます。このページから、再生結果に送信する合格/失敗メッセージを定義できます。ページの上部に Silk Test Workbench が検証する条件が以下のように表示されます。

```
If "webBrowser.browserWindow.modelCombo"."Text" 等しい "RX400"
```

この条件は検証のロジックを定義しています。条件では、モデル タイプと選択されたタイプが比較されます。

13 デフォルトの合格の説明を The model type is correct に置き換え、デフォルトの失敗の説明を The model type is NOT correct に置き換えます。

14 **次へ** をクリックします。**要約** ページが開きます。

検証ステップは選択したステップのあとに挿入されます。検証ステップのテキストは以下のようになります。

```
「If "webBrowser.browserWindow.modelCombo"."Text" 等しい "RX400"」を検証します
```

15 検証の内容を確認したら、**完了** をクリックします。

サンプル アプリケーションのプロパティの値を検証するテスト ロジックを挿入することで、記録したビジュアル テストを拡張できました。

アプリケーション データを保存するローカル変数の作成

変数を使用すると、テストの別の場所や別のビジュアル テストまたはスクリプトで使用するデータ値を保存できるため、ビジュアル テストを拡張できます。データは他の種類のファイルに出力することもできます。

このレッスンでは、チュートリアルのアートのレッスンで使用できるように、電子メール アドレスを表示するコントロールのテキストを保存します。このためには、まずテキストを保存するローカル変数を作成する必要があります。

1. **テストステップ** ペインで、**操作 > 挿入 > 変数 > ローカルの追加** をクリックします。**ローカル変数の追加** ダイアログ ボックスが開きます。

2. **変数名** テキスト ボックスに strEmailAddress と入力します。

3. **型** リストから **テキスト** を選択します。

このレッスンでは、**初期値** テキスト ボックスを空のままにします。これは、後続のレッスンで変数に値を保存するためです。

テキスト 型では、変数値がテキスト、つまり文字列データ型として保存されます。

4. **OK** をクリックします。ビジュアル テストに対して新しい変数が保存されます。変数を作成したあとは、<<開始>> ステップからそれを参照し、定義を編集することができます。

5. **strEmailAddress** 変数を表示するには、**テストステップ** ペインで <<開始>> ステップを選択します。



ヒント: <<開始>> ステップは、すべてのビジュアル テストで常に最初のステップになります。

ステップのプロパティが **プロパティ** ペインに表示されます。<<開始>> ステップを選択している場合、**変数** カテゴリに **strEmailAddress** が **テキスト** 変数として表示されます。

各変数型の値領域は、現在定義されている各変数型の数を示します。このレッスンではテキスト変数を 1 つ作成したので、その種類の値領域には 1 項目がこのテストに関連付けられていることが示されています。

見積りの電子メール アドレスを保存するローカル変数を作成したので、アプリケーションの電子メール アドレスを変数に保存します。

ローカル変数へのアプリケーション データの格納

サンプル アプリケーションの **Get Instant Auto Quote page** ページには、一意の電子メール アドレスが表示されます。このページの電子メール アドレスを含むテキストは、このページのコントロールのプロパティ値です。

このレッスンでは、前のレッスンで作成したローカル変数 *strEmailAddress* にこのテキストを保存します。

- 1. テスト ステップ** ペインで、電子メール アドレスの値を表示するステップを選択します。
ステップ テキストは次のようになります : 'jsmith@gmail.com' を入力します

ステップを選択すると、**Get Instant Auto Quote page** ページのキャプチャされた画面が **画面プレビュー** に表示されます。
- 2. テスト ステップ** ペインで、**操作 > 挿入 > コントロールのプロパティ** をクリックします。
これにより、ビジュアル テストの選択したステップの直後にステップが挿入されます。ステップ テキストは次のようになります : 「コントロールの "" プロパティを取得します」

このステップのプロパティを編集して、**Get Instant Auto Quote** ページの電子メール アドレスのテキストをローカル変数に保存します。使用するコントロール、そのコントロールのプロパティ、およびプロパティの値を保存する変数を指定するように、このステップのプロパティを編集します。変数は、前のレッスンで作成した *strEmailAddress* です。
- 3. 画面プレビュー** で **操作 > ズーム > 50%** をクリックし、スクロールバーを使用して電子メール アドレスのテキストがはっきり表示されるまで移動します。
- 4. プロパティ** ペインで、**ローケター** テキスト ボックスをクリックします。ローケターの値領域に、ローケター選択ボタンが表示されます。
- 5. 画面プレビューから識別** をクリックします。ポインタが **画面プレビュー** に移動します。
- 6.** ポインタをページの **E-Mail** テキストの上に移動します。
強調表示のボックスがページのテキストを囲んでいることを確認し、強調表示された領域をクリックしてコントロールを識別します。

Silk Test Workbench により、**プロパティ** ペインの **ローケター** 情報が次の値に更新されます :
webBrowser.browserWindow.autoquoteEMail
- 7. プロパティ** ペインで **プロパティ** テキスト ボックスをクリックし、その値リストから **Text** を選択します。
- 8. (ローカル変数の選択...)** をクリックし、リストから **strEmailAddress** を選択します。

これで、コントロール、必要な値を含むコントロールのプロパティ、およびプロパティの値を保存する変数の指定が終了したので、ステップのテキストは次のようになるはず : 「コントロールの 'Text' プロパティを変数 'strEmailAddress' に格納します」

プロパティ ペインとステップ テキストの両方の **ローカル変数名** の値が *strEmailAddress* に変わります。

テストでプロパティ値が正しくキャプチャされ、保存されることを確認するために、テストを再生して結果を確認します。

拡張したビジュアル テストの再生と分析

記録したテストにいくつかの拡張を行ったので、ビジュアル テストを再生し、結果を分析してみます。

- 次のいずれか 1 つのステップを行います :
 - **操作 > 再生** を選択します。
 - ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに、Enhanced test results for the recorded visual test と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。Silk Test Workbench によって、拡張したテストが再生されます。
5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

要約 タブに、ビジュアル テストに合格した (エラーなしで正常に再生されたことを意味します)、または失敗した検証が表示されます。

6. **合格 (1)** タブをクリックします。
カッコ内の数は、合格した検証の合計数を示します。**テスト ステップ** ペインには検証ステップが表示され、**結果の詳細** 列には検証の合格の説明が表示されます。
7. **詳細** タブをクリックして、各ステップの結果を表示します。
8. **結果** ウィンドウの **テスト ステップ** ペインで、変数に電子メール アドレスを保存した結果を示すステップまでスクロールします。ステップ テキストは、次のようになります: 「コントロールの 'Text' プロパティを変数 'strEmailAddress' に格納します」
9. *strEmailAddress* 変数の内容が **結果の詳細** 列に表示されます。**Text** プロパティの値全体を表示するには、カーソルをそのステップの **結果の詳細** 列の上に置きます。ツールヒントにプロパティの内容全体が表示されます。

お疲れ様でした。これで、サンプル アプリケーションを確実にテストするビジュアル テストを作成できました。次のレッスンでは、ビジュアル テストを他のビジュアル テストですばやく簡単に実行する方法など、より高度なテストの概念と機能について学習します。

ビジュアル テスト内でのビジュアル テストの実行 : 概要

このチュートリアルでは、Web アプリケーションから自動車保険の見積もりを取得するために必要な各操作を実行する単一のビジュアル テストを作成しました。単一のビジュアル テストは、単純なアプリケーションに対して基本的なテスト ケースを実行する場合に便利です。ただし、ほとんどのソフトウェア テストでは、アプリケーションのすべての側面をテストするなど、より厳格なアプローチが必要となります。さらに、テスト アプリケーションが変化したときに既存のビジュアル テストをすばやく更新することが必要です。

これらのテストの課題を解決する効率的な手段を提供するために、Silk Test Workbench はモジュール式テストをサポートしており、特定のテスト ソリューションの共通の操作を単一のテストにまとめ、同じ操作セットを必要とする他のビジュアル テストでビジュアル テストを再利用できるようにしています。

モジュール式テスト

ビジュアル テストやスクリプトなどの Silk Test Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロファイルの作成

- 注文の発注
- オンライン システムのログオフ

1つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが10通りある場合、それらのシナリオを処理するために10個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが1つ追加された場合、新しいフィールドへのデータ入力を処理するために10個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを1つ作成して、それをシナリオごとに10個作成するよりも、これらのトランザクション単位を1つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテスト シナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

モジュール式テストの基本が理解できたので、2つめのテストを作成し、前のレッスンで作成したテストに追加してみます。

2つめのビジュアル テストの記録

このセクションでは、2つめのビジュアル テストを記録し、ビジュアル テスト資産の別の作成方法を学習します。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **ビジュアル テスト** を選択し、**資産名** テキスト ボックスにビジュアル テストの名前を入力します。
このチュートリアルでは、名前として AddAccount と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、ビジュアル テストを資産として保存し、記録を開始します。**テストの構成** ダイアログ ボックスが開きます。
5. **Web** タブを選択します。
6. リストから **Internet Explorer** を選択します。
7. サンプル アプリケーションの **Home** ページで、**Login** セクションの **Sign Up** をクリックします。**Create A New Account** ページが開きます。
8. 適切なフィールドに、以下の情報を入力します。
次のフィールドに移動するには Tab キーを押します。


フィールド名	値
名 (First Name)	Pat
姓 (Last Name)	Smith
生年月日 (Birthday)	12/02/1990
	 注: カレンダー コントロールの月と年の隣にある下矢印をクリックして月と年を変更し、カレンダーで 12 を選択します。
電子メール アドレス (E-mail Address)	smith@test.com
住所 (Mailing Address)	1212 Test Way
市 (City)	San Diego
州 (State)	CA
郵便番号 (Postal Code)	92121
パスワード (Password)	test

9. **Sign Up** をクリックします。
- 10 **Continue** をクリックします。連絡先情報が表示されます。
- 11 ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。
- 12 **Log Out** をクリックします。
- 13 **Alt+F10** を押して記録を完了します。**記録完了** ダイアログ ボックスが開きます。
- 14 **保存** をクリックします。ビジュアル ナビゲータにビジュアル テストが表示されます。


別のビジュアル テストへのビジュアル テストの挿入

このセクションでは、ユーザー アカウントを追加する 2 つめのビジュアル テストを、元のビジュアル テストの自動見積もりの要求を行うステップの前に挿入します。

ビジュアル テスト内でビジュアル テストを実行すると、複数のビジュアル テストで効率的に同じ基本ステップをテストできます。


 **ヒント:** ビジュアル テストを別のビジュアル テストに挿入する際、テスト アプリケーションが正しい初期再生状態になっていることを確認することが重要です。

1. **開始画面** の **最新** リストで、**AutoQuote** をクリックして開きます。
AutoQuote はこのチュートリアルで最初に作成したビジュアル テストです。
2. **テスト ステップ** ペインで <<開始>> ステップを選択します。
3. **操作 > 挿入 > ビジュアル テスト** をクリックします。**ビジュアル テストの参照** ダイアログ ボックスが開きます。
4. **資産の選択** リストから **AddAccount** という名前のビジュアル テストを選択し、**OK** をクリックします。

 **ヒント:** **資産の選択** リストに多くのビジュアル テストが表示されている場合、**名前フィルタ** テキスト フィールドに **AddAccount** を入力してから、ビジュアル テストをクリックすると簡単です。

Silk Test Workbench によって、選択したステップの前にステップが挿入されます。挿入されたステップによって、選択したビジュアル テストが呼び出されます。ステップ テキストは、次のようになります：

ビジュアル テスト 'AddAccount' の再生

 **ヒント:** 再生中に上記のステップが実行されると、挿入されたビジュアル テストが再生される前に、元のビジュアル テストが最後まで再生されます。

次のレッスンでは、このモジュール式ビジュアル テストを再生し、再生エラーに対処する方法を学習します。

再生エラーへの対応：概要

再生中のエラーは、テスト アプリケーションの変更、ビジュアル テストの不適切なステップ フローなど、さまざまな要因により発生します。このようなエラーをデバッグ機能を使用してすばやく診断し、修正することにより、テストの保守を最小限にし、チームによるテスト作業の効率を上げることができます。

まず、前のレッスンで作成したモジュール式テストを再生します。

モジュール式テストの再生

前のレッスンでは、ビジュアル テスト AddAccount を別のビジュアル テスト AutoQuote に挿入することにより、モジュール式テストを作成しました。

このセクションでは、このモジュール式テストを再生し、再生中のエラーを確認します。

1. AutoQuote ビジュアル テストが開いている状態で、ツールバーの **再生** をクリックします。**再生** ダイアログ ボックスが開きます。
2. **結果の説明** テキスト ボックスに Responding to errors in a modular test と入力します。

3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

再生中に、**Create A New Account** ページでテストが停止し、エラー メッセージが表示されます。データベースでは、各顧客レコードに対して一意の電子メール アドレスが必要なため、このエラーが発生します。AddAccount ビジュアル テストの記録中にすでに電子メール アドレスを入力しているため、この電子メール アドレスはデータベースにすでに存在しており、テストが失敗します。

再生エラーを確認したので、テストのデバッグを開始します。

エラーのデバッグ

Silk Test Workbench で再生エラーが発生すると、Silk Test Workbench の **再生エラー** ダイアログ ボックスが表示され、デバッグ モードを開始するオプションが提供されます。デバッグ モードでは再生が一時停止され、Silk Test Workbench のデバッグ機能を使用して再生エラーを診断し、修正することができます。

このセクションでは、前のセクションでモジュール式テストを再生しているときに発生したエラーをデバッグしてみます。

1. **再生エラー** ダイアログ ボックスで、**デバッグ** をクリックします。

デバッグ モードでは再生が一時停止されます。そのため、エラーを発生させたステップのプロパティを編集したり、ステップを削除または無効にしたり、他のビジュアル テストからステップをコピー アンド ペーストしたりしてエラーを修正してから、再生を再開できます。

Silk Test Workbench はデバッグ モードに入り、AddAccount ビジュアル テストを表示します。エラーを発生させたステップが黄色で強調表示されます。

2. **編集 > 有効化/無効化** を選択してステップ「'signup continue' をクリックします」を無効化します。

このステップを無効にすると、次回テストを再生するときにエラーを回避できます。

ステップのテキストがグレーの斜体になります。これはステップが無効にされたことを示します。

3. ツールバーで **再生** をクリックします。 **再生エラー** ダイアログ ボックスが表示されます。

4. **デバッグ** をクリックします。

5. **編集 > 有効化/無効化** を選択してステップ「'logout-form logout' をクリックします」を無効化します。

テストでは Home ページで **Log Out** ボタンを探しますが、Sign Up Continue ステップが無効になっておりボタンがページに表示されないために、このエラーが発生します。このエラーは、チュートリアルの方で修正します。

6. メニューから **デバッグ** を選択します。

メニューの上部に表示される以下のステップ コマンドを使って、ステップ実行を制御できます。

ステップ イン (F8) 1 つずつステップを再生します。**ステップ イン** は各ステップのトレースに有効であり、また再生しているビジュアル テストに挿入された別のビジュアル テストやスクリプトにもステップ インします。挿入された各ビジュアル テストやスクリプトについても、一度に 1 ステップまたは 1 行ずつ実行されます。

ステップ イン はテストの詳細分析に有効であり、変数の使用やターゲット アプリケーションとの対話に対する各ステップの影響を確認することができます。

ステップ オーバー (Shift +F8) 別のビジュアル テストに挿入されたビジュアル テスト全体またはスクリプト全体を、1 つのステップとして実行します。**ステップ オーバー** は、デバッグ モードの再生中に、ビジュアル テストまたはスクリプトを 1 ステップとして再生する場合に使用します。これにより、挿入されたビジュアル テスト全体またはスクリプト全体が再生されます。挿入されたビジュアル テスト全体またはスクリプト全体が再生されると、元のビジュアル テスト内の次のステップで再生が停止し、デバッグ モードになります。

別のビジュアル テストまたはスクリプトを再生するステップ以外のステップで **ステップ オーバー** を使用すると、**ステップ イン** を使用した場合と同じになります。次のステップのみが実行されたあと、再生は停止し、再びデバッグ モードになります。

ステップ アウト (Ctrl + Shift + F8) 別のビジュアル テストから再生しているビジュアル テストまたはスクリプト内の残りのステップをすべて実行したあと、元のビジュアル テスト内の次のステップで再生が一時停止します。

ステップ アウト は、別のビジュアル テストに挿入されたビジュアル テスト内のステップまたはコード行で再生が停止されているときに、残りのビジュアル テストを再生して元のビジュアル テストに戻る場合に使用します。挿入されたビジュアル テストまたはスクリプトの残りの部分の再生が実行されたら、元のビジュアル テスト内の次の行で再生が停止し、再びデバッグ モードになります。

カーソルまで実行 (Ctrl + F8) 再生を一時停止するステップを選択できます。これにより、ビジュアル テストまたはスクリプト内の選択したセクションを「ステップ オーバー」することができます。

カーソルまで実行 は、ビジュアル テストまたはスクリプトを再生して、実行時エラーが発生したポイントの直前で停止する場合に使用します。これにより、ブレークポイントを挿入することなく特定の行またはステートメントで再生を停止できます。再生が停止したら、別のデバッグ オプションのいずれかを使用して再生を継続できます。

カーソルから実行 現在選択されているテスト ステップからビジュアル テストを再生します。

7. ステップ アウト を選択します。

このコマンドにより、AddAccount ビジュアル テストの残りのステップが実行され、その後、AutoQuote ビジュアル テストの次のステップで再生が一時停止されます。

ステップ アウト を選択すると、Silk Test Workbench により AddAccount の最後のステップとしてホームページが表示され、再生が一時停止します。そして、Silk Test Workbench により AutoQuote がデバッグ モードで開かれ、次のステップが強調表示されます。

次に、ビジュアル テストで使用されている変数の値を監視する方法について学習します。

再生中の変数の追跡

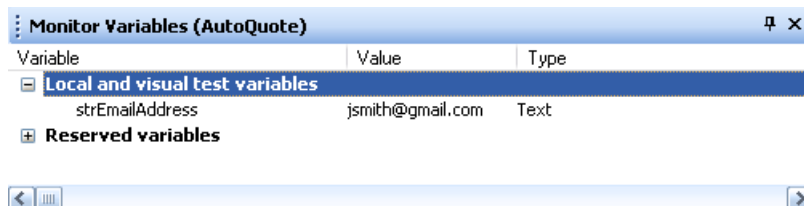
デバッグ モードでは、ローカル変数 ウィンドウを使用して、ローカル変数と予約変数の両方の再生値を追跡できます。

このセクションでは、ローカル変数 ウィンドウを使用して、前のレッスンで作成したローカル変数の値を参照します。


1. **デバッグ > ローカル変数** を選択します。ローカル変数 ウィンドウが開きます。

2. **ローカルおよびビジュアル テスト変数** ノードを展開します。

前のレッスンで作成したローカル変数 `strEmailAddress` が表示されます。再生中、この変数を使用するステップが実行されると、変数の値が **値** 列に表示されます。結果は、以下の図のようになります。



Variable	Value	Type
Local and visual test variables		
strEmailAddress	jsmith@gmail.com	Text
Reserved variables		

 **注:** 変数の結果を ローカル変数 ウィンドウで確認する場合にテストでエラーがない場合は、ブレークポイントをビジュアル テストに挿入します。ブレークポイントを挿入するには、variable ス

トップのあとにあるステップを選択し、**デバッグ > ブレークポイントの設定/クリア** を選択してからテストを再生します。

3. **カーソルまで実行** デバッグの動作を確認するには、「(24, 9) の場所で 'Lexus' をクリックします」ステップを選択し、**デバッグ > カーソルまで実行** を選択します。Silk Test Workbench は、残りのステップを「(24, 9) の場所で 'Lexus' をクリックします」ステップまで再生したあと、AutoQuote ビジュアル テストを表示します。

次に、モジュール式テストの再生を完了し、テスト結果を確認します。

結果の確認

ビジュアル テストのデバッグが終了したら、ビジュアル テストの結果を確認します。

1. ツールバーの **再生** をクリックして、AutoQuote ビジュアル テストの再生を完了します。**再生完了** ダイアログ ボックスが開きます。
2. **結果へ移動** をクリックします。AutoQuote の結果が表示され、デフォルトで **要約** タブが表示されます。

要約 タブにはテスト実行の全体的な詳細が表示されます。**ビジュアル テストまたは .NET スクリプト (実行回数)** フィールドには、「AutoQuote(1)」と挿入されたビジュアル テスト「AddAccount(1)」がリストされます。

3. **詳細** タブをクリックします。
4. **テスト ステップ** ペインで、青色のテキストのステップまでスクロールします。

結果 列と **結果の詳細** 列を確認すると、再生中に発生したエラーに関する情報をすばやく参照できます。



注: テスト ステップ ペインにすべてのステップを表示するには、[ステップと画面] が選択されていることを確認します。**操作 > 表示 > ステップと画面** をクリックします。**失敗** タブには、再生エラーを含むステップは表示されません。失敗した検証のみが表示されます。

再生エラーを診断し、デバッグする方法を学びました。

アクティブ データの使用

アプリケーションの使用を効率よく模倣するために、アプリケーション テストでは異なるデータ セットを使用して同じ操作を繰り返すことがよく行われます。たとえば、前のレッスンでは顧客レコードを作成するテストを記録しました。10 件の顧客レコードを作成するには、テストごとに顧客データ セットを用意し、10 個の異なるテストを記録できます。ただし、Silk Test Workbench では元のテストを拡張し、実行ごとに異なるデータ セットを使用してテストを 10 回繰り返すことができます。

アクティブ データ テストでは、外部ファイルのデータをテスト アプリケーションへの入力として使用し、実行ごとに異なるデータを使用して、指定したステップを繰り返し実行することができます。

このレッスンでは、以下の操作方法を学びます。

- アクティブ データ テスト資産を作成し、ビジュアル テストに関連付ける
- 実行ごとに異なるデータを使用し、指定したステップを特定の回数実行する繰り返しロジックを作成する
- ビジュアル テストでアクティブ データ ファイルを使用する方法を定義する
- 繰り返し実行するステップを定義する
- データ ファイルのアクティブ データをビジュアル テストのリテラル データにマップする

前のレッスンでは、Pat Smith の顧客情報を顧客データベースに入力するビジュアル テストを記録しました。このビジュアル テストを再生するたびに、最初の記録中にキャプチャしたリテラル データ値が Silk Test Workbench で使用されます。

このレッスンでは、Pat Smith の顧客情報を入力するために使用されるリテラル データ値をアクティブ データに置き換え、Silk Test Workbench がビジュアル テストを再生するときに外部ファイルに含まれる別の顧客情報が使用されるようになります。

ビジュアルテスト用のアクティブ データを作成する前に、アクティブ データ ファイルについて確認します。

アクティブ データ ファイルの確認

アクティブ データ資産を作成する際、使用するデータが含まれる既存のデータ ファイルを選択するか、資産によって使用されるデータを入れる新しいファイルを作成することができます。このチュートリアルでは、customers.csv という名前の、顧客情報を含むカンマ区切りの既存のファイルを使用します。このファイルは、Silk Test Workbench のインストール ディレクトリの Examples フォルダにあります。

customers.csv ファイル内の各列は、Age 列を除き、データベースに顧客情報を入力するために使用されるフィールドに対応しています。Age 列は、このチュートリアルのあとのレッスンで使用します。

このテストにアクティブ データを使用すると、Silk Test Workbench はテストを再生し、ファイル内の各顧客がデータベースに入力されるまで、毎回別の顧客のデータを使って顧客情報を入力するステップを繰り返します。



ヒント: アクティブ データ ファイルでは、パスワード値はコード化されたテキストになっています。

アクティブ データ ファイルには以下の列が含まれます。

- 名 (First Name)
- 姓 (Last Name)
- 生年月日 (Birthday)
- 電子メール アドレス (Email Address)
- 住所 (Mailing Address)
- 市 (City)
- 州 (State)
- 郵便番号 (Postal Code)
- パスワード (Password)
- 年齢 (Age) (後続のレッスンで使用)

次に、アクティブ データ テスト資産を作成し、ビジュアルテストに関連付けます。

アクティブ データ テスト資産の作成

アクティブ データ テストでデータ ファイルを使用するには、そのファイルを使用するアクティブ データ資産を作成し、そのアクティブ データ資産をビジュアルテストに関連付ける必要があります。

1. 開始画面の **最新** リストで、**AddAccount** というビジュアルテストをダブルクリックして開きます。ビジュアルナビゲータに AddAccount ビジュアルテストが表示されます。
2. **テストステップ** ペインで、**操作 > 挿入 > アクティブ データ > 新規作成** をクリックします。**アクティブ データ資産の設定** ウィンドウが表示されます。
3. **名前** テキストボックスにアクティブ データ資産の名前を customers と入力します。
4. **参照** をクリックして、アクティブ データ資産と関連付ける customers.csv ファイルを探します。**アクティブ データ資産の選択** ダイアログボックスが表示されます。
5. サンプル ファイルの場所に移動し、アクティブ データのサンプル ファイル customers.csv を選択します。
デフォルトの場所は、次のようになります : C:\Program Files\Silk\Silk Test\examples\customers.csv
6. **開く** をクリックします。**全般** タブの **ファイル** テキストボックスにパスとファイル名が表示されます。
7. **アクティブ データ資産の設定** ウィンドウで、**オプション** タブをクリックします。
8. **最初の行を見出しとして使用** チェックボックスをオンにします。



ヒント: **詳細** タブをクリックすると、アクティブ データ ファイルの内容を確認することができます。

この設定により、アクティブ データ ファイルのデータの最初の行が、データではなく見出し行として扱われます。

9. **保存して閉じる** をクリックしてアクティブ データ資産を作成し、AddAccount ビジュアル テストに関連付けます。

ビジュアル テストでは、関連付けられているアクティブ データ資産の情報はすべて、<<開始>> ステップの **プロパティ** ペインに格納されます。ビジュアル テストに関連付けられている各アクティブ データ資産を確認するには、<<開始>> ステップを選択し、**プロパティ** ペインの **アクティブ データ** プロパティを確認します。

これで、アクティブ データ資産が作成され、ビジュアル テストと関連付けられました。次に、テストでアクティブ データを使用する繰り返しロジックを作成します。

アクティブ データ ファイルの繰り返しロジックの作成

ビジュアル テストでアクティブ データを使用する場合、通常は一連のステップを繰り返し、その際毎回リテラル データをアクティブ データに置き換えます。

このレッスンでは、データベースに顧客を追加するリテラル データを、10 件の顧客レコードを含むアクティブ データ ファイル customers.csv 内のデータに置き換えます。繰り返しごとに、アクティブ データ ファイルの顧客レコードのデータが使用されて、InsuranceCo Web サイトの **Create a New Account** ページの各フィールドにデータが入力されます。

1. ビジュアル ナビゲータに AddAccount ビジュアル テストが表示されていることを確認します。
2. Silk Test Workbench のツールバーの **繰り返しタイプのロジックの作成** をクリックします。 **ようこそ** ページが表示された **テスト ロジック デザイナ** ウィザードが表示されます。
3. **次へ** をクリックします。 **ロジックの種類を選択** ページが開きます。
4. **アクティブ データ ファイルを使用してステップ シーケンスを繰り返す** をクリックし、**次へ** をクリックします。 **使用するアクティブ データ資産の定義** ページが表示されます。
5. このビジュアル テストに関連付けられているアクティブ データ資産は 1 つだけ (customers) なので、**アクティブ データの資産** リストにこの資産が表示されます。そうでない場合は、リストからこの資産を選択します。
6. 省略可能： **シート名** リストからデータ ファイル中の使用するシートを選択します。デフォルトでは、アクティブ データ資産で指定されているシートが使用されます。
7. **開始行** テキスト ボックスで、デフォルト値の **1** をそのままにします。
8. **データを含む最後の行で終了** チェック ボックスをオンにします。
9. **順序通りにすべての行を取得** をクリックします。

これで、繰り返しロジックでデータ ファイルを使用する方法、および繰り返し回数が定義できました。

- 繰り返しロジックの実行時、アクティブ データ ファイルの最初の行が最初に使用されます。
- 各行を使って、最後の行まで繰り返されます。
- 繰り返しロジックではすべてのデータ行が順番に使用され、アクティブ データ ファイルにはデータ行が 10 行あるため、データベースに顧客を追加するステップは 10 回繰り返されます。毎回、アクティブ データ ファイルの異なる顧客データ行が入力として使用されます。

10 **次へ** をクリックします。 **繰り返しの構築** ページが開きます。

アクティブ データの使用方法が定義できたので、次にビジュアル テスト内の繰り返すステップを指定します。

繰り返しステップの定義

繰り返しロジックで繰り返すステップを決定するときは、リテラル データを置き換えるステップだけでなく、繰り返しが必要なすべての操作を考慮します。

このレッスンでは、リテラル データを置き換えるステップがいくつかありますが、アクティブ データ ファイル内の各顧客を入力するには、InsuranceCo Web サイトの **Create a New Account** ページにアクセス

するプロセス全体を繰り返す必要があります。そのため、ページにアクセスし **Home** ページにテストを戻すステップも含め、実際のテストプロセスのすべてのステップを繰り返す必要があります。

1. **繰り返しの構築** ページで **ステップの開始** リンクをクリックします。**ステップの選択** ダイアログ ボックスが開きます。
2. InsuranceCo Web サイトとの対話を開始するステップを選択します。これは <<開始>> ステップのすぐあとのステップです。
ビジュアル テストには顧客を追加するプロセスに必要なステップがすべて含まれるため、すべてのステップを繰り返す必要があります。
3. **OK** をクリックします。**ステップの選択** ダイアログ ボックスが閉じられ、選択したステップのテキストが **ステップの開始** リンクに表示されます。
4. **繰り返しの構築** ページで **ステップの終了** リンクをクリックします。**ステップの選択** ダイアログ ボックスが開きます。
5. InsuranceCo Web サイトとの対話を終了するステップを選択します。これは <<終了>> ステップのすぐ前のステップです。
6. **OK** をクリックします。**ステップの選択** ダイアログ ボックスが閉じられ、選択したステップのテキストが **ステップの終了** リンクに表示されます。
7. **次へ** をクリックします。**テスト ロジック デザイナー** の **要約** ページが表示されます。このページには、ビジュアル テストに定義されたテスト ロジックが表示されます。
8. **終了** をクリックして、ビジュアル テストのテスト ステップにテスト ロジックを挿入します。

Silk Test Workbench により、ビジュアル テストの <<開始>> ステップのあとに、繰り返しロジックを開始するステップが挿入されます。ステップのテキストは次のようになります:「アクティブ データ 'customers' を使用して繰り返します」

また、Silk Test Workbench により、<<終了>> ステップのあとに、繰り返しロジックを終了するステップも挿入されます。ステップのテキストは次のようになります:

繰り返しの終了

新しく挿入された繰り返しロジック ステップの間のすべてのステップがインデントされ、繰り返しロジック ステップの間にネストされて表示されます。ビジュアル テストでアクティブ データを使用するための最後の作業は、アクティブ データ ファイル内のデータをビジュアル テストのリテラル データにマップすることです。

アクティブ データのリテラル データへのマッピング

ビジュアル テストでアクティブ データ ファイル内のデータを使用するには、該当するテスト ステップのデータをアクティブ データ ファイル内のデータ列にマップする必要があります。

1. ビジュアル テストで、フィールドにデータを入力する最初のステップを選択します。
これは、ビジュアル テストで最初に記録された、InsuranceCo Web サイトの **Create a New Account** ページの **First Name** テキスト ボックスに「Pat」と入力するステップとなります。
ステップのプロパティが **プロパティ ペイン** に表示されます。
2. **プロパティ ペイン** の **パラメーター** カテゴリで、**text** パラメーターを選択します。これは、**Pat** に設定されたテキストを示します。
3. 値領域の **選択** をクリックし、**アクティブ データ** をクリックします。



ヒント: **customers** が **アクティブ データの資産** リストにない場合、この資産はこのビジュアル テストに関連付けられていません。

アクティブ データ列の選択 ダイアログ ボックスが開きます。このダイアログ ボックスで、テスト ステップの選択したプロパティのデータをアクティブ データ ファイル内の列のデータにマップします。

アクティブ データの資産 テキスト ボックスには、ビジュアル テストに関連付けられたアクティブ データ資産 **customers** が表示されます。**列** リストには、customers アクティブ データ ファイルの列名が表示されます。

4. 列 リストで、**First Name** 列を選択します。

これが名前を含んだファイル内の列であるため、この列をマップして、リテラル データ「Pat」を置き換えます。

5. **OK** をクリックしてダイアログ ボックスを閉じ、データをマップします。


Silk Test Workbench によって、テスト ステップの実際のデータが、アクティブ データ ファイル内の選択したデータへのマップ式に置き換えられます。つまり、既存のステップ テキスト Enter 'Pat' は、アクティブ データ資産名、置換するデータ型、および列名を表示するようになります。

```
Enter '[[customers].Text("First Name")]'
```

6. ビジュアル テストに関連付けられたアクティブ データ ファイル内のデータに置換するデータを含む残りのテスト ステップすべてについて、ステップ 1 から 5 を繰り返します。次のステップは、「'Smith' を入力します」というステップ テキストが含まれるステップです。

下の表は以下の内容を示します。

- このビジュアル テスト内の、アクティブ データ ファイルのデータで置き換えるリテラル データを含む各ステップのテキスト
- 置き換えるデータを含むステップのプロパティの名前
- リテラル データの代わりに使用するデータを含むアクティブ データの列

 **注:** アクティブ データを使用して更新された、パスワード保護テキスト ボックスにテキストを入力するステップについては、ステップのテキストにアクティブ データ情報が示されません。

ステップ テキスト	プロパティ	アクティブ データ列
'Smith' を入力します	テキスト	Last Name
'smith@test.com' を入力します	テキスト	Email Address
'1212 Test Way' を入力します	テキスト	Mailing Address
'San Diego' を入力します	テキスト	City
リスト ボックス項目 'California' を選択します	リスト ボックス項目	State
'92121' を入力します	テキスト	Postal Code
'test' を入力します	テキスト	Password

アクティブ データ ファイルのデータを使用するようにビジュアル テストを設定できたので、これを再生し、結果を確認します。

アクティブ データのビジュアル テストの再生と分析

顧客を追加するビジュアル テストを、アクティブ データ ファイルの顧客レコードを使用するように拡張できたので、再生してみます。このビジュアル テストを再生すると、アクティブ データ ファイルの顧客レコードが実際に顧客データベースに入力されます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに AddCustomers using ActiveData と入力します。

3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザがマシンにインストールされている場合、**ブラウザの選択** ダイアログ ボックスが開きます。ブラウザを選択して、**実行** をクリックします。Silk Test Workbench によって、拡張したテストが再生されます。


5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

6. 詳細 タブをクリックして、各ステップの結果を表示します。

詳細 タブには、再生中に実行された各ステップの結果が表示されます。スクロールバーを使用して、実行されたステップをスクロールします。

結果 ウィンドウの **テスト ステップ** ペインには、ビジュアル テストの実際のステップよりもかなり多くのステップがあります。これは、繰り返しごとに、それぞれ実行ステップのセットがあるためです。

 **注:** また、式を使用して、ビジュアル テストでアクティブ データを更新することもできます。**式デザイナ** を使用すると、アクティブ データ ファイルの列のデータを入力として使用でき、データの値を変更する式を作成できます。更新されたデータは、同じアクティブ データ ファイルの同じ列に保存できます。詳細については、オンライン ヘルプの「ビジュアル テストでの式を使用したアクティブ データの更新」を参照してください。

ビジュアル テストからのスクリプトの再生

スクリプトとビジュアル テストは、資産を使用してテスト アプリケーションでのメニュー項目の選択やデータの入力などの手動のユーザー操作を自動化するという点で同じです。違いは、ユーザーの操作がどのように表現されるかです。スクリプトでは、Microsoft .NET Framework で稼動するスクリプト記述言語である Microsoft の Visual Basic が使用されますが、ビジュアル テストでは、ビジュアル ナビゲータのポイントアンドクリック インターフェイスによって生成されるステップが使用されます。


いずれの場合も、互いに独立して実行できるパワフルで柔軟性の高い自動テストを作成できます。どちらを選ぶかは、ユーザーのニーズと好みによります。また、両方を組み合わせて使用することもできます。たとえば、特殊なタスクを実行するスクリプトを作成し、ビジュアル テストにそのスクリプトを再生するステップを挿入することができます。このようにして、スクリプトで使用されるスクリプト記述言語の力を利用してビジュアル テストを補強することができます。

ビジュアル テストからスクリプトを再生すると、スクリプトは、繰り返しテスト タスクまたは特殊なテスト タスクを実行するために必要に応じて呼び出せる関数と同様に機能します。このアプローチはチームによるテスト環境で役立ちます。たとえば、経験の豊富なテスト担当者が共通するテスト機能を実行するスクリプトのライブラリを作成し、経験の浅い開発者がそこからスクリプトを選択して、ビジュアル テストを作成することができます。

乱数を生成するスクリプトの作成

まず最初に、ユーザーの年齢に使用できる乱数を生成するスクリプトを作成します。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスに `function_randomAge` と入力して **OK** をクリックします。

 **ヒント:** スクリプトに「function」と名前を付けることにより、共通して使用されるテスト タスクのライブラリを整理でき、ここからすばやくタスクを選択して、ビジュアル テストに挿入することができます。

スクリプトが **コード** ウィンドウに表示されます。

3. `Sub Main()` の下の行にカーソルを置き、以下のように入力します。

```
Dim rand As New Random()  
Dim TSrandomAge As Integer = rand.Next(10000, 99999)  
MsgBox(TSrandomAge)
```

4. ツールバーで **再生** をクリックします。**再生** ダイアログ ボックスが開きます。
5. **OK** をクリックします。メッセージ ボックスが表示され、10,000 から 99,999 の生成された乱数が表示されます。
6. **OK** をクリックします。**再生完了** ダイアログ ボックスが開きます。
7. **.NET スクリプトへ移動** をクリックします。

8. 数値範囲パラメータの (10000, 99999) をパラメータ *MinVal* と *MaxVal* に置き換えます。これら2つのパラメータを使用することにより、スクリプトを開いて値を変更するのではなく、このスクリプトを再生するビジュアル テストから乱数の範囲を設定することができます。以下のように入力します。

```
Dim TSrandomAge As Integer = rand.Next(MinVal, MaxVal)
```

9. スクリプトが適切に動作することを確認するために、Sub Main() の直後で、以下の値をパラメータに割り当てます。

```
Dim MinVal=16  
Dim MaxVal=105
```

スクリプトは以下のコードのようになっているはずですが、

```
Public Sub Main()  
    Dim rand As New Random()  
    Dim MinVal=16  
    Dim MaxVal=105  
    Dim TSrandomAge As Integer = rand.Next(MinVal, MaxVal)  
  
    MsgBox(TSrandomAge)  
End Sub
```

- 10 ツールバーで **再生** をクリックします。 **再生** ダイアログ ボックスが開きます。

- 11 **OK** をクリックします。メッセージ ボックスが表示され、16 から 105 の生成された乱数が表示されます。

- 12 **OK** をクリックします。 **再生完了** ダイアログ ボックスが開きます。

13. **NET スクリプトへ移動** をクリックします。

- 14 入力パラメータを使用して *MinVal* および *MaxVal* パラメータを設定するには、以下のステップを実行します。

- a) 作成する入力パラメータが含まれるように Main() sub を変更します。

```
Public Sub Main(args As IDictionary(Of String, Object))  
    Dim MinVal=args ("MinVal")  
    Dim MaxVal=args ("MaxVal")
```

"MinVal"と "MaxVal" は、次の手順で作成する入力パラメーターの名前です。

- b) 作成する出力パラメータが含まれるように Main() sub を変更します。

```
Public Sub Main(args As IDictionary(Of String, Object))  
  
    args ("TSrandomAge")=TSrandomAge
```

"TSrandomAge" は、次の手順で作成する出力パラメーターの名前です。

- c) 年齢はスクリプトからビジュアル テストに渡されるため、ステートメントの先頭にアポストロフィを入力してメッセージ ボックスのコードをコメントにします。

```
'MsgBox(TSrandomAge)
```

スクリプト全体は以下のようにになっているはずですが、

```
Public Module Main  
    Dim _desktop As Desktop = Agent.Desktop  
  
    Public Sub Main(args As IDictionary(Of String, Object))  
  
        Dim rand As New Random()  
  
        Dim MinVal= args("MinVal")  
        Dim MaxVal= args("MaxVal")  
  
        Dim TSrandomAge As Integer = rand.Next(MinVal, MaxVal)  
        args ("TSrandomAge")=TSrandomAge  
        'MsgBox(TSrandomAge)
```

```
End Sub
End Module
```

次に、ランダム年齢をビジュアル テストに渡すために使用するスクリプトの入力パラメータと出力パラメータを定義します。

スクリプトの入力パラメータの定義

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。このタスクでは、function_randomAge スクリプトで作成された乱数の範囲を設定する 2 つの入力パラメータを定義します。

1. **プロパティ ペイン**で、**入力パラメータの追加** を右クリックして選択します。 **スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
2. **名前** テキスト ボックスに **MinVal** と入力します。
3. **型** リストで、**数値 (Double)** を選択します。
4. **デフォルト値** テキスト ボックスに 16 と入力します。
必要に応じて、ビジュアル テストにデフォルト値を設定することもできます。
5. **OK** をクリックします。 **プロパティ ペイン**の入力パラメータのリストに、入力パラメータが表示されます。
6. **プロパティ ペイン**で、**入力パラメータの追加** を右クリックして選択します。 **スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
7. **名前** テキスト ボックスに **MaxVal** と入力します。
8. **型** リストで、**数値 (Double)** を選択します。
9. **デフォルト値** テキスト ボックスに 105 と入力します。
必要に応じて、ビジュアル テストにデフォルト値を設定することもできます。
10. **OK** をクリックします。 **プロパティ ペイン**の入力パラメータのリストに、入力パラメータが表示されます。

スクリプトの出力パラメータの定義

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。このタスクでは、function_randomAge スクリプトで作成された乱数をビジュアル テストに渡すパラメータを定義します。

1. **プロパティ ペイン**で、**出力パラメータの追加** を右クリックして選択します。 **スクリプト出力パラメータの追加** ダイアログ ボックスが開きます。
2. **名前** テキスト ボックスに **TSrandomAge** と入力します。
3. **型** リストで、**数値 (Double)** を選択します。
4. **デフォルト値** テキスト ボックスは空のままにします。
5. **OK** をクリックします。 **プロパティ ペイン**の出力パラメータのリストに、出力パラメータが表示されます。

ビジュアル テストで出力パラメータを受け取るローカル変数を定義します。

スクリプトのデータを使用するビジュアル テストの設定

スクリプトのデータを使用するようにビジュアル テストを設定するには、スクリプトを再生し、スクリプトのデータを格納するローカル変数を作成し、このローカル変数をスクリプトの出力変数と関連付けるステップを挿入する必要があります。さらに、ビジュアル テストからスクリプトにデータを渡すために、スクリプトの入力変数の値を設定する必要があります。

1. **AutoQuote** ビジュアル テストを開きます。

2. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開きます。
3. AutoQuote ビジュアル テストを変更するため、元のテストと区別できるよう **AutoQuote_Modified** と名前を変更し、**OK** をクリックします。
4. 以下のステップ テキストのあとにあるステップを選択します。
ビジュアル テスト 'AddAccount' の再生
5. **テスト ステップ** ペインで、**操作 > 挿入 > .NET スクリプト** を選択します。 **.NET スクリプトの参照** ダイアログ ボックスが開きます。
6. **function_randomAge** を選択し、**OK** をクリックします。 Silk Test Workbench により、スクリプトを再生するステップが挿入されます。
7. スクリプトのデータを格納するローカル変数を追加するには、以下のステップを実行します。
 - a) **テスト ステップ** ペインで、**操作 > 挿入 > 変数 > ローカルの追加** をクリックします。 **ローカル変数の追加** ダイアログ ボックスが開きます。
 - b) **変数名** テキスト ボックスに VTrandomAge と入力します。
 - c) **型** リストから **数値 (Double)** を選択し、**初期値** を 0 に設定します。
 - d) **OK** をクリックします。 <<開始>> ステップから変数を参照し、定義を編集することができます。
8. スクリプトのステップを選択します。
ステップは次のようになります : 「.NET スクリプト 'function_randomAge' の再生」
9. 挿入されたスクリプト ステップの **プロパティ** ペインで、**'TSrandomAge' の内容の引き渡し先** プロパティの値領域をクリックし、ローカル変数 **VTrandomAge** を選択します。
入力パラメータをスクリプトで追加したため、その値が **プロパティ** ペインに表示されます。ただし、値を編集することで、ビジュアル テストで使用される入力パラメータを変更できます。これらの値は、スクリプトでは変更されません。

これで、スクリプトのデータを使用するようにビジュアル テストが設定されました。

AutoQuote_Modified ビジュアル テストが再生されるたびに、function_randomAge スクリプトが再生され、ビジュアル テストのローカル変数 VTrandomAge に固有の乱数が渡されます。

次に、function_randomAge スクリプトで生成された乱数を含むビジュアル テストのローカルパラメータを使用するように、年齢情報を入力するステップを変更します。

ビジュアル テストでのスクリプト データの使用

このセクションでは、function_randomAge スクリプトで生成された乱数を含むビジュアル テストのローカルパラメータを使用するように、年齢情報を入力するステップを変更します。

1. **Age** テキスト ボックスで、データを入力するステップを選択します。ステップ テキストは、次のようになります :
'42' を入力します
2. **プロパティ** ペインで、**テキスト** プロパティを選択します。
3. **選択** ボタンをクリックして、**変数** を選択します。 **変数の選択** ダイアログ ボックスが開きます。
4. **VTrandomAge** を選択し、**OK** をクリックします。ステップ テキストは、次のようになります :
「'[VTrandomAge]' を入力します」

このステップで、VTrandomAge ローカル変数の値が **Age** テキスト ボックスに入力されるようになります。

テスト結果の再生と確認

1. 次のいずれか 1 つのステップを行います :

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。


2. **結果の説明** テキスト ボックスに Adding Customers using data passed from a script と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。Silk Test Workbench によって、拡張されたテストが再生され、スクリプトから渡された乱数とビジュアル テストで定義した追加の顧客情報を使用して 10 人の新規顧客が作成されます。
5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。
結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。
6. **詳細** タブをクリックして、各ステップの結果を表示します。


Silk Test Workbench スクリプト チュートリアル

Silk Test Workbench スクリプト チュートリアルによろこそ。このチュートリアルでは、スクリプトの作成、スクリプトの再生、および再生結果の分析に必要な基本ステップについて学習します。また、多数の機能を使用して、記録したスクリプトをすばやく更新および拡張する方法についても学習します。

このチュートリアルでは、Microsoft Visual Basic および Microsoft .NET Framework の基本的な知識があることが前提になっています。.NET Framework についての知識をお持ちでない場合は、Microsoft の Web サイトを参照してください。

このチュートリアルでは、Silk Test Workbench を使用して繰り返し可能なテストを作成する練習をするための Silk Test サンプル Web アプリケーション (<http://demo.borland.com/InsuranceWebExtJS/>) を使用します。

 **注:** このチュートリアルで使用されているサンプル アプリケーションは、Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Microsoft Focus は Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行することをお勧めしません。

 **注:** Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザ アドオンを無効にします。Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

各レッスンは前のレッスンの出力に基づいて進められるため、チュートリアルのレッスンは順番に完了してください。

スクリプトの記録 : 概要

サンプル Web アプリケーションで保険の見積もり要求を作成するための操作を行うと、Silk Test Workbench はその操作を記録します。スクリプトに必要な操作の記録を完了すると、記録されたスクリプトが **コード** ウィンドウに表示されます。

サンプル Web アプリケーションの起動

このチュートリアルでは、Silk Test サンプル Web アプリケーションを使用します。この Web アプリケーションは、デモ用に提供されています。

Internet Explorer で、Silk Test サンプル Web アプリケーションを使用します。チュートリアルのレッスンと同じユーザー体験を得るには、Internet Explorer 以外のサポート対象ブラウザでサンプル Web アプリケーションを実行しないでください。

1. テストの速度と信頼性を高めるために DOM 関数を記録するには、以下のステップを実行します。
 - a) **ツール > オプション** をクリックします。
 - b) **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。**記録** オプションが右側のパネルに表示されます。

- c) **xBrowser** をクリックします。
- d) **ネイティブなユーザー入力を記録する** リスト ボックスから、**いいえ** を選択します。
- e) **OK** をクリックします。



注: 通常、Web アプリケーションをテストする場合、DOM 関数ではなく、ユーザーの入力そのものを使用します。Flash および Java アプレットなどのプラグインや、AJAX を使用するアプリケーションは、ネイティブなユーザー入力ではサポートされますが、高レベルの API 記録ではサポートされません。

2. Web アプリケーションの記録または再生を行う前に、ブラウザのアドオンをすべて無効にする必要があります。ブラウザのすべてのアドオンを確実に無効にするには、以下のステップを実行します。
 - a) Internet Explorer で**ツール > インターネット オプション** を選択します。**インターネット オプション** ダイアログ ボックスが開きます。
 - b) **プログラム** タブをクリックし、**アドオンの管理** をクリックします。**アドオンの管理** ダイアログ ボックスが開きます。
 - c) アドオンのリストの**状態** 列で、各アドオンの状態が**無効** になっていることを確認します。**状態** 列に**有効** と表示されている場合は、アドオンを選択してから**無効にする** をクリックします。
 - d) **閉じる** をクリックし、**OK** をクリックします。
3. サンプル アプリケーションにリモートでアクセスするには、<http://demo.borland.com/InsuranceWebExtJS> をクリックします。サンプル アプリケーションの Web ページが開きます。

サンプル Web アプリケーションに対するスクリプトの記録

記録中は、記録を停止するまでテスト アプリケーションでのすべての操作 (Silk Test Workbench の操作は除く) が Silk Test Workbench によって記録されます。記録が完了したら、生成したスクリプトを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。
たとえば、タイトルに AutoQuote と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。**アプリケーションの選択** ダイアログ ボックスが開きます。
5. **Web** タブを選択します。
6. リストから **Internet Explorer** を選択します。
7. **移動する URL の入力** テキスト ボックスに、demo.borland.com/InsuranceWebExtJS/ を入力します。
8. **OK** をクリックします。Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
9. Insurance Company Web サイトでは、次のステップのいずれかを行います：
 - a) **Select a Service or login** リスト ボックスから **Auto Quote** を選択します。**Automobile Instant Quote** ページが開きます。
 - b) 郵便番号と電子メール アドレスを適切なテキスト ボックスに入力し、自動車タイプをクリックして、**Next** をクリックします。
たとえば、郵便番号に 92121、電子メール アドレスに jsmith@gmail.com をそれぞれ入力し、自動車タイプとして Car を指定します。
 - c) 年齢を指定し、性別と運転履歴タイプをクリックして、**Next** をクリックします。
たとえば、年齢に 42 を入力し、性別と運転履歴タイプに Male および Good をそれぞれ指定します。
 - d) 製造年、車種、モデルを指定し、財務情報タイプをクリックして、**Next** をクリックします。
たとえば、製造年に 2010 と入力し、車種とモデルに Lexus および RX400 をそれぞれ指定し、財務情報タイプとして Lease を指定します。
指定した情報の概要が現れます。

e) **Purchase** をクリックします。

Purchase A Quote ページが開きます。

f) ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。

10Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、スクリプトが表示されます。

11.NET スクリプトへ移動 をクリックします。スクリプトが **コード** ウィンドウに表示されます。

12保存 をクリックします。

記録されたスクリプトの確認

Silk Test Workbench では、それ以外のすべてのアプリケーション内の操作がすべて記録されます。指示に忠実に従っていれば、サンプル アプリケーションの Web サイトで実行した操作だけが Silk Test Workbench によってキャプチャされます。Silk Test Workbench は再生中にこれらの操作を繰り返します。

スクリプトは、以下のサンプルのようになります。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        _desktop.Control("control1").TypeKeys("9")
        With _desktop.BrowserApplication("webBrowser")
            With .BrowserWindow("browserWindow")
                .DomTextField("autoquoteZipcode").SetText("92121")
                .DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
                .DomRadioButton("autoquoteVehicle0").Select()
                .DomButton("autoquoteNext").Select()
                .DomTextField("autoquoteAge").SetText("42")
                .DomRadioButton("autoquoteGender0").Select()
                .DomRadioButton("autoquoteType1").Select()
                .DomButton("autoquoteNext").Select()
                .DomTextField("autoquoteYear").SetText("2010")
                .DomElement("img").DomClick(MouseButton.Left, New Point(8, 9))
                .DomElement("lexus").DomClick(MouseButton.Left, New Point(87, 7))
                .DomElement("img3").DomClick(MouseButton.Left, New Point(11, 10))
                .DomElement("rX400").DomClick(MouseButton.Left, New Point(96, 11))
                .DomRadioButton("autoquoteFinInfo2").Select()
                .DomButton("autoquoteNext").Select()
                .DomLink("home").Select()
            End With
        End With
    End Sub
End Module
```

スクリプトが前述の例と完全に一致しないことがあります。各ユーザーは、それぞれ異なる方法でアプリケーションと対話するためです。たとえば、フォームに情報を入力するとき、各フィールドをクリックしていくユーザーもいれば、Tab キーを使用して移動していくユーザーもいます。結果は同じですが、これらの操作の Silk Test Workbench による記録内容は異なります。このような違いはありますが、スクリプトは正しく再生されるはずで

記録されたテストの再生

スクリプトを記録し、保存したら、再生して、スクリプトが適切に動作することを確認できます。

1. 次のいずれか 1 つのステップを行います：

- **操作** > **再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。このダイアログ ボックスで、結果をどのように保存するかを指定できます。

2. **結果の説明** テキスト ボックスに Initial test results for the recorded test と入力します。

3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザがマシンにインストールされている場合、**ブラウザの選択** ダイアログ ボックスが開きます。ブラウザを選択して、**実行** をクリックします。

各結果は、一意のテスト実行番号によって識別されます。

Silk Test Workbench が最小化され、スクリプトが再生されます。再生中、サンプル アプリケーションに対して、スクリプトの記録中に行った操作が画面上で再生されます。再生が正常に完了すると、**再生完了** ダイアログ ボックスが開きます。

5. **結果へ移動** をクリックします。**結果** ウィンドウが表示されます。

結果の分析：概要

テストを再生すると、Silk Test Workbench によってテスト結果が生成されます。テスト結果にはスクリプトの再生に関する情報が含まれています。スクリプトの名前、実行番号、各ステップが実行された日付と時刻、各ステップの合格/失敗のステータスなどの重要な情報が含まれます。

スクリプトの拡張：概要

テストの拡張には、既存のテストが新しいバージョンのテスト アプリケーションで使用できるようにテストを更新することなどが含まれます。たとえば、テスト アプリケーションのさまざまな状態を処理し、検証するために、テスト ロジックを挿入することができます。さらに、テストの内容をわかりやすくしたり、重要な点について自分自身や他のユーザーに注意を促すために、メッセージ ボックスを挿入することができます。

これらは、Silk Test Workbench で既存のテストを拡張し、よりパワフルで堅牢かつ柔軟なテストを作成する、ほんの数例です。

検証の挿入

検証とは、ユーザー定義の条件を評価し、合格/失敗のメッセージを再生結果に送信するテスト ロジックです。

このレッスンでは、検証を挿入して、見積もりで正しい車両モデルが使用されていることを確認します。

1. 自動見積もりのモデル タイプを定義する、以下のテキストを選択します。

```
.DomElement("RX400").DomClick(MouseButton.Left, New Point(96, 11))
```

2. モデル (**Model**) タイプを指定する Instant Quote ウィザードのページに移動し、異なるモデル タイプに注目します。

たとえば、**GS430** は車種 **Lexus** のモデル タイプです。

3. Silk Test Workbench でモデル タイプを変更します。

textContents コードを

```
.DomElement("RX400").DomClick(MouseButton.Left, New Point(96, 11))
```

から次のように変更します：

```
.DomElement("GS430").DomClick(MouseButton.Left, New Point(96, 11))
```

4. 期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify ("GS430", .DomTextField("modelCombo").Text, "The model type is correct")
```


サンプルアプリケーションのプロパティの値を検証するテストロジックを挿入することで、記録したスクリプトを拡張できました。

アプリケーションデータの作成とローカル変数への保存

変数を使用すると、スクリプトの別の場所で使用するデータ値を保存できるため、テストを拡張できます。データは他の種類のファイルに出力することもできます。

サンプルアプリケーションの **Get Instant Auto Quote** ページには、一意の電子メールアドレスが表示されます。このページの電子メールアドレスを含むテキストは、このページのコントロールのプロパティ値です。

このレッスンでは、ローカル変数 *stremailAddr* に、このテキストを保存します。

1. スクリプトで、電子メールの値に移動します。

コードは以下のようになります。

```
.DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
```

2. 電子メールの値のあとに、以下のコードを挿入します。

```
Dim StremailAddr As String  
StremailAddr = .DomTextField("autoquoteEMail").Text
```

このステップによって、新しいローカル変数 *StremailAddr* が作成されます。このローカル変数に、**Get Instant Auto Quote page** ページの電子メールアドレスのテキストを保存します。

3. 再生中に変数テキストを表示する出力を含めるには、スクリプトに `Console.Write` コマンドを含めます。以下に例を示します。

```
Console.Write (StremailAddr)
```

4. コンソール出力を表示するには、**表示 > 出力** を選択します。**出力** ウィンドウが開きます。スクリプトを再生すると、**出力** ウィンドウが入力されます。

テストでプロパティ値が正しくキャプチャされ、保存されることを確認するために、テストを再生して結果を確認します。

拡張したスクリプトの再生と分析

記録したスクリプトにいくつかの拡張を行ったので、スクリプトを再生し、結果を分析してみます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに、Enhanced test results for the script と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。Silk Test Workbench により、拡張されたテストが再生されます。
5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

要約 タブに、スクリプトが合格した（エラーなしで正常に再生されたことを意味します）、または失敗した検証が示されます。

6. **合格 (1)** タブをクリックします。

カッコ内の数は、合格した検証の合計数を示します。**テストステップ** ペインには検証ステップが表示され、**結果の詳細** 列には検証の合格の説明が表示されます。

7. **詳細** タブをクリックすると、各操作の結果を表示されます。

8. **出力** ウィンドウで、変数に電子メール アドレスを保存した結果を示すステップまでスクロールします。
テキストは以下のようなものです。

```
jsmith@gmail.com
```

9. **合格** タブをクリックすると、モデル タイプの検証結果が表示されます。

テキストは以下のようなものです。

```
Main:Verify 合格:The model type is correct
```

検証結果は、**結果** 列および **結果の詳細** 列にも表示されます。

お疲れ様でした。これで、サンプル アプリケーションを確実にテストするスクリプトを作成できました。次のレッスンでは、スクリプトを他のスクリプトですばやく簡単に実行する方法など、より高度なテストの概念と機能について学習します。

スクリプト内でのスクリプトの実行：概要

このチュートリアルでは、Web アプリケーションから自動車保険の見積もりを取得するために必要な各操作を実行する単一のスクリプトを作成しました。単一のスクリプトは、単純なアプリケーションに対して基本的なテスト ケースを実行する場合に便利です。ただし、ほとんどのソフトウェア テストでは、アプリケーションのすべての側面をテストするなど、より厳格なアプローチが必要となります。さらに、テストアプリケーションが変化したときに既存のスクリプトをすばやく更新できることが必要です。

これらのテストの課題を解決する効率的な手段を提供するために、Silk Test Workbench はモジュール式テストをサポートしており、特定のテスト ソリューションの共通の操作を単一のテストにまとめ、同じ操作セットを必要とする他のスクリプトでスクリプトを再利用できるようにしています。

モジュール式テスト

ビジュアル テストやスクリプトなどの Silk Test Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロフィールの作成
- 注文の発注
- オンライン システムのログオフ

1 つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが 10 通りある場合、それらのシナリオを処理するために 10 個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが 1 つ追加された場合、新しいフィールドへのデータ入力を処理するために 10 個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテストシナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいこととなります。

モジュール式テストの基本が理解できたので、2 つめのテストを作成し、前のレッスンで作成したテストに追加してみます。

2 つめのスクリプトの記録

このセクションでは、チュートリアル の 2 つめのスクリプトを記録し、スクリプトの別の作成方法を学習します。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。
このチュートリアルでは、名前として **AddAccount** と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。**アプリケーションの選択** ダイアログ ボックスが開きます。
5. **Web** タブを選択します。
6. リストから **Internet Explorer** を選択します。
7. サンプル アプリケーションの **Home** ページで、**Login** セクションの **Sign Up** をクリックします。**Create A New Account** ページが開きます。
8. 適切なフィールドに、以下の情報を入力します。
次のフィールドに移動するには Tab キーを押します。

フィールド名	値
名 (First Name)	Pat
姓 (Last Name)	Smith
生年月日 (Birthday)	12/02/1990
	 注: カレンダー コントロールの月と年の隣にある下矢印をクリックして月と年を変更し、カレンダーで 12 を選択します。
電子メール アドレス (E-mail Address)	smith@test.com
住所 (Mailing Address)	1212 Test Way
市 (City)	San Diego
州 (State)	CA
郵便番号 (Postal Code)	92121
パスワード (Password)	test

9. **Sign Up** をクリックします。
10. **Continue** をクリックします。連絡先情報が表示されます。
11. ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。
12. **Log Out** をクリックします。
13. **Alt+F10** を押して記録を完了します。**記録完了** ダイアログ ボックスが開きます。
14. **保存** をクリックします。スクリプトが **コード** ウィンドウに表示されます。

別のスクリプトへのスクリプトの挿入

このセクションでは、ユーザー アカウントを追加する 2 つめのスクリプトを、元のスクリプトの自動見積もりの要求を行うコードの前に挿入します。

スクリプト内でスクリプトを実行すると、スクリプトで同じ基本操作を効率的にテストできます。



ヒント: スクリプトを別のスクリプトに挿入する際、テストアプリケーションが正しい初期再生状態になっていることを確認することが重要です。

1. **ファイル > 開く** を選択します。 **アセット ブラウザ** が開きます。
2. 左側のペインで **.NET スクリプト** を選択し、スクリプトのリストを表示します。
3. リストから **AutoQuote** をダブルクリックして開きます。
AutoQuote はこのチュートリアルで最初に作成したテストです。
4. **コード ウィンドウ** の `Public Sub Main()` コードの後ろにカーソルを置き、**Enter** を押して、新規行を追加し、以下のように入力します。

```
Workbench.RunScript ("AddAccount")
```

ここで、`AddAccount` は、作成した 2 つめのスクリプトの名前です。

見積もりステップを実行する前にアカウント情報を追加するため、`With` ステートメントの前に `Workbench.RunScript` コマンドが追加してあります。見積もりステップのあとで `AddAccount` スクリプトを実行するには、`Workbench.RunScript` コマンドを `End With` ステートメントのあとに追加します。

再生エラーへの対応：概要

再生中のエラーは、テストアプリケーションの変更、不適切なワークフローなど、さまざまな要因により発生します。このようなエラーをすばやく診断し、修正することにより、テストの保守を最小限にし、チームによるテスト作業の効率を上げることができます。

まず、前のレッスンで作成したモジュール式スクリプトを再生します。

モジュール式スクリプトの再生

前のレッスンでは、`AddAccount` を `AutoQuote` スクリプト挿入することにより、モジュール式スクリプトを作成しました。

このセクションでは、このモジュール式スクリプトを再生し、再生中のエラーを確認します。

1. `AutoQuote` スクリプトが開いている状態で、ツールバーの **再生** をクリックします。 **再生** ダイアログボックスが開きます。
2. **結果の説明** テキスト ボックスに `Responding to errors in a modular test` と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

再生中に、**Create A New Account** ページでテストが停止し、エラー メッセージが表示されます。

データベースでは、各顧客レコードに対して一意の電子メール アドレスが必要なため、このエラーが発生します。`AddAccount` スクリプトの記録中にすでに電子メール アドレスを入力しているため、この電子メール アドレスはデータベースにすでに存在しており、テストが失敗します。

結果の確認

スクリプトの結果を確認します。

1. **終了** をクリックして、再生を停止します。 **再生完了** ダイアログボックスが開きます。
2. **結果へ移動** をクリックします。 `AutoQuote` の結果が表示され、デフォルトで **要約** タブが表示されます。
要約 タブにはテスト実行の全体的な詳細が表示されます。 **ビジュアル テストまたは .NET スクリプト (実行回数)** フィールドには、「`AutoQuote(1)`」と挿入されたスクリプト「`AddAccount(1)`」がリストされます。
3. **詳細** タブをクリックします。

4. 青色のテキストのステップまでスクロールします。

結果 列と **結果の詳細** 列を確認すると、再生中に発生したエラーに関する情報をすばやく参照できます。



注: 失敗 タブには、再生エラーを含むステップは表示されません。失敗した検証のみが表示されます。

再生エラーを診断する方法を学びました。このチュートリアルのも目的のため、一旦スクリプトの再生を成功するためにスクリプトの電子メールを手動で変更してエラーを回避することができます。つまり、`AddAccount` スクリプトの

```
smith@test.com
```

を次のように変更します：

```
psmith@test.com
```

ヘルプについてのヘルプ

このセクションには、以下の情報があります。

- Silk Test Workbench ヘルプ
- ヘルプで使用されている表記規則

Silk Test Workbench ヘルプ

Silk Test Workbench のヘルプには、概念的な概要と手順を説明するトピックがあります。これらのトピックのタイプでは、必要に応じて、一般的な情報から特定の情報に移動できます。

概念的な概要 概念的な概要では、Silk Test で作業するために必要な製品のアーキテクチャ、コンポーネント、およびその他の情報を提供します。ほとんどの概要の最後には、関連するより詳細な情報へのリンクがあります。

手順の説明 手順の説明では、ステップごとに説明を提供します。ほとんどの手順の最後には、関連する手順へのリンクがあります。また、ほとんどの概念的な概要には、関連する手順へのリンクがあります。

ヘルプで使用されている表記規則

Silk Test Workbench のヘルプでは、以下の表記規則が使用されます。

規則	表す内容
等幅フォント	入力する必要があるソース コードおよびテキスト。
ボールド体	ダイアログ ボックスおよびその他のユーザー インターフェイス要素への参照。
イタリック体	変数などの識別子。イタリック体のテキストは新しい用語を強調するためにも使用されます。

ヘルプを表示する方法

Silk Test Workbench ヘルプ メニューには以下のコマンドがあります。

- **目次** をクリックすると、Silk Test Workbench ヘルプの目次が表示されます。
- **検索** をクリックすると、フルテキスト検索を使用してトピックを検索できます。
- **索引** をクリックすると、キーワードの索引を使用してトピックを検索できます。

- **チュートリアル** をクリックすると、Silk Test Workbench のチュートリアルを実行できます。このチュートリアルでは、サンプル アプリケーションでビジュアル テストを実際に作成、実行しながら、Silk Test Workbench の機能を学べます。
- Silk Test Workbench の**バージョン情報** をクリックすると、現在インストールされているバージョン、著作権情報、およびシステム情報が表示されます。

選択したダイアログ ボックスにある **ヘルプ** ボタンをクリックすることによっても、Silk Test Workbench ヘルプにアクセスできます。

ヘルプ トピックの印刷

1. **目次** タブをクリックします。
目次 タブが表示されていない場合は、**表示** をクリックして、ナビゲーション ペインを表示します。
2. 印刷するトピックを選択します。
3. **印刷** をクリックします。
4. 複数のトピックを印刷するには、ブックを選択し、**印刷** をクリックして、**選択された見出しおよびすべてのサブトピックを印刷** オプション ボタンをクリックします。
トピックは見出しごとに別のページに印刷されるのではなく、連続して印刷されます。

使用状況データの収集の有効化/無効化

Micro Focus では、テスト全体にわたるユーザー エクスペリエンスの向上を図るため、Micro Focus ソフトウェアとサービスの使用方法に関する情報を収集し、Micro Focus に提供していただくことを望んでいます。Silk Test Workbench のインストール時に使用許諾契約の条項に同意することにより、Silk Test Workbench の使用方法に関する情報と、Silk Test Workbench をインストールしたコンピューターに関する情報の収集を Micro Focus に許可したことになります。Micro Focus は、名前や住所などの個人の特定を可能にする情報の収集や、スクリプトやパスワードなどのデータ ファイルの収集は行いません。この情報の収集を Micro Focus に許可することにより、Micro Focus では、その傾向と使用方法のパターンの識別に役立てることができま。

Micro Focus による使用状況データの収集を有効化または無効化するには：

1. メニューから **へるぷ > Silk Test Workbench のバージョン情報** をクリックします。
2. バージョン情報ダイアログ ボックスで、**カスタマ フィードバック オプション** をクリックします。
3. 以下のオプションのいずれかを選択します。
 - 使用状況データの収集を有効化するには、**はい、参加します** をクリックします。
 - 使用状況データの収集を無効化するには、**いいえ、参加しません** をクリックします。
4. **OK** をクリックします。

Micro Focus へのお問い合わせ

Micro Focus は、世界的規模のテクニカル サポートおよびコンサルティング サービスを提供します。すべての顧客のビジネスを成功に導くために、信頼できるサービスをタイムリーに提供するように、Micro Focus はワールドワイドのサポート体制を整えています。

保守およびサポート契約を結んだすべてのお客様、および製品を評価中のお客様は、カスタマー サポートを受けることができます。高度なトレーニングを積んだスタッフが、お客様の質問にできる限り迅速かつ専門的にお答えします。

<http://supportline.microfocus.com/assistedservices.asp> にアクセスするか、またはメールを supportline@microfocus.com に送信して、Micro Focus SupportLine と直接連絡できます。

また、<http://supportline.microfocus.com> の Micro Focus SupportLine では、最新のサポートに関するニュースや、さまざまなサポート情報を得ることができます。このサイトに初めてアクセスした場合は、ユーザー登録が必要な場合があります。

Micro Focus SupportLine が必要とする情報

Micro Focus SupportLine をご利用の場合は、可能な限り次の情報を提供ください。情報が多ければ多いほど、Micro Focus SupportLine はお客様に適切なサービスを提供できます。

- 問題の原因と思われるすべての製品の名前およびバージョン番号
- 使用しているコンピュータの製造元およびモデル
- システム情報 (オペレーティング システムの名前やバージョン、プロセッサやメモリの詳細など)
- 問題の詳細な説明 (問題の再現手順など)
- 発生したエラー メッセージ
- お客様のシリアル番号

これらの番号は、Micro Focus から受け取った Electronic Product Delivery Notice 電子メールの件名および本文に記述されています。

テストの作成

ビジュアルテストとスクリプトを作成する場合のプロセスおよび考慮事項について説明します。

ビジュアルテストの作成

ビジュアルテストは、一連の単純な操作が記録されたものである場合もあれば、関数呼び出しやアクティブデータが含まれる複雑なものである場合もあります。

ビジュアルテストをすばやく作成するには、記録機能を使用するのが最適の方法です。記録機能により、Silk Test Workbench が生成するビジュアルテストのテストステップを学習することができます。これにより、ビジュアルテスト作成の基本を理解し、効率的なテストソリューションの作成についての理解を深めることができます。

記録中、Silk Test Workbench との対話を除くすべての操作が記録されます。ただし、記録を中断して、検証を挿入することもできます。最初の記録が終了したら、ビジュアルテストを変更できます。

ビジュアルテストの記録

記録中は、記録を停止するまで、テストアプリケーションでのすべての操作（Silk Test Workbench の操作は除く）が Silk Test Workbench によって記録されます。記録が完了したら、生成したビジュアルテストを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. 省略可能 : Web アプリケーションに対してテストをする際に、画面プレビューから記録したコントロールを更新できるようにするには、ビジュアルテストを記録する前にブラウザコントロールのキャプチャを有効化してください。

詳細については、「記録の出力オプションの設定」を参照してください。



注: Web アプリケーションの記録中にコントロールのキャプチャを有効化すると、ビジュアルテストのパフォーマンスが低下する恐れがあります。Micro Focus では、コントロールのキャプチャを無効化することをお勧めします。

2. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログボックスが開きます。
3. [資産の種類] リストから **ビジュアルテスト** を選択し、**資産名** テキストボックスにビジュアルテストの名前を入力します。



注: デフォルトでは、アクティブになっているプロジェクトにビジュアルテスト資産が作成されます。別のプロジェクトにビジュアルテスト資産を作成するには、**プロジェクト** リストで該当するプロジェクトを選択します。

4. 省略可能 : Silk Test Workbench の記録オプションを変更する場合は、**オプション** をクリックします。
5. 記録をすぐに開始するために、**記録の開始** チェックボックスをオンにします。
6. **OK** をクリックして、ビジュアルテストを資産として保存し、記録を開始します。

記録の開始 チェックボックスをオンにせずに **OK** をクリックすると、Silk Test Workbench によりビジュアルテストが資産として保存され、**ビジュアルナビゲータ** が表示されます。

アプリケーションの選択 ダイアログボックスが開きます。



注: テストするアプリケーションがリストに表示されない場合は、**キャプションを持たないプロセスを表示しない** チェックボックスをオフにします。このオプションはデフォルトでオンになっており、キャプションが付いているアプリケーションのみをフィルタ処理するために使用します。

7. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。

- ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
8. 標準アプリケーションをテストするには、リストからアプリケーションを選択します。
9. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
- a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
10. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
- a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。
このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
Silk Test Workbench は、モバイル デバイスまたはエミュレータ上に指定したアプリをインストールします。
11. **OK** をクリックします。Silk Test Workbench が最小化され、アプリケーションと **記録中** ウィンドウが開きます。
12. テストする操作を記録します。
記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
13. サポートするブラウザ上で Web アプリケーションを記録している場合で、かつ UI で直接アクセスできないコントロールを扱いたい場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**記録中** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択します。
この機能は、次のブラウザに対するテストで利用できます。
- Microsoft Edge
 - Apple Safari
 - Mozilla Firefox 41 以降
 - Google Chrome 50 以降
 - モバイル ブラウザー
14. 省略可能：検証ロジックをテストに追加します。
- 標準アプリケーションまたは Web アプリケーションをテストしている場合、テストに検証ロジックを追加するには、**Ctrl+Alt** をクリックします。Silk Test Workbench により記録が一時的に停止し、**テスト ロジック デザイナ** ウィザードが表示されます。
ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。
 - Microsoft Edge 上の Web アプリケーション、またはモバイル アプリケーションをテストしている場合、検証したいオブジェクトをクリックし、**操作の選択** ダイアログ ボックスで **検証の追加** をクリックします。
- 検証の追加に関する詳細については、「記録中のスクリプトへの検証の追加」を参照してください。

15 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバーアイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、ビジュアルテストは **テスト ステップ** ペインに表示されます。

16 次のいずれか 1 つのステップを行います：

- **記録完了** ダイアログ ボックスを閉じて、テスト スクリプトを保存するには、**再生** をクリックします。
再生 ダイアログ ボックスが表示され、結果情報を指定してテストを再生できます。
- **ビジュアル ナビゲータ** の **テスト ステップ** ペインでビジュアルテストを開くには、**ビジュアルテストへ移動** をクリックします。
- ビジュアルテストを保存して、**記録完了** ダイアログ ボックスを終了するには、**保存** をクリックします。
ビジュアルテストが **ビジュアル ナビゲータ** の **テスト ステップ** ペインに表示されます。

記録されたテストが **ビジュアル ナビゲータ** に表示されているときには、いつでもそのテストを再生できます。

開始画面からのビジュアルテストの記録

記録中、セッションを停止するまで、Silk Test Workbench との対話を除くすべての操作が記録されます。記録が完了したら、ビジュアルテストを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. 省略可能：Web アプリケーションに対してテストをする際に、画面プレビューから記録したコントロールを更新できるようにするには、ビジュアルテストを記録する前にブラウザコントロールのキャプチャを有効化してください。

詳細については、「記録の出力オプションの設定」を参照してください。



注： Web アプリケーションの記録中にコントロールのキャプチャを有効化すると、ビジュアルテストのパフォーマンスが低下する恐れがあります。Micro Focus では、コントロールのキャプチャを無効化することをお勧めします。

2. **開始画面** で、**新しいビジュアルテストの記録** をクリックします。**アプリケーションの選択** ダイアログボックスが開きます。

3. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。

- ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
- Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
- ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。

4. 標準アプリケーションをテストするには、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、リストからアプリケーションを選択します。

5. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。

a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。

b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。

たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。

- c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
- d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
6. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
- a) アプリをテストするモバイル デバイスをリストから選択します。
- b) **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。
このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
Silk Test Workbench は、モバイル デバイスまたはエミュレータ上に指定したアプリをインストールします。
7. **OK** をクリックします。Silk Test Workbench が最小化され、アプリケーションと **記録中** ウィンドウが開きます。
8. テストする操作を記録します。
記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
9. サポートするブラウザー上で Web アプリケーションを記録している場合で、かつ UI で直接アクセスできないコントロールを扱いたい場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**記録中** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択します。
この機能は、次のブラウザーに対するテストで利用できます。
- Microsoft Edge
 - Apple Safari
 - Mozilla Firefox 41 以降
 - Google Chrome 50 以降
 - モバイル ブラウザー
- 10 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、ビジュアル テストは **テスト ステップ** ペインに表示されます。
- 11 次のいずれか 1 つのステップを行います：
- **記録完了** ダイアログ ボックスを閉じて、テスト スクリプトを保存するには、**再生** をクリックします。
再生 ダイアログ ボックスが表示され、結果情報を指定してテストを再生できます。
 - **ビジュアル ナビゲータ** の **テスト ステップ** ペインでビジュアル テストを開くには、**ビジュアル テストへ移動** をクリックします。
 - ビジュアル テストを保存して、**記録完了** ダイアログ ボックスを終了するには、**保存** をクリックします。
ビジュアル テストが **ビジュアル ナビゲータ** の **テスト ステップ** ペインに表示されます。
- 記録されたテストが **ビジュアル ナビゲータ** に表示されているときには、いつでもそのテストを再生できます。

複数のテスト アプリケーションをテストするビジュアル テストの記録

複数のテスト アプリケーションをテストするビジュアル テストを記録できます。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

1. テストする主要なアプリケーション用に、ビジュアル テストを記録するか手動で作成します。
2. ビジュアル テストを開きます。
3. **タスク** ペインで <<開始>> ステップをクリックします。
4. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
5. **アプリケーション構成** テキスト ボックスをクリックします。
6. **アプリケーション構成の追加** をクリックします。**アプリケーションの選択** ダイアログ ボックスが表示されます。
7. 追加でテストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
8. 標準アプリケーションをテストするには、リストからアプリケーションを選択します。
9. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
 - a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
10. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
 - a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。
このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
Silk Test Workbench は、モバイル デバイスまたはエミュレータ上に指定したアプリをインストールします。
11. **OK** をクリックします。Silk Test Workbench により、新しいアプリケーション構成が **プロパティ** ペインに追加されます。
12. 新しいアプリケーション構成を使用して、ビジュアル テストの追加ステップを記録します。

ビジュアル テストでのロケーターからオブジェクト マップ エントリへの移動

オブジェクト マップ エントリの ID 以外の情報、つまりコマンドが実行されたときに Open Agent が使用する生のロケーターを参照したい場合は、以下の手順に従います。

1. ビジュアル テストを開きます。
2. **テストステップ** ペインでステップを選択します。**プロパティ** ペインで、**ロケーター** フィールドにロケータ テキストが表示されます。
3. カーソルを **ロケーター** フィールドのテキストに置いてください。**オブジェクト マップを開く** ボタンが右側に表示されます。
4. **オブジェクト マップを開く** ボタンをクリックします。選択された適切な項目が表示された状態で、**オブジェクト マップ** ウィンドウが、ツリー ビューに表示されます。

Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するビジュアルテストの構成

Java Network Launching Protocol (JNLP) を使用して起動するアプリケーションでは、Silk Test Workbench に追加の構成が必要です。これらのアプリケーションは Web から起動されるため、実際のアプリケーションと、「Web Start」を起動するアプリケーションを起動するように、アプリケーション構成を手動で構成する必要があります。このようにしない場合、アプリケーションがすでに実行されていないかぎり、ビジュアルテストを再生すると、失敗します。

1. テストするアプリケーションのビジュアルテストを記録します。
2. **タスク** ペインで <<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
4. **アプリケーション構成** フィールドをクリックし、**アプリケーション構成の追加** をクリックします。**アプリケーションの選択** ダイアログ ボックスが表示されます。
5. **Windows** タブを選択します。
6. テストする Java アプリケーションを選択します。
7. **OK** をクリックします。Silk Test Workbench は新しいアプリケーション構成を追加します。
8. javaws.exe を使用して Web Start をダウンロードし、起動するように、最初のアプリケーション構成 (**構成 1**) を変更します。
 - a) **構成 1** フィールドをクリックし、**アプリケーション構成の編集** をクリックします。
 - b) **実行可能ファイル** フィールドに、Web Start の実行可能ファイルの名前とファイルへのパスを入力します。
たとえば、以下のように入力します。%ProgramFiles%¥Java¥jre6¥bin¥javaws.exe
 - c) **作業ディレクトリ** フィールドに、ダウンロードするファイルの一時ディレクトリの場所を入力します。
たとえば、C:¥temp と入力します。temp は、ローカル C ドライブのディレクトリです。
 - d) **ロケーター** フィールドには、テストするアプリケーションのメイン ウィンドウを識別するためのロケーターを指定します。
ビジュアルテストを記録したときに正しいロケーターが記録されているため、ほとんどの場合、これを変更する必要はありません。
たとえば、SwingSet3 JNLP アプリケーションをテストする場合、ロケーターは SwingSet3 です。
 - e) **コマンド ライン引数** フィールドに、テスト アプリケーションを起動する JNLP の URL を入力します。
たとえば、SwingSet3 JNLP アプリケーションを使用するには、http://download.java.net/javadesktop/swingset3/SwingSet3.jnlp と入力します。
 - f) **コマンド ラインパターン** フィールドのすべての値を削除します。
 - g) 基本状態に指定されたロケーターの Silk Test Workbench による検索が何ミリ秒経過後にタイムアウトするかを示す値を変更するには、**タイムアウト** フィールドに値を入力します。
デフォルトで、**タイムアウト (ミリ秒)** は 30000 に設定されています。ほとんどのアプリケーションでは、この値で十分です。
 - h) **基本状態の実行** テキスト ボックスは **True** に設定されたままにします。
9. テストする Java アプリケーションを使用するように、新しいアプリケーション構成を変更します。
 - a) **実行可能ファイル** フィールドに、テストするアプリケーションの実行可能ファイルの名前とファイルへのパスを入力します。
たとえば、SwingSet3 JNLP アプリケーションを使用するには、 *¥javaw.exe と入力します。
 - b) **作業ディレクトリ** フィールドは空のままにします。
 - c) **ロケーター** フィールドには、テストするアプリケーションのメイン ウィンドウを識別するためのロケーターを指定します。
これは、ほとんどの場合、**構成 1** で指定したロケーター名と一致します。

- たとえば、SwingSet3 JNLP アプリケーションをテストする場合、ロケータは SwingSet3 です。
- d) **コマンドライン引数** フィールドのすべての値を削除します。
 - e) **コマンドラインパターン** フィールドに、テストアプリケーションのコマンドライン引数のパターンを入力します。
コマンドライン引数のパターンを含むアプリケーション構成では、プロセス名とコマンドライン引数のパターンの両方が一致するプロセスのみが、テストに対して有効化されます。コマンドライン引数のパターンが定義されていない場合、指定されたプロセス名の実行可能パターンを持つプロセスがすべて有効化されます。
たとえば、SwingSet3 JNLP アプリケーションを使用するには、*SwingSet* と入力します。この引数により、SwingSet3 Java アプリケーションのみが確実に起動されます。
 - f) 基本状態に指定されたロケータの Silk Test Workbench による検索が何ミリ秒経過後にタイムアウトするかを示す値を変更するには、**タイムアウト** テキストボックスに値を入力します。
デフォルトで、**タイムアウト (ミリ秒)** は 30000 に設定されています。ほとんどのアプリケーションでは、この値で十分です。
 - g) **基本状態の実行** チェックボックスをオンにします。

ビジュアルテストを再生すると、JNLP アプリケーションが予期したとおりに起動します。

新しいブラウザ ウィンドウで開くビジュアルテストの構成

ビジュアルテスト用に既存のブラウザ ウィンドウを使用するのではなく、別のブラウザ ウィンドウを開く場合があります。たとえば、Silk Central Test Manager から Silk Test Workbench ビジュアルテストを実行すると、テストは既存の Silk Central ウィンドウを使用します。Silk Central に戻るには、ユーザーはブラウザ ウィンドウで **戻る** をクリックする必要があります。Silk Central から実行するテスト用に別のブラウザ ウィンドウを開くテストを構成すると、この動作を回避できます。

1. テストするアプリケーションのビジュアルテストを記録します。
2. **テストステップ** ペインで、<<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
4. **アプリケーション構成** フィールドをクリックし、**アプリケーション構成の追加** をクリックします。**アプリケーションの選択** ダイアログボックスが表示されます。
5. アプリケーション構成をクリックし、**アプリケーション構成の編集** をクリックします。**アプリケーション構成の編集** ダイアログボックスが表示されます。
6. **ブラウザの種類** リストで、使用するブラウザを選択します。
7. **基本状態の実行** チェックボックスをオフにします。
8. **テストステップ** ペインで、**挿入 > 外部プログラム** を選択します。Silk Test Workbench によって、「プログラム」の実行 (通常表示) が **テストステップ** ペインに追加されます。
9. **プロパティ** ペインで、**コマンド** カテゴリに移動します。
10. **パス名** フィールドをクリックし、**ファイルの選択** をクリックします。**ファイルの選択** ダイアログボックスが表示されます。
11. ブラウザーの実行可能ファイルへ移動し、**開く** をクリックします。
たとえば、Internet Explorer を使用するには、C:\Program Files\Internet Explorer\iexplore.exe を指定します。
12. **操作** リストボックスで **開く** が指定されていることを確認します。
13. テストする Web アプリケーションの URL を、**パラメータ** フィールドに入力します。
たとえば、サンプル Web アプリケーションをテストするには、http://demo.borland.com/InsuranceWebExtJS/index.jsf を入力します。

ビジュアルテスト内での特定のステップの表示

ビジュアルテストで表示するテストステップをカスタマイズして、特定のステップだけを表示することができます。ビジュアルテストに表示されるステップをフィルタ処理すると、テストの性質をわかりやすくしたり、レビューが必要なステップのみにフォーカスしたりできます。特定のタイプのステップのみが表示されているビジュアルテストで追加の操作を記録すると、記録後は表示条件を満たす記録済み操作のみが表示されます。表示オプションを **ステップと画面** に変更すると、新しく記録した他のステップが表示されます。

1. ビジュアルテストで **テストステップ** ペインを開き、**操作 > 表示** を選択します。
2. ビジュアルテストで以下のステップ表示オプションを選択します。
3. 画面のみを操作するビジュアルテストのステップをすべて非表示にする場合は、**ステップのみ** をクリックします。
4. テスト操作を実行するステップをすべて非表示にする場合は、**画面のみ** をクリックします。
5. ビジュアルテストのテストステップをすべて表示する場合は、**ステップと画面** をクリックします。これはデフォルトの表示設定です。
6. **テストステップ** ペインに **説明** 列を表示するには、**ステップの説明** をクリックします。この列は、デフォルトでは表示されません。説明が記述されたステップには、**説明** 列に **テストステップの説明** のアイコンが表示されます。**プロパティ** ペインでステップの **説明** の値を更新すると、テストステップに説明を入力できます。

テストアプリケーションでのビジュアルテスト用コントロールの識別

テストを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。ビジュアルテストを記録した後で、テスト対象アプリケーションで可視のコントロールを識別して、ビジュアルテストで使用するコントロールを変更したり、ビジュアルテストに手動でテストステップを挿入することができます。

1. **テストステップ** ペインで、コントロールを変更するステップを選択します。
2. **プロパティ** ペインで、**ローケター** テキストボックスの **コントロールの識別** メニューを展開します。
3. **テスト対象アプリケーション** を選択します。
4. テストしているアプリケーションにカーソルを移動します。
カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが濃い枠で強調表示されます。



ヒント: Esc を押すと、識別モードが終了し、ビジュアルテストに戻ります。

5. 使用するコントロールが強調表示されているときに、コントロールをクリックします。




注: 別のクラスにアクセスしている自動ステップを編集した場合、ステップによってアクセスされているクラスが変更されたこと、またその変更を適用するかどうかを確認するメッセージが Silk Test Workbench から表示されます。**はい** をクリックして新しいクラスとコントロール情報で **ローケター** を更新するか、**いいえ** をクリックしてコントロールを識別せずに **ビジュアルナビゲータ** に戻ります。


ローケター テキストボックスが新たに選択されたコントロール情報に更新され、このコントロールが再生時に使用されます。

画面プレビューでのビジュアルテスト用コントロールの識別

テストを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。ビジュアルテストを記録し、テスト対象アプリケーションが利用できない場合、**画面プレビュー** を使用してビジュアルテストで使用するコントロールを変更したり、ビジュアルテストに手動でテストステップを挿入することができます。


 **注:** Web アプリケーションに対して、ビジュアルテストの**画面プレビュー** でコントロール名を識別するには、ビジュアルテストの記録時にブラウザ コントロールのキャプチャが有効化されている必要があります。詳細については、「記録の出力オプションの設定」を参照してください。

1. **テストステップ** ペインで、コントロールを変更するステップを選択します。
2. **プロパティ** ペインで、**ローケター** テキストボックスの **コントロールの識別** メニューを展開します。
3. **画面プレビュー** を選択します。
4. **画面プレビュー** でキャプチャされた画面にカーソルを移動します。

 **ヒント:** **画面プレビュー** で Esc を押すと、識別モードが終了し、ビジュアルテストに戻ります。

カーソルを **画面プレビュー** 内の各コントロールに動かすたびにコントロールが濃い枠で強調表示されます。

5. 使用するコントロールが強調表示されているときに、コントロールをクリックします。

 **注:** 別のクラスにアクセスしている自動ステップを編集した場合、ステップによってアクセスされているクラスが変更されたこと、またその変更を適用するかどうかを確認するメッセージが Silk Test Workbench から表示されます。**はい** をクリックして新しいクラスとコントロール情報で **ローケター** を更新するか、**いいえ** をクリックしてコントロールを識別せずに **ビジュアルナビゲータ**に戻ります。

ローケター テキストボックスが新たに選択されたコントロール情報に更新され、このコントロールが再生時に使用されます。

オブジェクトの識別ダイアログを使用したビジュアルテスト用コントロールの識別

ビジュアルテストを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。ビジュアルテストを記録した後で、ビジュアルテストで使用するコントロールを変更したり、ビジュアルテストに手動で追加のテストステップを挿入することができます。**オブジェクトの識別** ダイアログを使用して、コンテナのようなテスト対象アプリケーションの非表示コントロールを識別できます。

1. **テストステップ** ペインで、コントロールを変更するステップを選択します。
2. **プロパティ** ペインで、**ローケター** テキストボックスの **コントロールの識別** メニューを展開します。
3. **オブジェクトの識別ダイアログ** を選択します。
4. Web アプリケーションをテストする場合は、**ブラウザの選択** ダイアログボックスが開きます。コントロールを識別するブラウザを選択し、**識別の開始** をクリックします。**オブジェクトの識別** ダイアログボックスが開きます。
5. **識別の開始** をクリックします。
6. テストしているアプリケーションにカーソルを移動します。カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが強調表示されます。関連するローケター文字列またはオブジェクトマップ項目は、**選択済みローケター** テキストボックスに表示されます。
7. 使用するコントロールが強調表示されているときに、コントロールをクリックします。
8. 省略可能：**ローケターの詳細の表示** をクリックすると、ローケター ツリーや関係する属性のすべてが **ローケター属性** テーブルに表示されます。

9. 省略可能：記録したロケータ属性は、**ロケータ属性** テーブルの別のロケータ属性で置き換えることができます。

たとえば、記録したロケータは以下のように表示されます。

```
/BrowserApplication//BrowserWindow//input[@id='loginButton']
```

ロケータ属性 テーブルに textContents Login がリストされている場合、以下のようにしてロケータを手動で変更できます。

```
/BrowserApplication//BrowserWindow//input[@textContents='Login']
```

新しいロケータは、**選択済みロケータ** テキストボックスに表示されます。

- 10 **テスト** をクリックして、ロケータがコントロールを正しく認識することを確認します。

- 11 ロケータを使用する新しいステップをビジュアル テストに追加するには、**貼り付け** をクリックします。**操作の選択** ダイアログ ボックスが開きます。コントロールに対して実行する操作を選択し、**OK** をクリックします。

特定のテスト ステップでの追加操作の記録

ビジュアル テスト ステップを作成したあと、テストを開き、テストの任意の場所から追加操作を記録できます。これにより、いつでも既存のテストを追加操作で更新できます。

1. 既存のビジュアル テストを開きます。
2. 追加操作を記録する場所の直後のテスト ステップを選択します。



注： 記録した操作は、選択した場所の後に挿入されます。テスト対象アプリケーション (AUT) は基本状態に戻りません。代わりに、テストスクリプトの直前の操作が記録された範囲で AUT を開いておきます。

3. **操作 > 記録** をクリックします。

Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。

4. AUT に対して実行したい追加操作を記録します。

記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。

5. 記録を停止するには、**記録中** ウィンドウの **停止** をクリックします。

記録を停止するキーの組み合わせ（デフォルトでは **Alt+F10**）を押すか、タスクバーの Silk Test Workbench アイコンをクリックして、記録を停止することもできます。

テスト アプリケーションからビジュアル テストへの画面の挿入

ビジュアル テスト ステップを作成したあと、追加の画面と操作を記録するのではなく、テストを開き、画面キャプチャを挿入して、関連するテストステップを手動で追加することができます。画面キャプチャを挿入すると、テスト アプリケーションの更新された画面を含むビジュアル テストを手動で構築できます。この方法を使用すると、新しい画面を記録するよりも効率が上がる場合があります。

1. テストするアプリケーションを開き、キャプチャするページまたはダイアログに移動します。
2. 既存のビジュアル テストを開きます。
3. 新しい画面の挿入箇所の直後にあるテスト ステップを選択します。



注： テスト ステップと画面を **テスト ステップ** ペインに表示するには、**操作 > 表示 > ステップと画面** をクリックします。

4. **挿入 > Using** を選択します。

Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。

5. キャプチャする画面をクリックするか、またはホット キーのキー入力の組み合わせを押して、画面をキャプチャします。

クリックまたはホット キーのキー入力の組み合わせによって画面がキャプチャされるかどうかは、ロケータの記録設定オプションによって決まります。この設定を変更するには、**ツール > オプション** を選択し、**オプション** メニュー ツリーの **記録** を展開して、**ロケーター** をクリックします。

新しい Using ステップが、**テスト ステップ** ペインで、選択したテストステップのあとに表示されます。

必要に応じて、関連付けられている自動テスト ステップを追加します。

ビジュアル テストのオブジェクト マップ項目またはロケーターの手動による記録

開始する前に、テストするアプリケーションが実行中であることを確認します。

オブジェクトの識別 ダイアログ ボックスを使用して、オブジェクト マップ項目またはロケーターを手動でキャプチャし、ビジュアル テストに対応するテスト ステップを作成します。

1. 変更するビジュアル テストを開きます。記録された各操作に対応するステップは、**テスト ステップ** ペインに表示されます。
2. オブジェクトと操作を挿入するテスト ステップをクリックします。
新しいオブジェクトと挿入は、選択したテストステップの下に挿入されます。
3. **ツール > オブジェクトの識別** を選択します。**オブジェクトの識別** ダイアログ ボックスが開きます。
4. **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケーターを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケーターをキャプチャします。

5. **識別の開始** をクリックします。

Google Chrome でオブジェクト マップ項目またはロケーターをキャプチャする場合は、Google Chrome の選択されたインスタンスが適切なオートメーション パラメータを使用して起動されているかどうかを Silk Test Workbench が検出します。適切なパラメータを使用して起動されていない場合は、Silk Test Workbench が Google Chrome を終了し、オートメーション パラメータ セットを使用して再起動します。

6. 省略可能 : ロケーターを記録する前に、テスト対象アプリケーションを特定の状態にするには、**識別の停止** をクリックします。これ以降、テスト対象アプリケーションで実行する操作は、記録されなくなります。ロケーターの記録を再開する場合は、**識別の開始** をクリックします。

7. 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホット キー** モードでオブジェクトをキャプチャします。
デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケーター文字列が **選択済みロケーター** テキスト ボックスに表示されます。

8. **ロケーターの詳細** テーブルでのロケーターの表示方法を絞り込むには、以下のいずれかの操作を実行します。

- **ロケーターの詳細の非表示** : **ロケーターの詳細** テーブルを非表示にするには、このリンクをクリックします。
- **オブジェクト マップの表示** : オブジェクト マップ項目名を **ロケーター** 列に表示するには、このチェック ボックスをオンにします。オブジェクト マップ項目名は、コントロールまたはウィンドウに対して、コントロールやウィンドウのロケーターではなく論理名 (エイリアス) を関連付けます。デ

フォルトでは、オブジェクト マップ項目名が表示されます。ロケーターを使用するには、このチェックボックスをオフにします。

- **完全ロケーターの表示**：完全ロケーター名を表示するには、このチェックボックスをオンにします。オブジェクトに関連付けられた属性のみを表示するには、このチェックボックスをオフにします。
- **プロパティの表示**：**ロケーターの詳細** テーブルで選択したオブジェクトの属性および属性値を表示するには、このチェックボックスをオンにします。このテーブルの属性を選択して、ロケーターの識別に使用できます。[プロパティ] サブツリーを非表示にし、ロケーターの詳細のみを表示するには、このチェックボックスをオフにします。

9. 選択済みロケーター テキスト ボックスに表示されるオブジェクトが使用するオブジェクトであることをテストするには、**テスト** をクリックします。Silk Test Workbench により、ロケーターに対応するオブジェクトが、テストするアプリケーションで強調表示されます。

10 記録したロケーターを置換するには、**ロケーターの詳細** テーブルで、使用するロケーターを選択します。新しいロケーターは、**選択済みロケーター** テキスト ボックスに表示されます。

11 オブジェクトをコピーし、ビジュアル ステップでテスト ステップを作成するには、**貼り付け** をクリックします。**操作の選択** ダイアログ ボックスが開きます。

12 オブジェクトで実行する操作を選択し、**OK** をクリックします。Silk Test Workbench により、新しいテスト ステップのオブジェクトと操作が、ビジュアル テストに追加されます。

既存のビジュアル テストを開く

既存のビジュアル テストは、**開始画面** または **アセットブラウザ** から開くことができます。

1. 開始画面 からビジュアル テストを開くには、**開始画面** の **スクリプト** ペインで以下のステップのいずれかを実行します。

- **最新** リストで、適切なビジュアル テストの **ビジュアル テスト 'プロジェクト名 - テスト名'** を開くをクリックします。
- **開く** をクリックして **アセットブラウザ** を開き、適切なビジュアル テストを右クリックして **開く** を選択します。


2. アセットブラウザ からビジュアル テストを開くには、以下のステップを実行します。

- a) **表示** > **アセットブラウザ** を選択します。
- b) **資産の種類** ペインで **ビジュアル テスト** を選択し、適切なビジュアル テストを右クリックして **開く** を選択します。

ビジュアル ナビゲータ の **テスト ステップ** ペインで、ビジュアル テストが開きます。

ビジュアル テストの保存

新しいビジュアル テストまたは変更したビジュアル テストは明示的に保存することができます。更新したビジュアル テストを終了するときに、変更を保存するかどうかを確認するメッセージが表示されます。

 **ヒント**: 保存した変更を取り消すことはできません。

- 1. ファイル** > **名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開きます。
- 2. 省略可能** : ビジュアル テスト名を変更し、説明を入力します。
- 3. OK** をクリックします。

ビジュアル テストの概要

ビジュアル テストは、自動テスト ソリューションの基本要素です。Silk Test Workbench では、アプリケーションのテスト中に実行する操作をビジュアル テストを使用して模倣します。ユーザーがキー入力やマウス操作でメニュー、リスト項目、ボタンなどを選択するのと同じように、アプリケーションや Web ページを制御します。ビジュアル テストを記録すると、Silk Test Workbench によりキー入力とマウス クリックがすべて生成されます。

メニューの選択、データの入力、データ処理の確認など、アプリケーションのテストに使用する操作は、ビジュアルテストの個々のステップで表されます。各ステップがビジュアルテスト内の1行になります。記録したステップは、変更したり、再生したりできます。

Silk Test Workbench では、ビジュアルテストをすばやく記録して再生することができます。ビジュアルテストを変更して、記録できないステートメントを追加したり、テストアプリケーションの変更を反映した修正を加えたりできます。また、既存のビジュアルテストからステップをカットアンドペーストして新しいビジュアルテストを作成することもできます。

ビジュアルテストを自動的に記録したり、手動で編集したり、または、自動と手動のテストを組み合わせで使用したりできます。

Silk Test Workbench VB .NET スクリプトには、一連のコード行が含まれ、ビジュアルテストには一連のテストステップが含まれます。コードの各行または各テストステップは、テストアプリケーションに対する個々の操作を実行します。

ビジュアルテストの利点

ビジュアルテストには以下のような利点があります。

- プログラミング言語に基づくスクリプトを生成せずに、テストアプリケーションに対してテストを記録し、編集できる。
- テストステップレベルで、視覚的にテストアプリケーションのコントロールに対して操作と値を設定できる。
- ビジュアルテストのテキストベースの操作ステップとその操作のオブジェクトの間関係を参照できる。
- 自動処理されるビジネスプロセスに精通しているユーザーにとって、テストプロセスでのオーナーシップが大きくなる。
- 既存のスクリプトや他のすべての既存のテスト資産をビジュアルテストで使用することにより、有効活用できる。
- テスト対象のアプリケーションにアクセスしなくても、テストアプリケーションとコントロールを再キャプチャできる。
- テストを作成し直さなくても、テストアプリケーションの更新に従って画面キャプチャを自動的に更新できる。
- フラグを使用して、テストプロジェクトの問題、割り当て、その他のテストチーム情報に関して、協業/共有できる。
- ビジュアルテストを再生することでアプリケーションをテストできる。ビジュアルテストは、アプリケーションテストのためにいつでも再生できる。ビジュアルテストでは、以下のことが可能。
 - 指定した時刻に自動的に実行する。
 - 完全に無人で実行する。
 - ユーザーとの対話を行って、ID やパスワードなどの認証情報を受け取ることができる。

スクリプトと比較したビジュアルテストの利点

スクリプトと同様に、ビジュアルテストはアプリケーションのテストに使用する操作を記録します。これらの操作には、メニューの選択、データの入力、アイコンのクリック、結果の確認などが含まれます。ビジュアルテストは Silk Test Workbench の主要なテスト単位です。テストソリューションにスクリプトだけを使用する場合と比較して、以下のような利点があります。

- ビジュアルテストでは、一連のステップが技術的ではない明確な言語で表示されるため、テスト担当者が容易に理解できます。
- ビジュアルテストはスクリプト言語に縛られないため、プログラミング言語の知識がないテスト担当者でも、詳細で強力なテストソリューションを作成できます。
- ビジュアルテストはストーリーボードにサムネイルイメージでグラフィカルに表示されるため、ビジュアルテストの全体のフローをすぐに確認できます。
- ビジュアルテストを記録したあとは、**画面プレビュー**を使用して、テストアプリケーションにアクセスしなくても、そのビジュアルテストを更新できます。

- ビジュアルテストはテストソリューションのビルドプロセスを円滑にします。たとえば、1人のテスト担当者がテストを記録してそれを別のテスト担当者に渡します。そのテスト担当者はロジック、検証、ループの他、予期しない状況に特定の方法で対処する機能を加えて、ビジュアルテストに情報を追加します。
- ビジュアルテストは、テストアプリケーションが変わった場合に、作成し直す必要がありません。テストステップをすぐに更新できます。
- 既存のビジュアルテストをコピーして変更することにより、新しい手順を簡単に作成できます。
- テストへのモジュール式アプローチが推進され、再利用率を高めることができます。他のビジュアルテストを呼び出せます。
- Silk Test Workbench では、他のテスト担当者とのテストプロジェクト情報のやり取りや協業ができます。

サンプル ビジュアルテスト

以下のサンプル ビジュアルテストでは、ビジュアルテストの一般的な機能を使用する方法を示しています。

サンプル ビジュアルテスト：アクティブ データの使用

このサンプル ビジュアルテストでは、ビジュアルテストにおけるアクティブ データの使用を示します。このビジュアルテストでは、既存のアクティブ データを使用し、**ListView** コントロールに複数の名前（姓と名）を入力することによって Windows テスト アプリケーションをテストします。ファイル内の各レコードには、姓と名が含まれています。サンプル ビジュアルテストはファイル内のデータを循環し、アプリケーションの **ListView** コントロールにデータを入力することによって Windows アプリケーションをテストします。

サンプル ビジュアルテスト：アクティブ データの使用：パート 1 シナリオ

テスト シナリオ

テスト担当者は、スプレッドシートに保存された顧客の姓と名のデータを使用して、Windows テスト アプリケーションに複数の名前レコードを入力します。テスト時に、データは Names1.xls という名前のスプレッドシートにあります。

これを行うには、以下を実行します。

- Windows テスト アプリケーションの **ListView** コントロールへのアクセスを記録し、コントロールで [First Name] および [Last Name] 列を作成してから、最初の姓と名のペアを入力します。
- ビジュアルテストに関連付けるアクティブ データ資産を作成します。アクティブ データ資産の作成時に、使用するデータが含まれる Names1.xls データ ファイルを選択します。
- そのアクティブ データ ファイルをビジュアルテストに関連付けます。
- 繰り返しロジックを作成して、テスト アプリケーションの **ListView** コントロールに姓と名を入力する記録済みステップを繰り返します。
- 姓と名を入力するステップのプロパティを変更して、アクティブ データ ファイルのデータが使用されるようにします。

サンプル ビジュアルテスト：アクティブ データの使用：パート 2 詳細

サンプル ビジュアルテスト

以下の図に、このテストシナリオに基づいた **テストステップ** ペインのサンプル ビジュアルテストステップを示します。テスト内のステップの進行を強調表示しています。

Test Steps		Actions
1	<<Start>>	
2	Using Window 'Test Application'	
3	Select menu item 'List view'	
4	Using Dialog 'List View'	
5	Enter 'First Name'	
6	Using Dialog 'List View'	
7	Select button 'Add Column'	
8	Using Dialog 'List View'	
9	Set selection range	
10	Using Dialog 'List View'	
11	Enter 'Last Name'	
12	Using Dialog 'List View'	
13	Select button 'Add Column'	
14	Repeat using activedata 'NamesList'	
15	Using Dialog 'List View'	
16	Enter '[[NamesList].Text("First Name")]'	
17	Using Dialog 'List View'	
18	Select button 'Add Item'	
19	Using Dialog 'List View'	
20	Enter '[[NamesList].Text("Last Name")]'	
21	Using Dialog 'List View'	
22	Select button 'Add Item'	
23	End Repeat	
24	<<End>>	

ステップの詳細

ビジュアルテストの各ステップの詳細は、以下のとおりです。

ステップ 1: <<開始>> ステップ。このステップのプロパティには、ビジュアルテストで使用されるアクティブデータ資産の名前が表示されます。

このテストシナリオでは、アクティブデータカテゴリの**ステップ 1 項目 1**プロパティにアクティブデータ資産の関連付けが表示されます。このアクティブデータ資産には、ファイル Names1.xls が含まれています。ビジュアルテストでは、このファイルのデータが使用されます。

以下の図に、このビジュアルテストの<<開始>>ステップの**項目 1**プロパティを示します。

Properties (1, <<Start>>)	
General	
ActiveData	1 Items
Item 1	NamesList

ステップ 2 から 3: テストアプリケーションで **List view** メニュー項目を選択して **ListView** にアクセスする、記録されたステップ。このステップのあと、繰り返しステップが開始されます。

ステップ 4 から 13: **ListView** で First Name と Last Name の各列を作成する、記録されたステップ。これらの列は、アクティブデータファイルから適切な列に姓と名を入力するためのテストを設定します。

- 説明が「Using Dialog」で始まるステップでは、データを入力するためのウィンドウおよびコントロールにテストが関連付けられます。

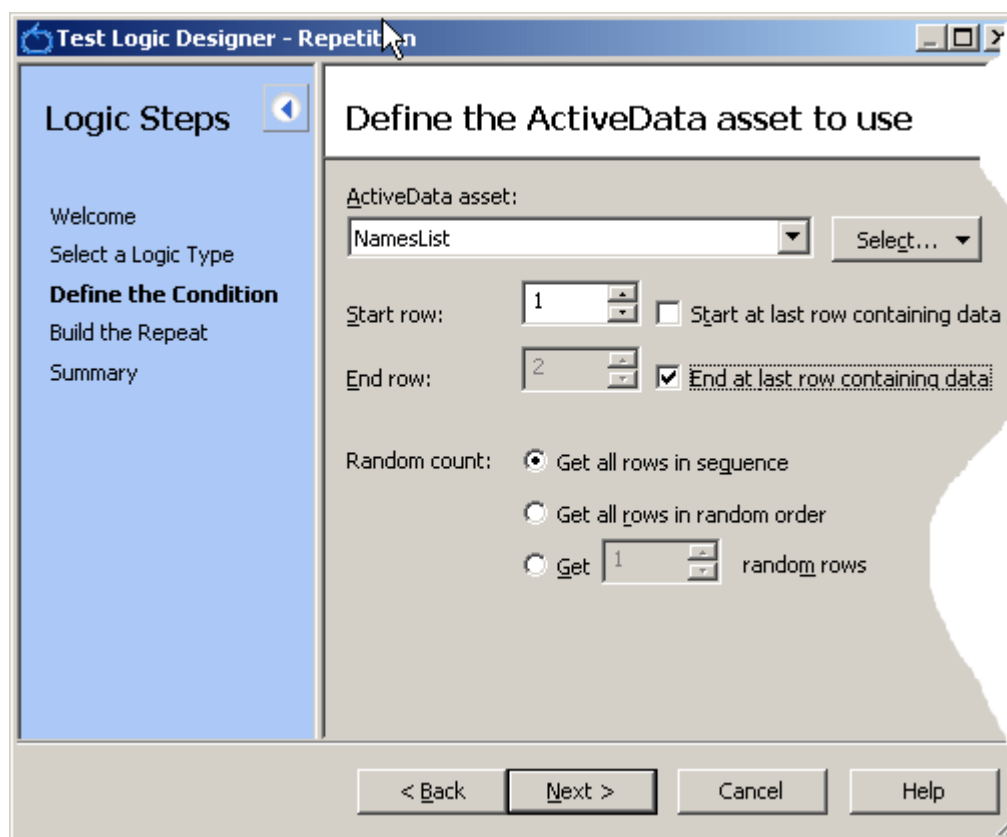
- このシーケンスの他のステップは、列に名前を付け、それを **List view** に追加します。

ステップ 14：「アクティブ データを使用して繰り返します」ステップでは、繰り返しロジックが開始します。テスト担当者は、テストを記録したあとで、このステップを作成します。このステップと繰り返しの終了ステップ（ステップ 23）の間にあるすべてのステップは、アクティブ データ ファイルの名前レコードごとに繰り返されます。使用する繰り返しロジックの種類は、**アクティブ データ ファイルを使用してステップ シーケンスを繰り返す** です。

繰り返し回数は、繰り返しロジックを作成したときに入力した **開始行**、**終了行**、および **ランダム数** の値によって決まります。また、この設定によって、アクティブ データ テストに使用するアクティブ データ ファイルのレコードも決まります。

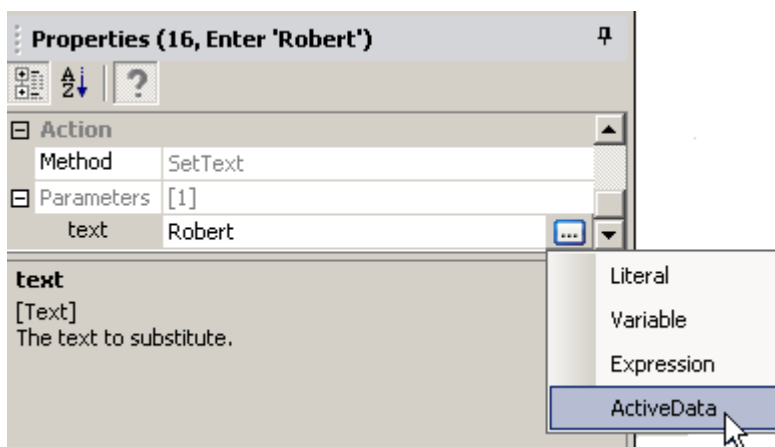
このテスト シナリオでは、データ ファイル内のすべての行を順番に使用して、アクティブ データ テストを行います。データ ファイルに 10 行のデータが含まれており、すべてが使用される場合、繰り返しロジックステップの繰り返し回数は 10 回になります。

以下の図に、このビジュアル テストの繰り返しロジックを作成したときの **テスト ロジック デザイナ** の使用する **アクティブ データ資産の定義** ページを示します。

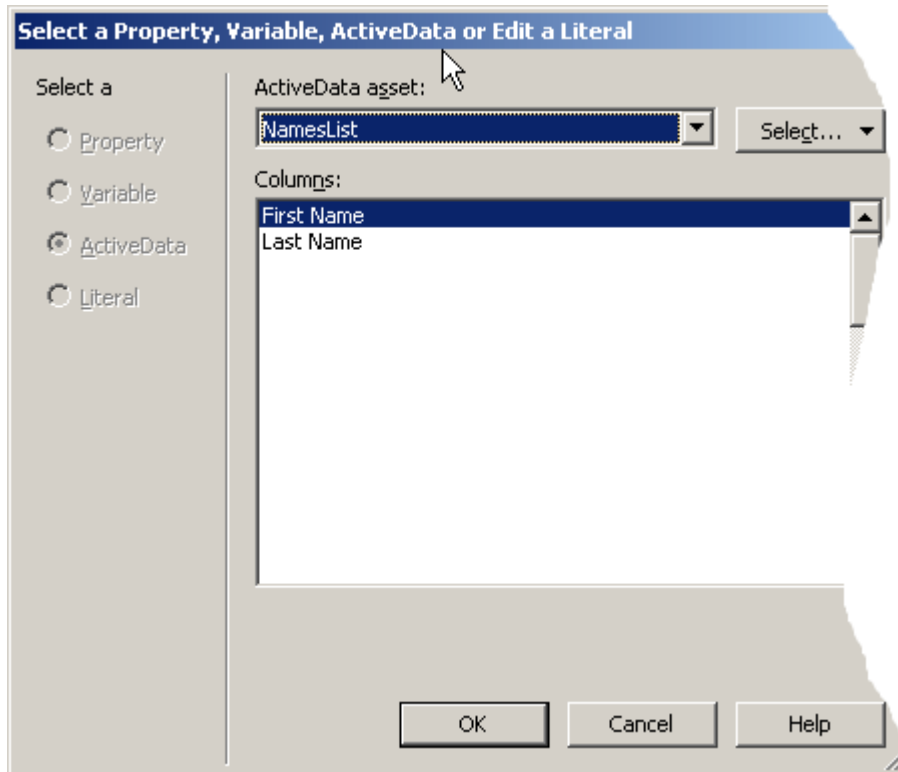


ステップ 15 から 22：アクティブ データ ファイルのデータを使用するように記録後に変更されたステップ。テスト アプリケーションの **ListView** コントロールで、[First Name] と [Last Name] の列にデータを入力します。

アクティブ データ ファイルのデータを使用するようにこれらのステップを変更すると、リテラル値ではなくアクティブ データ値が使用されるように入力値が更新されます。これを行うには、以下の図に示すように、ステップの **Text** プロパティの値領域をクリックして、メニューから **アクティブ データ** を選択します。



ダイアログ ボックスが表示され、データ ファイルを含むアクティブ データ資産を選択します。また、**Text** プロパティの入力値にマップする値を含むデータ ファイル内の列を選択します。このシナリオのステップ 16 では、Windows テスト アプリケーションの [First Name] のリテラル テキストが、アクティブ データ ファイルの **名 (First Name)** 列のデータで置き換えられます。以下の図に、このダイアログ ボックスにおける設定を示します。



この変更により、ステップ テキストには以下が表示されるようになります。

- テストで使用されるアクティブ データ資産の名前
- 使用されるデータ型 (テキスト)
- ステップで使用されるデータが含まれる列の名前

アクティブ データ ファイルのデータを使用した自動テスト ステップの変更の詳細については、「アクティブ データ ファイルのデータのビジュアル テスト データへのマッピング」を参照してください。

ステップ 23 : ビジュアル テストの繰り返しロジックの一部として作成された 繰り返しの終了 ステップ。このステップで、繰り返しロジックのループが閉じます。

ステップ 24 : <<終了>> ステップ。ビジュアル テストの再生を完了します。

サンプル ビジュアル テスト : コントロールの選択を確認するための判断ロジックの使用

このサンプル ビジュアル テストでは、Windows サンプル アプリケーションで **ComboBox** コントロールが選択されていることを確認します。テスト対象の **ComboBox** がレンダリングされたあとに選択が行われていない場合、**ComboBox** にはテキストが表示されません。テスト担当者は、判断ロジックを使用して、**ComboBox** のテキストが null であるかどうかを判断し、その結果に応じて異なる操作を実行します。

サンプル ビジュアル テスト : コントロールの選択を確認するための判断ロジックの使用 : パート 1 シナリオ

テスト シナリオ

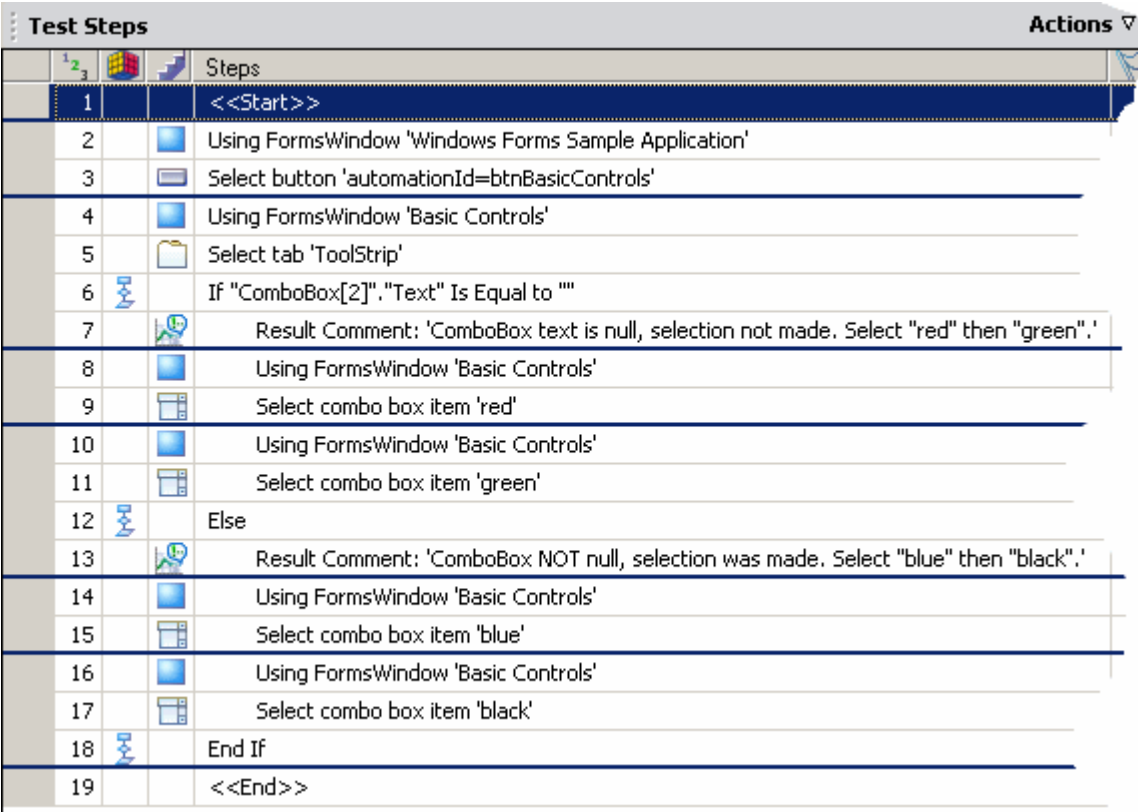
テスト担当者は、コントロールの値が以前に選択されているかどうかに基づいて、**ComboBox** に対して異なる選択を行います。**ComboBox** の既存のテキストが空（選択されていないことを示す）の場合は **ComboBox** に対する一連の操作をテストし、そうでない場合は別の操作をテストします。

これを行うには、テスト担当者は、操作を実行する前にコントロールの **Text** プロパティにテキストが含まれているかどうかを評価するようにビジュアル テストを設定します。判断ロジックでは、コントロールの **Text** プロパティのテキストの有無に基づいて、後続の操作が決まります。判断ロジックの If/Else 構文には、判断条件に基づく異なる操作を含めます。また、テスト担当者は、判断ロジックの結果と実行された操作をテスト結果にレポートする必要があります。

サンプル ビジュアル テスト : コントロールの選択を確認するための判断ロジックの使用 : パート 2 詳細

サンプル ビジュアル テスト

以下の図に、このテスト シナリオに基づいた **テスト ステップ** ペインのサンプル ビジュアル テスト ステップを示します。テスト内のステップの進行を強調表示しています。



Test Steps		Actions
1		<<Start>>
2		Using FormsWindow 'Windows Forms Sample Application'
3		Select button 'automationId=btnBasicControls'
4		Using FormsWindow 'Basic Controls'
5		Select tab 'ToolStrip'
6		If "ComboBox[2]". "Text" Is Equal to ""
7		Result Comment: 'ComboBox text is null, selection not made. Select "red" then "green".'
8		Using FormsWindow 'Basic Controls'
9		Select combo box item 'red'
10		Using FormsWindow 'Basic Controls'
11		Select combo box item 'green'
12		Else
13		Result Comment: 'ComboBox NOT null, selection was made. Select "blue" then "black".'
14		Using FormsWindow 'Basic Controls'
15		Select combo box item 'blue'
16		Using FormsWindow 'Basic Controls'
17		Select combo box item 'black'
18		End If
19		<<End>>

ステップの詳細

ビジュアルテストの各ステップの詳細は、以下のとおりです。

ステップ 1: <<開始>> ステップ。

ステップ 2: 記録された指定の画面を使用して後続の自動ステップが再生されることを示す画面ステップ。このステップが **テスト ステップ** ペインで強調表示されると、関連する画面が **画面プレビュー** に表示されます。

ステップ 3: テストアプリケーション内の **Button** コントロールをクリックする、記録されたステップ。ボタンをクリックすると、一連のコントロールがさまざまなタブに表示されます。**Toolstrip** タブには、テスト対象のコントロールが含まれています。

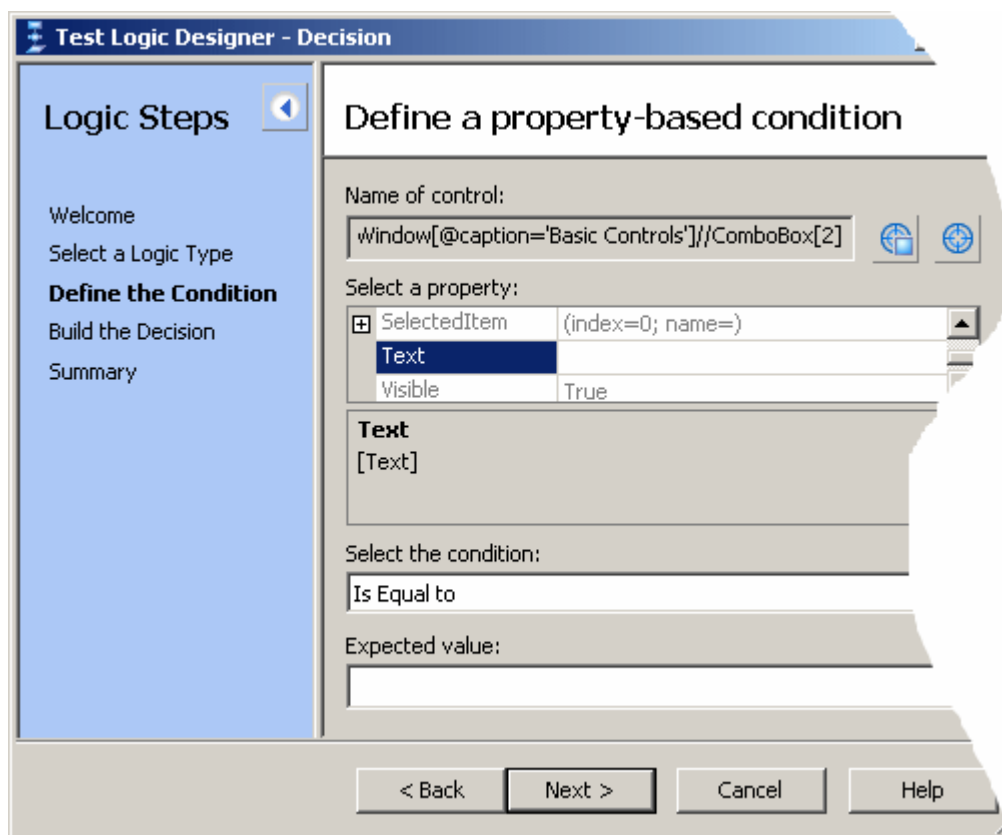
ステップ 4 から 5: 記録されたステップは、テスト対象の **ComboBox** コントロールを含む **Toolstrip** タブにアクセスして選択します。

ステップ 6: 判断ロジックを開始する If ステップ。判断ロジックの一部は、テストを記録したあとで、テスト担当者が作成します。ロジック アイコンは、このステップが判断ロジック ステップであることを示します。

判断ロジックによって、**ComboBox** コントロールの **Text** プロパティの内容が評価されます。コントロールに値が設定されていない場合、**Text** プロパティは **Nothing** または空白です。値が **Nothing** または空白の場合は、ビジュアル テストはある一連のステップを実行し、値が **Nothing** または空白以外の場合は、異なる一連のステップを実行します。

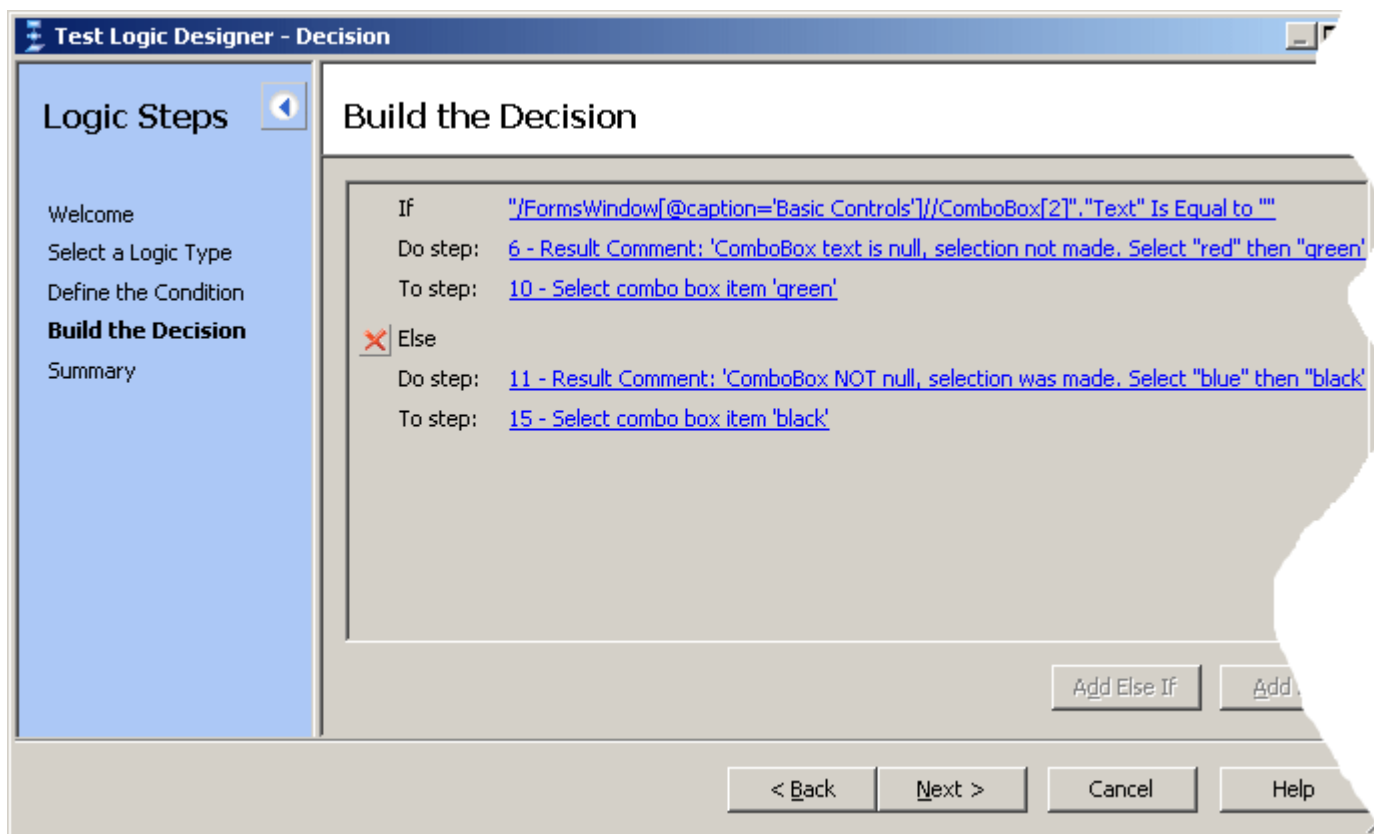
テスト ロジック デザイナ を使用して判断ロジックを設定するには、以下の操作を実行します。


- **コントロールのプロパティに基づく** ロジックの種類を選択します。
- 以下の操作を行うため、**テスト ロジック デザイナ** の [プロパティに基づく条件の定義] ページを使用します。
 - テスト対象アプリケーションを使用して、コントロールの名前を識別します。
 - **テスト ロジック デザイナ** の **プロパティの選択** リストで **テキスト** を選択します。
 - 以下の図に示すように、条件が **等しい** であり、**期待値** フィールドが空白であることを確認します。



- **テスト ロジック デザイナ** の **判断の構築** ページを使用して、条件ロジックの結果に応じて実行する一連のステップを指定します。このシナリオでは、結果が **true** の場合にステップ 6 から 10 を実行します。

次に、**Else の追加** をクリックして If ステップを判断ロジックに追加し、以下の図に示すように、結果が false の場合にステップ 11 から 15 を実行します。



 **注:** これらのステップがビジュアルテストに挿入されると、If および Else ステップ以降のステップ番号が変わります。このシナリオでは、以下のステップ番号は、前に図に表示されている番号ではなく、判断ロジックがテストに追加されたあとの番号に基づいています。

ステップ 7: テスト担当者によって作成された結果コメントステップ。このステップは、判断ロジックが true と評価される場合に実行されます。ステップが実行されると、テスト結果にメッセージが送信されます。このメッセージは、**ComboBox** にテキストがないため、以前に選択されていないことを示します。また、コントロールに使用できる値が後続のステップで選択されることも示します。

ステップ 8 から 11: 判断ロジックが true と評価される場合、これらのステップは、使用可能な異なる **ComboBox** 項目にアクセスして選択します。まず、**ComboBox** の red が選択され、次に、green が選択されます。

ステップ 12: 判断ロジックの Else ステップ。判断ロジックでは、**Else の追加** がロジックに組み込まれている場合に、このステップが作成されます。Else ステップは、判断ロジック内の If ステップの条件に一致しない場合に、判断ロジックの Else ステップと End If ステップの間にあるステップが再生されることを示します。

ステップ 13: テスト担当者によって作成された別の結果コメントステップ。このステップでは、テスト結果にメッセージが送信されます。このメッセージは、**ComboBox** テキストが Nothing ではなく、使用可能な項目が選択されることを示します。

ステップ 14 から 17: 判断ロジックが false と評価される場合、これらのステップは、使用可能な異なる **ComboBox** 項目にアクセスして選択します。まず、**ComboBox** の blue が選択され、次に、black が選択されます。

ステップ 18: ビジュアルテストの判断ロジックの一部として作成された End If ステップ。このステップで、If...Else ロジックが閉じます。

ステップ 19: <<終了>> ステップ。ビジュアルテストの再生を完了します。

プロパティ

このトピックでは、ビジュアルテストのプロパティを示します。

アクティブ データ プロパティ

アクティブ データ プロパティ カテゴリには以下のプロパティが含まれます。

保存オプション 更新されたアクティブ データを指定したアクティブ データ ファイルにすぐに保存するか、または再生中にはまったく保存を行わないかを指定します。
リストから以下のいずれかの値を選択します。

- **アクティブ データ ファイルにデータを直ちに保存する**：ビジュアル テストに関連付けられた指定のアクティブ データ資産で更新されたデータを、ステップ実行時にそれぞれのアクティブ データ ファイルに保存します。ビジュアル テストで使用されるアクティブ データは、通常、式によって更新されます。データを保存するアクティブ データ ファイルを含む資産は、**保存するアクティブ データ** プロパティで指定します。



注：ビジュアル テストに保存操作を取り消すステップが含まれていないかぎり、ビジュアル テストの再生完了時には、ビジュアル テストの再生中に更新されたアクティブ データがそれぞれのアクティブ データ ファイルに保存されます。

- **アクティブ データ ファイルにデータを保存しない**：更新されたアクティブ データは **保存するアクティブ データ** プロパティで指定したアクティブ データ資産のアクティブ データ ファイルに保存されません。

名前 **保存オプション** プロパティで選択した値に応じて、再生中に更新データがまったく保存されないファイルか、またはこのステップの実行時に更新データが保存されたファイルを含むアクティブ データ資産を指定します。詳細については、「アクティブ データ資産のビジュアル テストへの関連付け」を参照してください。

リストから適切なアクティブ データ資産を選択します。リストにはビジュアル テストに関連付けられたアクティブ データ資産が表示されます。

詳細設定プロパティ

詳細設定 プロパティ カテゴリには以下のプロパティが含まれます。

表示するボタン メッセージ ボックスに表示する 1 つまたは複数のボタンを決定します。
リストから表示するボタンを選択します。リストには以下が含まれています。

- **OK**：メッセージ ボックスに **OK** ボタンを表示します。
- **OK、キャンセル**：メッセージ ボックスに **OK** ボタンと **キャンセル** ボタンを表示します。
- **中止、再試行、無視**：メッセージ ボックスに **中止**、**再試行**、および **無視** ボタンを表示します。
- **はい、いいえ、キャンセル**：メッセージ ボックスに **はい**、**いいえ**、および **キャンセル** ボタンを表示します。
- **はい、いいえ**：メッセージ ボックスに **はい** と **いいえ** ボタンを表示します。
- **再試行、キャンセル**：メッセージ ボックスに **再試行** ボタンと **キャンセル** ボタンを表示します。

デフォルトボタン メッセージ ボックスを表示するときに自動的に選択するボタンを指定します。
リストから **デフォルト** ボタンを選択します。リストの内容は、**表示するボタン** プロパティで選択した値に基づきます。

表示するアイコン メッセージ ボックスに選択したアイコンを表示します。メッセージの内容を示す適切なアイコンを表示できます。有効な値は以下のとおりです。

- **(なし)** : メッセージ ボックスにアイコンを表示しません。これはデフォルト値です。
- **重要** : メッセージ ボックスに **重要** アイコンを表示します。このアイコンを使用して、注意が必要な重大なエラーまたは状態をテスト担当者に警告します。
- **警告の照会** : メッセージ ボックスに **警告の照会** アイコンを表示します。このアイコンを使用して、応答が必要な状態をテスト担当者に警告します。
- **警告メッセージ** : メッセージ ボックスに **警告メッセージ** アイコンを表示します。このアイコンを使用して、応答が不要なエラーまたは状態をテスト担当者に警告します。
- **通知メッセージ** : メッセージ ボックスに **通知メッセージ** アイコンを表示します。メッセージ ボックスに情報を表示するときにこのアイコンを使用します。

割り当てプロパティ

割り当て プロパティを使用して以下の設定を行います。

- 指定したコントロールの有効なプロパティを変数に割り当てます。これにより、ビジュアル テストでプロパティ値をより効率的に使用できます。これらのプロパティは「コントロールの " プロパティを変数 " に格納します」ステップで表示されます。
- グローバルの取得ステップでグローバル変数の値を格納するローカル変数を指定します。
- グローバルの設定ステップでグローバル パラメータを定義します。
- eCATT スクリプトからインポートするか、または eCATT スクリプトにエクスポートする eCATT スクリプト引数を指定します。



注: このカテゴリに表示されるプロパティはテスト ステップのタイプに基づきます。このプロパティ カテゴリを含むすべてのステップ タイプで、すべてのプロパティが使用されるわけではありません。

このカテゴリには以下のプロパティが含まれます。

グローバル変数名 (グローバルの設定ステップで) 作成または (グローバルの取得ステップで) 取得されるグローバル変数の名前を指定します。

目的のグローバル変数名を入力します。グローバル変数名に有効な文字は、大文字と小文字の英数字とアンダースコアです (「ABC」、「abc」、「_」)。グローバル変数名は一意である必要があります。



ヒント: <<開始>> ステップで作成したグローバル変数を取得する場合は、あらかじめその名前を確認しておく必要があります。グローバル変数を設定するビジュアル テストを開いて、適切なグローバルの設定ステップのグローバル変数名プロパティから名前を取得できます。

eCATT 引数名 : eCATT スクリプトからインポートするか、または eCATT スクリプトにエクスポートする eCATT スクリプト引数の名前を指定します。

値をインポートまたはエクスポートする eCATT スクリプト引数の名前を入力します。引数名は、**eCATT 引数コンテナ** ダイアログ ボックスを使用してビジュアル テストの引数コンテナに定義する必要があります。入力した名前が引数コンテナで定義された名前と一致していない場合、再生エラーが発生します。

プロパティ名 ステップで識別されるコントロールで使用可能なプロパティの名前を指定します。

ステップで識別されるコントロールが、**プロパティ** ウィンドウの **オブジェクトの種類** プロパティに表示されます。リストから使用可能なプロパティを選択して、関連する値を変数に設定します。

ローカル変数名 指定した値が格納されるローカル変数の名前を指定します。

リストからローカル変数を選択します。ローカル変数をリストに表示するには、事前にローカル変数をビジュアル テストで作成しておく必要があります。

コントロールのプロパティをローカル変数に設定するステップの場合、選択したプロパティの値が指定したローカル変数に格納されます。

eCATT スクリプト引数をインポートするステップの場合、スクリプト引数の値が指定したローカル変数に格納されます。

値 開始時にグローバル変数に指定される値。値を入力するか、または、値エリア内を右クリックして、値を割り当てます。

コマンドプロパティ

このカテゴリには以下のプロパティが含まれます。

パス名 起動するアプリケーションまたはプログラムのパス名とファイル名を指定します。
ファイルの選択 をクリックし、**ファイルの選択** ダイアログ ボックスを使用して目的のプログラムまたはアプリケーションを起動するファイルに移動し、このファイルを選択します。
実行する操作が **検索** または **探索** の場合、このプロパティではパス名のみが必要です。

操作 アプリケーションまたはプログラムが起動したときに実行する操作を指定します。
実行する操作をリストから選択します。リストには以下が含まれています。

- **開く** (デフォルト) : 指定したパラメータを使用して、指定ファイルを開きます。ファイルには、アプリケーション、ドキュメント ファイル、またはフォルダを指定できます。
- **編集** : エディタを起動して、編集するファイルを開きます。 **パス名** プロパティで指定されたファイルが、ドキュメント エディタで開くことのできるドキュメント ファイルではない場合、この操作は実行されません。
- **探索** : Windows エクスプローラを使用して、 **パス名** プロパティで指定されたフォルダを開きます。
- **検索** : 指定されたディレクトリを起点として、Windows エクスプローラを検索モードで起動します。ディレクトリは、 **パス名** プロパティまたは **ディレクトリ** プロパティを使用して指定できます。 **パス名** プロパティが使用され、かつ、ファイル名が含まれている場合は、テストを再生しても、このステップは実行されません。
- **印刷** : **パス名** プロパティで指定されたドキュメント ファイルを印刷します。 **パス名** プロパティで指定されたファイルがドキュメント ファイルではない場合、テストを再生してもこのステップは実行されません。

パラメータ アプリケーションを起動するために渡すパラメータを指定します。

パス名 プロパティで指定されたファイルがアプリケーション ファイルの場合は、このプロパティを使用して、アプリケーションに渡して起動時に使用するパラメータを含む NULL 終端文字列を指定します。 **パス名** プロパティで指定されたファイルがドキュメント ファイルまたはその他のタイプのファイルの場合は、値を空白のままにします。

ディレクトリ **操作** プロパティで指定された操作に使用できるデフォルト ディレクトリを指定します。

ウィンドウスタイル アプリケーションまたはプログラムが起動したときの表示方法を指定します。

この値は、選択したウィンドウ スタイルを表す整数としてテスト ステップに表示されます。値はリストから選択します。選択できる値は、以下のとおりです。

- **非表示** : プログラムまたはアプリケーションが起動したとき、ウィンドウは非表示です。フォーカスは、他のウィンドウに渡されます。
- **最大化** : アプリケーション ウィンドウを最大化します。
- **最小化** : 起動時にアプリケーション ウィンドウを最小化し、Z オーダーで次に最上位にくるウィンドウをアクティブにします。

- **復元** : アプリケーション ウィンドウを起動して表示します。ウィンドウが最小化または最大化されていた場合は、そのウィンドウの元のサイズと位置に復元されます。この値は、最小化されたウィンドウを復元するときに使用します。
- **表示** : アプリケーション ウィンドウを起動し、現在のサイズと位置に表示します。
- **デフォルトの表示** : アプリケーションを起動したプログラムからの情報に基づいて、ウィンドウの表示状態を設定します。
- **表示の最大化** : アプリケーションを起動し、ウィンドウを最大化して表示します。
- **表示の最小化** : アプリケーションを起動し、ウィンドウを最小化して表示します。
- **表示を最小化し、アクティブ化しない** : 最小化されたウィンドウでアプリケーションを表示します。現在のアクティブ ウィンドウは、アクティブなままです。
- **現在の状態で表示し、アクティブ化しない** : 現在のウィンドウ状態でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
- **最新の位置に表示し、アクティブ化しない** : 最後に使用したときのウィンドウ サイズと位置でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
- **通常表示** : アプリケーションを起動して、ウィンドウを表示します。最後に使用したウィンドウの位置と最小化または最大化の状態が復元されます。この値は、はじめてウィンドウを表示するときに使用します。

遅延プロパティ

このカテゴリには以下のプロパティが含まれます。

遅延タイプ ステップの遅延間隔を設定します。

プ リストから **秒** または **ミリ秒** を選択します。

遅延時間 遅延タイプに基づいて遅延時間を指定します。

リテラル値を入力するか、あるいは **選択** をクリックして変数、式の結果、またはアクティブ データ値を割り当てます。

フラグの設定プロパティ

フラグの設定 プロパティ カテゴリには以下のプロパティが含まれます。

エラー時のフラグの作成 エラー発生時にフラグを作成するかどうかを指定します。このプロパティを編集するには、値を **True** または **False** に設定します。 **True** に設定した場合、フラグが作成されます。 **False** に設定した場合、フラグは作成されません。

フラグの説明 エラー時に作成されるフラグの説明を含めます。説明は、ビジュアル テストの結果と **開始画面** に表示されます。このプロパティを編集するには、フラグの説明を入力します。

全般プロパティ

全般 プロパティ カテゴリには以下のプロパティが含まれます。

メッセージボックスのタイトル メッセージ ボックスのタイトル バーに表示するテキストを決定します。デフォルト値は、Silk Test です。

デフォルトを変更するには、値を入力するか、**選択** をクリックして、メッセージ ボックスのタイトルとして値を割り当てます。

メッセージテキスト メッセージ ボックスに表示されるテキストの内容を指定します。事実上、ビジュアル テストで使用可能ななどの情報でもメッセージの内容に使用できます。

メッセージの内容を直接入力するか、**選択** をクリックし、メッセージ テキストとして値を割り当てます。

リテラルテキストと変数に返されたデータなどの情報を組み合わせるには、**選択** をクリックしてメニューから **式** を選択します。**式デザイナー** を使用して、文字列の組み合わせを作成します。

たとえば、説明テキストと ComboBox の項目数を表示する場合、以下のような式になります。

```
"Message box text is: " + ToString (intCount)
```

ここで、各値の内容は以下のようになります。

"Message box text is: " は説明テキストの文字列です。

+ は文字列の組み合わせに使用する演算子です。

ToString (intCount) は ToString () 変換関数を使用して文字列に変換される、コントロール intCount 内の項目数を保持する変数です。

アクティブデータ データ ファイルがビジュアル テストに関連付けられているアクティブ データ資産の数を表示します。

値エリアで **アクティブデータの参照** をクリックし、**アクティブデータの参照** ダイアログ ボックスを使用して資産を選択することによって、アクティブ データ資産をビジュアル テストに関連付けることができます。

項目 n ビジュアル テストに関連付けられているアクティブ データ資産の名前を表示します。

再生設定の取得プロパティ

再生設定の取得 プロパティ カテゴリには以下のプロパティが含まれます。

再生設定 個々の再生情報を制御します。

「再生設定 " を取得して、ローカル変数 " に格納します」ステップでは、値は再生設定で、この再生設定の現在の設定値がローカル変数に格納されます。値は以下のいずれかから取得されます。

- 再生オプションに設定されている再生設定の現在値。
- 定義済みの実行環境（ビジュアル テストで **RunEnvironment** 再生設定が設定されている場合）。
- ビジュアル テストの **再生設定の設定** ステップ。

リストから **再生設定** を選択します。

ローカル変数名 値が格納されるローカル変数の名前を指定します。

「再生設定 " を取得して、ローカル変数 " に格納します」ステップでは、選択した再生設定の値が指定したローカル変数に格納されます。リストからローカル変数を選択します。ローカル変数をリストに表示するには、事前にローカル変数をビジュアル テストで作成しておく必要があります。

その他のプロパティ

その他 プロパティ カテゴリには以下のプロパティが含まれます。

説明 選択したテスト ステップの説明を表示します。このプロパティを編集するには、プロパティ値 テキスト ボックスに選択したテスト ステップの説明を入力します。

最新の変更 テスト ステップが最後に変更された日付と時刻を表示します。このプロパティは読み取り専用です。

オブジェクト プロパティ

オブジェクト プロパティ カテゴリには以下のプロパティが含まれます。

- クラス名** 選択されたオブジェクトのクラス名を識別します。
- ロケータ** テストするオブジェクトを識別します。 **画面プレビュー** ペインまたは実際のアプリケーションを使用すると、テストするオブジェクトを識別できます。あるいは、リテラル値、式、アクティブ データ、または変数を選択し、ロケータとして使用できます。
- プロパティ** 指定したオブジェクトから取得できるプロパティ、および変数への割り当てが可能なプロパティのリストを返します。リストからプロパティを選択します。あるいは、リテラル値、式、アクティブ データ、または変数を選択し、プロパティとして使用できます。
- ローカル変数名** 指定した値が格納されるローカル変数の名前を指定します。

エラー時の移動先プロパティ

エラー時の移動先 プロパティ カテゴリには以下のプロパティが含まれます。

実行する操作 ビジュアル テストの再生中にエラーが発生した場合に、実行する操作を設定します。リストから以下のいずれかのオプションを選択します。

- **デフォルトの再生エラー ダイアログを表示します : 再生エラー** ダイアログ ボックスを表示し、そこでそのエラーに対して実行する操作を決定します。 **再生エラー** ダイアログ ボックスを使用してデバッグ モードに入り、ビジュアル テストでエラーを診断します。
再生エラー ダイアログ ボックスの表示は、再生でエラーが発生した場合のデフォルトの動作です。
- **再生の終了** : エラーが発生したときに、ビジュアル テストの再生を終了します。ビジュアル テストのその他のステップは再生されません。再生が終了すると、 **再生完了** ダイアログ ボックスが表示され、そこで再生の完了時に実行する操作を決定します。
- **テストの終了 : 資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップで終了します。再生が終了すると、 **再生完了** ダイアログ ボックスが表示され、そこで再生の完了時に実行する操作を決定します。
- **ステップの再試行 : 資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップの実行を試行します。 **資産** プロパティにビジュアル テストが指定されていない場合、単にエラーが発生したステップの再実行を試行します。 **再試行の回数** プロパティには、再生を試行する回数を指定できます。デフォルト値は、5 です。指定した再試行の回数に到達した時点でステップ実行に失敗すると、再生エラーが生成されます。



注: 再生エラーは、ステップを再実行している間は Silk Test Workbench は生成しません。 **再試行の回数** プロパティに 0 を設定すると、ステップは永久に再試行されます。これにより、再生時に失敗したステップが継続して再試行されている間、Silk Test Workbench が応答しなくなったように見える場合があります。

この **ステップの再試行** 操作は、ビジュアル テストで、エラーを確実に処理したり、テストで問題が予想される潜在的なエラーを修正したりするために役立ちます。たとえば、ビジュアル テストでアクセスするブラウザ アプリケーションのページが、確実に読み込まれていない可能性がある場合、「アタッチの失敗」エラーが発生することがあります。 **ステップの再試行** 操作では、ビジュアル テストを、ページを確実に読み込んでからそのページとの対話を試行する別のビジュアル テストに移行させることによって、再生エラーを解決できます。

- **再開して次に進む : 資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップの直後のステップの実行を試みます。エラーの原因となったステップは、再実行されません。
- **移動 : 資産** プロパティに指定されているビジュアル テストを再生し、 **ラベル** プロパティに指定されている事前定義のラベル ステップに移動して、再生を続行します。 **移動** を使用すると、ビ

ビジュアルテストで、再生中に発生したエラーと同様のエラーが発生する可能性のあるステップをスキップして、テストの別のセクションに移動できます。



注: エラー処理に使用された特定のラベルステップが、エラーが発生するステップの前にある場合、関係するステップがループして繰り返し再生されるため、そのエラーは繰り返し発生します。

資産 ビジュアルテストの再生中にエラーが発生した場合に実行するビジュアルテストを指定します。別のビジュアルテストを選択するには、**ビジュアルテストの参照** をクリックしてから、ビジュアルテストを選択します。



ヒント: 資産を指定してステップを作成したあとに **実行する操作** プロパティで別の操作を選択すると、ステップの動作を変更できます。たとえば、**実行する操作** プロパティを **テストの終了** に設定し、**資産** プロパティで資産を選択した場合、ステップテキストは次のようになります：

エラー時にビジュアル テスト '[ビジュアル テスト]' を再生して 'テストの終了'

ここで **実行する操作** プロパティの値を変更すると、ステップの動作を変更できます。値を **ステップの再試行** に変更すると、ステップは次のようになります：

エラー時にビジュアル テスト '[ビジュアル テスト]' を再生して 'ステップの再試行'

ラベル エラーが発生した場合にビジュアルテストが移動できる使用可能なラベルステップのリストが含まれます。リストからラベルステップの名前を選択します。

このプロパティは、**実行する操作** プロパティが **移動** に設定されている場合にのみ、表示されません。

ビジュアルテストの再生設定

ビジュアルテストの再生動作の構成には、以下の再生設定を使用できます。

アプリケーション準備完了タイムアウト 新しく起動したアプリケーションが準備完了状態になるまで待機する時間をミリ秒で指定します。指定したタイムアウト時間内にアプリケーションの準備が整わない場合は、Silk Test Workbench で例外が発生します。

ウィンドウを閉じるためのボタン CloseSynchron メソッドでウィンドウを閉じるために使用するボタンを指定します。

確認ダイアログに対するボタン CloseSynchron メソッドでウィンドウを閉じる際に表示された確認ダイアログ ボックスを閉じるために使用するボタンを指定します。

無応答のアプリケーションを閉じる 無応答のアプリケーションを閉じるかどうかを指定します。タイムアウトなどの理由で、エージェントとアプリケーション間の通信に失敗した場合に、アプリケーションが無応答になります。

ウィンドウを閉じるためのタイムアウト ウィンドウ閉じるための次の方式を試行する前に待機する時間をミリ秒で指定します。最終的に失敗と判断する前に、4種類の方式が Agent によって実行されます。つまり、閉じるのに失敗するまでにかかる合計時間は、指定した値の4倍の時間になります。

コントロールのキャプチャ 再生時に Silk Test Workbench がコントロールとそのプロパティをキャプチャできるようにします。記録オプションでコントロールのキャプチャ設定をいいえに設定した場合は、このオプションが上書きされ、再生時にコントロールがキャプチャされます。



注: 記録の **画面キャプチャ** 設定を **なし** に設定した場合は、**コントロールのキャプチャ** を **はい**、再生の **画面キャプチャ** 設定を **なし** 以外の値に設定して、再生時にコントロールをキャプチャする必要があります。

オブジェクトがアクティブであることを保証する	ターゲット オブジェクトがアクティブであることを保証します。
再生中にオブジェクトをハイライトする	現在のオブジェクトが再生中にハイライトされるかどうかを指定します。
キーボード イベント遅延	再生時のキー入力間の遅延をミリ秒で指定します。 テストするアプリケーションに応じて、選択する最適な値が異なることに注意してください。たとえば、Web アプリケーションをテストする場合、1 ミリ秒に設定すると、ブラウザが極端に遅くなります。ただし、この値を 0 (ゼロ) に設定すると、基本的なアプリケーション テストに失敗する可能性があります。
ダイアログを閉じるためのキー	CloseSynchron メソッドでウィンドウを閉じる際に表示されたダイアログ ボックスを閉じるキー シーケンスを指定します。例：<ESC>、<Alt+F4>。
ウィンドウを閉じるためのメニュー項目	CloseSynchron メソッドでウィンドウを閉じるために使用するメニュー項目を指定します。例：「ファイル/*終了」、「ファイル/*中止」。
マウス イベント遅延	各マウス イベントの前に使用される遅延をミリ秒で指定します。
オブジェクト有効化タイムアウト	再生中にオブジェクトが有効になるまで待機する時間をミリ秒で指定します。オブジェクトが有効になるとすぐに、Silk Test Workbench はこのオブジェクトと対話することができます。たとえば、オブジェクトがボタンの場合、このボタンが有効になると、Silk Test Workbench でボタンをクリックできるようになります。
オブジェクト解決タイムアウト	再生中にオブジェクトの解決を待機する時間をミリ秒で指定します。オブジェクトが解決されるとすぐに、Silk Test Workbench はこのオブジェクトを認識することができます。
再生モード	コントロールの再生方法を定義します。「低レベル」を使用すると、マウスとキーボードを使用して各コントロールが再生されます。「高レベル」を使用すると、API を使用して各コントロールが再生されます。コントロールごとにデフォルトの再生モードが割り当てられています。デフォルトの再生モードが選択されると、各コントロールごとのデフォルトの再生モードが使用されます。デフォルト モードを使用すると、最も信頼できる結果が得られます。「低レベル」または「高レベル」の再生を選択すると、すべてのコントロールの再生モードが選択した再生モードで上書きされます。
結果合格条件 (パーセンテージ)	今後の実行の成功を定義する全般条件を設定します。たとえば、結果の合格条件を 90% に設定した場合、10 件の検証のうち 9 件に合格しないと、再生の結果が合格になりません。
すべての情報を保存	再生が完了したあとに、ビジュアル テストのすべての情報を保存するかどうかを指定します。
画面キャプチャ	Silk Test Workbench で再生中にテスト アプリケーションの画面をキャプチャする方法を指定します。 以下のオプションがあります。 <ul style="list-style-type: none"> • 記録と同じ: ビジュアル テストが記録されたときに 記録 オプションに存在した 画面キャプチャ 設定と同じ設定を使用します。 • アクティブ ウィンドウ: 記録中にテスト アプリケーションのアクティブなウィンドウのみをキャプチャします。 • アプリケーション: テスト アプリケーションとテスト アプリケーション内のすべてのウィンドウをキャプチャします。

- **デスクトップ** : テスト アプリケーションを含む、記録中にデスクトップに表示されているすべてのアプリケーションの画面をキャプチャします。
- **なし** : 記録中に画面をキャプチャしません。



注: この設定を **なし** に設定すると、**コントロールのキャプチャ** および **画面キャプチャ (テスト ステップ)** 設定に指定した値に関係なく、コントロールまたは画面は **結果** ウィンドウにキャプチャされません。

画面キャプチャ (テスト ステップ)

Silk Test Workbench が記録中にテスト アプリケーションの個別の画面をキャプチャするかどうかを決定します。これによって、**テスト ステップ** ペインの個々のテスト ステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較できます (記録時にもテスト ステップの画面がキャプチャされた場合)。

以下のオプションがあります。

- **記録と同じ** : ビジュアル テストが記録されたときに **記録** オプションに存在した **画面キャプチャ (テスト ステップ)** 設定と同じ設定を使用します。
- **はい** : 記録中に画面をキャプチャします。 **画面キャプチャ** 設定を使用して、画面をキャプチャする方法を決定します。
- **いいえ** : 再生中に個別のテスト ステップをキャプチャしません。 **画面キャプチャ** 設定が **なし** 以外に設定された場合、個別のテスト ステップに関連付けられたグループ画面が **画面プレビュー** に表示されます。

スクリーンショット遅延

新しいウィンドウにフォーカスを設定してから、ウィンドウのスクリーン ショットを取得するまで待機する時間をミリ秒で入力します。

同期除外リスト

ページ同期中に除外するサービスまたは Web ページの URL を指定します。AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を **同期除外リスト** 設定で指定します。

複数の URL をカンマで区切って指定できます。 `http://test.com/timeService` のように URL 全体を入力したり、 `timeService` のように URL の一部を入力します。

同期モード

サポートする同期モード (HTML または AJAX) を設定します。HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれるより複雑なシナリオが使用される場合は、WaitForObject、WaitForProperty、WaitForDisappearance、または WaitForChildDisappearance などの同期関数を使用して手動でスクリプトを記述することが必要になる可能性があります。AJAX モードを使用すると、同期関数を手動で記述する必要はなくなります。

同期タイムアウト

オブジェクトが準備完了状態になる最大時間をミリ秒で指定します。

無応答のアプリケーションのタイムアウト

応答のない再生操作をキャンセルするまでのタイムアウト値をミリ秒で指定します。

挿入されたスクリプトまたはビジュアル テストの再生設定

再生設定 は、ビジュアル テスト内で実行されているビジュアル テストまたはスクリプトで使用される再生設定を指定します。リストから以下のいずれかの値を選択します。

システムのデフォルトを使用	各再生オプションについて、 オプション ダイアログ ボックスに表示されているデフォルトを使用します。
親の設定を継承する	現在の再生の実行に、親ビジュアル テストの再生設定を使用します。
親から設定をコピー	現在の再生の実行および今後すべての再生の実行に、親ビジュアル テストの再生設定を使用します。

結果プロパティ

結果 プロパティ カテゴリには以下のプロパティが含まれます。

ステップ	テスト ステップ ペインで選択したテスト ステップの名前を表示します。
結果	テスト ステップ ペインで選択したステップの結果を表示します。
結果の詳細	テスト ステップ ペインで選択したステップの結果の詳細を表示します。
行番号	テスト ステップ ペインで選択したステップの行番号を表示します。
コマンドの詳細	テスト ステップ ペインで選択したステップのログの詳細を表示します。
日付/時刻	テスト ステップ ペインで選択したステップの日付と時刻を表示します。
ミリ秒	テスト ステップ ペインで選択したステップのミリ秒数を表示します。
名前	テスト ステップ ペインで選択したステップの資産名を表示します。
ユーザー名	ビジュアル テストを再生したユーザーの名前を表示します。
マシン名	ビジュアル テストが再生されたコンピュータの名前を表示します。

再生設定の設定プロパティ

再生設定の設定 プロパティ カテゴリには以下のプロパティが含まれます。

再生設定	リストから 再生設定 を選択します。
値	<p>選択した再生設定に値を割り当てます。</p> <p>割り当てられた値は、再生設定の設定 ステップを含むビジュアル テストでのみ使用されます。</p> <p>選択した再生設定に従って、リストから値を選択するか、選択 をクリックし、再生設定値として値を割り当てます。</p>

タイマープロパティ

タイマー プロパティ カテゴリには以下のプロパティが含まれます。

タイマーの動作	<p>タイマー テスト ステップの操作を設定します。</p> <ul style="list-style-type: none"> • 開始 : タイマーを開始します。タイマーが停止すると、タイマーは 0 にリセットされます。すでに実行されているタイマーを開始しようとすると、エラーが発生します。 • 停止 : タイマーを停止します。タイマーには時間が保持され、必要な場合は 再開 操作を使用してタイマーを再開できます。 • 再開 : 停止 操作を使用して停止されたタイマーを再開します。タイマーは以前に記録された時間からカウントを再開します。 • リセット : 時間を 0 にリセットします。
----------------	---

タイマー番号 タイマーの番号を設定します。複数の条件の時間を計るときに、各タイマーに一意的な番号を割り当ててタイマーを区別できます。リストからタイマーの番号を選択します。ビジュアル テスト

には、最大で 10 個のタイマーを組み込むことができます。このプロパティは、一連のステップの実行にかかる時間を検証する場合のみ有効です。

検証結果の説明プロパティ

検証結果の説明プロパティ カテゴリには以下のプロパティが含まれます。

検証の '合格 テキスト' 指定した検証ロジックの条件に適合した場合にビジュアル テストの結果説明に表示するメッセージが含まれます。このプロパティを編集するには、プロパティを選択して、必要な説明を入力します。

検証の '失敗 テキスト' 指定した検証ロジックの条件に適合しなかった場合にビジュアル テストの結果説明に表示するメッセージを作成します。このプロパティを編集するには、プロパティを選択して、必要な説明を入力します。

待機プロパティ

待機プロパティ カテゴリには以下のプロパティが含まれます。

ロケータ ステップが、どのオブジェクトが存在または消滅するまで待機するかを指定します。以下のいずれかの方法を使用して、コントロールを識別します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別ダイアログ** ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- **選択**：このボタンをクリックして、リテラル、変数、式の結果、またはアクティブ データ値を割り当てます。

待機の種類 オブジェクトが存在、または消滅するまで待機する時間を指定します。

タイムアウト オブジェクトが存在、または消滅するまで待機する最大時間をミリ秒で指定します。このプロパティを編集するには、プロパティを選択して数値を入力します。デフォルト値は、**デフォルトの待機タイムアウト (ミリ秒)** オプション値に設定されています。タイムアウト期間内に一致するオブジェクトがない場合は、再生エラーが発生します。

スクリプトの作成

Silk Test Workbench は、ビジュアル テストと同じようにスクリプトを使用して、アプリケーションのテスト中に実行する操作を模倣します。ユーザーがキー入力やマウス操作でメニュー、リスト項目、ボタンなどを選択するのと同じように、アプリケーションや Web ページを制御します。記録中、Silk Test Workbench によりアプリケーションをテストするキー入力やマウス クリックがすべて生成されます。

Silk Test Workbench のスクリプト言語は、Microsoft .NET Framework 環境で実行されるすべてのアプリケーションを完全に制御できる堅牢なプログラミング言語である、Microsoft の Visual Basic です。.NET スクリプトは、高水準プログラミング言語が持つ諸機能の他、ソフトウェアの制御とテスト専用に設計された機能も備えています。この言語を使用すると、以下のようなスクリプトを作成できます。

- 指定した時刻に自動的に実行する。
- 完全に無人で実行する。
- ユーザーとの対話を行って、ID やパスワードなどの認証情報を受け取ることができる。

メニューの選択、データの入力、処理の確認など、アプリケーションのテストに使用する操作は、スクリプトで VB.NET コマンドとして表されます。これらのコマンドはスクリプトに挿入され、変更や再生が可能です。

Silk Test Workbench により、スクリプトをすばやく記録し実行することができます。スクリプトには、記録できない「ハンドコーディング」のステートメントを追加したり、テストアプリケーションの変更を反映した修正を加えたりできます。また、既存のスクリプトからコードをカット アンド ペーストして新しいスクリプトを作成することもできます。また、**オブジェクトの識別** ダイアログ ボックスを使用してロケータまたは個々のオブジェクトのオブジェクト マップ項目を記録し、それをスクリプトで使用することもできます。

スクリプトをすばやく作成するには、記録機能を使用するのが最適の方法です。記録機能により、Silk Test Workbench が生成するスクリプトのコーディング方法を学習することができます。これによりスクリプトの作成の基本を理解し、さらに効率的なテスト ソリューションの作成、スクリプトのモジュール化、独立したコード モジュールの作成についての理解を深めることができます。

スクリプトを使用する利点

スクリプトでは、アプリケーションをテストするために、操作がオブジェクト レベルで記録されます。これらの操作には、メニューの選択、データの入力、アイコンのクリック、結果の確認などが含まれます。テスト手順をスクリプトで管理すると、以下の利点があります。

- スクリプトは高度に拡張可能であるため、機能と柔軟性を強化できます。.NET スクリプトの機能を使用して、複雑なテスト ケースに対する複雑なソリューションを開発できます。
- スクリプトでは、外部アセンブリで提供される機能を活用して、スクリプト内で使用できる機能を拡張できます。
- 開発者とテスト担当者が、共通スクリプト言語 VB.NET を共有できます。
- テストアプリケーションが変更された場合に、スクリプトを最初から作成し直す必要がありません。
- 既存のスクリプトをコピーして変更することにより、新しいテスト手順を簡単に作成できます。
- テストへのモジュール式アプローチが推進され、再利用率を高めることができます。他のスクリプトまたは機能呼び出すことができます。
- スクリプトをループさせて、プロセスを繰り返しテストできます。
- 予想結果を検証し、予想しない状況が発生した場合にそれに対処する、インテリジェントなテストを迅速に作成できます。
- テストプロセスを文書化できます。
- テストアプリケーションが完了する前に、スクリプトを作成できます。

テスト スクリプト作成のベスト プラクティス

テスト ケースの書き方によっては、テスト セットのパフォーマンスと安定性に非常に大きな影響を与える場合があります。記録時に、可能な限り速く安定したスクリプトを Silk Test Workbench は作成します。しかし、テスト スクリプトを手動で作成したり編集する必要がある場合もあります。このトピックでは、いくつかの一般的なガイドラインを提示します。このガイドラインに従うことによって、保守性、再利用性の高いテスト スクリプトの作成、およびテストの安定性が高まるでしょう。

- テストの名前の一貫性を保ち、内容のわかりやすい名前をつけてください。テスト対象アプリケーションとテストの機能を表す名前にすると良いでしょう。たとえば、テストの名前を *MyApp_SuccessfulLogin* や *MyApp_FailingLogin* とすると、*Untitled_42* や *Untitled_43* とするよりも、他のユーザーにとって非常にわかりやすいものになります。
- できるだけ細かくテスト ケースの説明をコメントに記述してください。テスト ケースについての説明が普通の言葉で記述されていないと、実装されたコードを変更する必要がある人が、そのテストが正確に何を行っているのかを理解することが困難になります。
- テスト ケースを開始する時に、テスト対象アプリケーションが確実に適切な状態になるようにしてください。テスト ケースの操作を実行する前に、テスト対象アプリケーションを正しい状態に戻します。

- テストケースを完了する時に、テスト対象アプリケーションが確実に適切な状態になるようにしてください。追加のテストがそのテストの結果に依存する場合は、それらが開始できる状態にします。テストケースの操作が実行される時に、テスト対象アプリケーションを正しい状態に戻します。
- 可能な限り、テストケースは他のテストケースの結果に依存しないようにしてください。これが不可能な場合、テストケースが正しい順番で実行されるようにしてください。
- 機能的な流れだけでなく、テスト対象アプリケーションの正確さをテストするために、テストに検証を追加してください。
- キーワード駆動テストを使用して、操作セットの再利用性を高めてください。共通に使用する操作をキーワードにまとめ、頻繁に同じ順番で実行されるキーワードをキーワードシーケンスに結合し、キーワードやキーワードシーケンスの組み合わせをキーワード駆動テストとして実行します。
- テストの保守性と再利用性を保つためには、複雑なテストケースを記述するよりも、複数の単純なテストケースを記述して組み合わせる方が好まれます。
- テストセットの冗長性を避けるためには、新しいテストケースを追加するよりも、既存のテストケースを更新することが好まれます。

スクリプトの構文

記録されたスクリプトの行の構文を理解すると、スクリプトが読みやすくなります。オブジェクトをテストするコマンドは、同一の構文要素で構成されています。構文要素には、アプリケーションウィンドウ(アプリケーションのメインウィンドウやダイアログボックスなど)のための With ステートメントに続き、クラス、属性、操作を識別するロケーターまたはオブジェクトマップ項目が含まれます。

オブジェクトマップの使用

オブジェクトマップはテスト資産の一種であり、コントロールまたはウィンドウのロケーターではなく、コントロールまたはウィンドウに論理名(エイリアス)を関連付ける項目が含まれています。

Silk Test Workbench では、スクリプトを記録すると、デフォルトで、スクリプトコンテキストにオブジェクトマップ項目が組み込まれます。

以下の例は、Web アプリケーションをテストするスクリプトで記録される一般的な操作を示しています。

```
With _desktop.BrowserApplication("webBrowser")
  With .BrowserWindow("browserWindow")
    .DomListBox("quickLinkJumpMenu").Select("Auto Quote")
  End With
End With
```

With_ desktop.<application> の部分は、メインアプリケーションウィンドウを示しています。

上記の例の DomListBox の部分は、使用するクラスを示しています。

("quickLinkJumpMenu") の部分は、オブジェクトの属性を示しています。この場合、属性はリストボックスのリンクメニューを示しています。一方、Select() の部分は、オブジェクトに対して実行する操作またはコマンドを示しています。

最初の With コマンドが定義されたら、そのウィンドウに対する追加の呼び出しでコマンドを繰り返す必要はありません。テストする追加の各ウィンドウに対して、With ステートメントを指定する必要があります。たとえば、以下のコードは、同じスクリプト内から複数のウィンドウを呼び出す方法を示しています。

```
With _desktop.Window("untitledNotepad")
  .MenuItem("aboutNotepad").Select()
  With .Dialog("aboutNotepadDialog")
    .PushButton("ok").Select()
  End With
End With
```

オブジェクトマップ項目はかっこ引用符 ("") で囲まれ、ロケーターのキャプションを使用する必要がなくなります。

ローケータの使用

Silk Test Workbench では、識別されたオブジェクトのリテラル参照はローケータと呼ばれます。Silk Test Workbench では、スクリプトを記録すると、デフォルトで、スクリプト コンテキストにオブジェクト マップ項目が組み込まれます。オブジェクト マップをオフにすると、スクリプトの記録時に、オブジェクト マップ項目ではなくスクリプト コンテキストにローケータが含まれます。

以下の例は、Web アプリケーションをテストするスクリプトで記録される一般的な操作を示しています。

```
With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")
    .DomLink("@textContents='Court: Gender pay lawsuit can go to trial']").Select()
End With
```

With_ desktop.<application> の部分は、メイン アプリケーション ウィンドウを示しています。

最初の例の DomLink の部分と 2 番めの例の MenuItem の部分は、使用するクラスを示しています。

("@textContents='Court: Gender pay lawsuit can go to trial'") の部分は、オブジェクトの属性を示しています。この場合、属性はテキスト コンテンツを示しています。一方、Select() の部分は、オブジェクトに対して実行する操作またはコマンドを示しています。

最初の With コマンドが定義されたら、そのウィンドウに対する追加のローケータ文字列でコマンドを繰り返す必要はありません。テストする追加の各ウィンドウに対して、With ステートメントを指定する必要があります。たとえば、以下のコードは、同じスクリプト内から複数のウィンドウを呼び出す方法を示しています。

```
With _desktop.Window("@caption='Untitled - Notepad'")
    .MenuItem("@caption='About Notepad']").Select()
    With .Dialog("@caption='About Notepad'")
        .PushButton("@caption='OK']").Select()
    End With
End With
```

caption 属性は、テストするオブジェクトを示しています。caption は、アプリケーションのメイン ウィンドウ ("@caption='Untitled - Notepad'") に続き、メニュー項目 ("@caption='About Notepad'")、後続のダイアログ ボックス ("@caption='About Notepad'")、クリックするボタン ("@caption='OK'") を示しています。

ドライバ スクリプト

ドライバ スクリプトは、テスト アプリケーションをテストするための別のスクリプトをトリガし、それを監視します。ドライバ スクリプトでは、条件ロジックまたは他の制御ロジックを使用してドライバ内の他のテストのオプションの実行を管理することはありますが、通常、ドライバ スクリプト自体はテストを実行しません。

1 つのテストスイートには、テスト アプリケーション内の異なる時点で実行される、多数のテストが含まれています。テストスイートにある各スクリプトは、ドライバ スクリプトによって呼び出すことができます。ドライバ スクリプトは、スクリプトと同様に作成しますが、テスト アプリケーションの実際のテストは行いません。テスト アプリケーションをテスト開始の正しいポイントまで「ドライブ」(駆動) し、1 つまたは複数のスクリプトを呼び出してそのテスト サイトでテストを開始するだけです。テストが終了すると、ドライバ スクリプトに制御が戻され、次のテスト サイトに進みます。ドライバ スクリプトの作成は、個々のスクリプトを手動で記録する場合と同様に行うことができます。

ドライバ スクリプトは、アプリケーションのテストに対するモジュール式アプローチを推進するものです。完全なテストを実行する 1 つのスクリプトを作成する代わりに、テスト フロー全体のうちの個々の部分をテストする再利用可能なスクリプトを複数作成して、ドライバ スクリプトで各部分呼び出すようにすることができます。テスト全体のうちの一部を変更する場合は、変更が必要なスクリプトだけを更新します。ドライバ スクリプトを変更する必要はありません。

適切に設計されたスクリプトは、完了前にテスト アプリケーションを元のテスト サイトに戻します。これには以下のような目的があります。

- ドライバ スクリプトが常に停止した箇所から再開できるようにする。
- ドライバのパスを変更せずに、テスト サイトでテストを追加したり削除したりできるようにする。

ドライバ スクリプトの作成

ドライバ スクリプトは、アプリケーションのテストに対するモジュール式アプローチを推進するものです。完全なテストを実行する 1 つのスクリプトを作成する代わりに、テスト フロー全体のうちの個々の部分をテストする再利用可能なスクリプトを複数作成して、ドライバ スクリプトで各部分呼び出すようにすることができます。

1. 新しいスクリプトを記録します。
たとえば、このスクリプトは、テスト アプリケーションを起動したり、ダイアログ ボックスまたはウィンドウを表示したりします。Web アプリケーションをテストしている場合は、ドライバ スクリプトはブラウザを起動したり、特定の Web ページを表示したりします。
2. 別のスクリプトを再生する箇所まで到達したら、スクリプトの記録を停止します。
3. コード ウィンドウで、カーソルを新しい行に移動し、以下のように入力します。

```
Workbench.RunScript ("script")
```

script はスクリプト名です。

適切に設計されたスクリプトは、完了前にテスト アプリケーションを元のテスト サイトに戻します。これは、ドライバ スクリプトが常に停止した箇所から再開できるようにするためです。

Oracle データベースに格納されるスクリプト名では、大文字と小文字が区別されます。Oracle データベースから挿入するスクリプトの名前は、データベースでの名前と正確に一致する必要があります。一致していないと、Silk Test Workbench でスクリプトの読み込みに失敗します。


4. テスト アプリケーション内の別のテスト サイトでスクリプトを再生する場合は、引き続き、テスト アプリケーションを適切な状態にするステップを記録します。記録を停止し、別の RunScript コマンドを挿入します。
5. 目的のテストをすべて挿入したら、テスト アプリケーションを閉じるためのステップを記録します。他のテストまたはテストスイートを実行する場合は、それに適した状態までテスト アプリケーションを戻すためのステップを記録します。

手動によるスクリプト作成

記録機能を使用するとスクリプトをすばやく効率的に作成できますが、多くの場合、そのスクリプトを再利用可能にしたり、信頼性や効果を高めたりするには、微調整や拡張が必要です。このことを行うために、任意のスクリプトにコードを手動で追加することができます。Silk Test Workbench はスクリプト言語として VB.NET を使用しているため、VB.NET に関する多少の知識が必要です。Silk Test Workbench は VB.NET スクリプト言語を拡張し、堅牢で強力なスクリプト作成機能を提供します。クラス、メソッド、およびプロパティの詳細と、それぞれのテスト スクリプトでの使用方法については、『*Silk Test Workbench* 言語リファレンス』を参照してください。

また、Silk Test Workbench にはその他の機能もあり、VB.NET のスクリプト機能を利用して手動によるスクリプト作成をより容易にしています。これらの機能には以下のものがあります。

- **オブジェクトの識別** ダイアログ ボックス : **オブジェクトの識別** ダイアログ ボックスでは、ロケータの記録方法や、オブジェクトに使用できるプロパティを表示できます。
- **オートコンプリートおよび構文に関するヒント** : 手動によるコーディングを容易にします。オートコンプリートおよび構文に関するヒントを使用すると、VB.NET 言語要素に自動的にアクセスし、表示して、自動的に入力することができます。これには、すべてのオブジェクトのキーワード、列挙、ブール値、コード構文、使用可能なメソッド、および使用可能な VB.NET プロパティが含まれます。

 **注:** 新しいテスト スクリプトを手動で作成する場合、使用するクラスの拡張をインポートするステートメントをスクリプトに追加する必要があります。たとえば、Win32 メソッドの場合、次の行をスクリプトの開始部分に挿入する必要があります :

```
Imports SilkTest.Ntf.Win32
```


アクセス可能なコントロールでのクリックを記録すると、インポートステートメントがスクリプトに自動的に挿入されます。共通 Silk Test クラス (BaseGuiTestObject など) は常に使用できるため、インポートする必要はありません。

スクリプトの記録

Silk Test Workbench を使用して、アプリケーションをテストするときに実行する操作を記録します。ある操作を実行すると、その操作とそれに対するアプリケーションの応答がコマンドとしてスクリプトに記録されます。

記録中は、記録を停止するまでテスト アプリケーションでのすべての操作 (Silk Test Workbench の操作は除く) が Silk Test Workbench によって記録されます。記録が完了したら、生成したスクリプトを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。

 **注:** デフォルトでは、Common プロジェクトにスクリプトが作成されます。別のプロジェクトにスクリプトを作成するには、**プロジェクト** リストで該当するプロジェクトを選択します。

Silk Test Workbench を Oracle データベースで使用する場合は、保存されたすべてのスクリプト名の太文字と小文字が区別されます。つまり、*SCRIPT1* という名前のスクリプトと *Script1* という名前のスクリプトは同じ Oracle データベースに共存でき、重複とはみなされません。ただし、Silk Test Workbench を SQL Server または Access で使用する場合は、保存されたスクリプトの太文字と小文字は区別されません。

3. 省略可能 : Silk Test Workbench の記録オプションを変更する場合は、**オプション** をクリックします。
4. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
5. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。

記録の開始 チェック ボックスをオンにせずに **OK** をクリックすると、Silk Test Workbench によりスクリプトが資産として保存され、スクリプト テンプレートが表示されます。

アプリケーションの選択 ダイアログ ボックスが開きます。

6. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
7. 標準アプリケーションをテストするには、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、リストからアプリケーションを選択します。
8. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
 - a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能 : あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。

たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能 : ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能 : **ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。

9. ネイティブ モバイル アプリケーション (アプリ) をテストするには :

- a) アプリをテストするモバイル デバイスをリストから選択します。
- b) **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。

このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。

Silk Test Workbench は、モバイル デバイスまたはエミュレータ上に指定したアプリをインストールします。

10OK をクリックします。Silk Test Workbench が最小化され、アプリケーションと **記録中** ウィンドウが開きます。

11テストする操作を記録します。

記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。

12サポートするブラウザー上で Web アプリケーションを記録している場合で、かつ UI で直接アクセスできないコントロールを扱いたい場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**記録中** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択します。

この機能は、次のブラウザーに対するテストで利用できます。

- Microsoft Edge
- Apple Safari
- Mozilla Firefox 41 以降
- Google Chrome 50 以降
- モバイル ブラウザー

13省略可能 : 検証ロジックをテストに追加します。

- 標準アプリケーションまたは Web アプリケーションをテストしている場合、テストに検証ロジックを追加するには、Ctrl+Alt をクリックします。Silk Test Workbench により記録が一時的に停止し、**テスト ロジック デザイナ** ウィザードが表示されます。

ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。

- Microsoft Edge 上の Web アプリケーション、またはモバイル アプリケーションをテストしている場合、検証したいオブジェクトをクリックし、**操作の選択** ダイアログ ボックスで **検証の追加** をクリックします。

検証の追加に関する詳細については、「記録中のスクリプトへの検証の追加」を参照してください。

14Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、スクリプトが表示されます。

15次のいずれか 1 つのステップを行います :

- **記録完了** ダイアログ ボックスを閉じて、スクリプトを保存するには、**再生** をクリックします。
再生 ダイアログ ボックスが開き、結果情報を指定してスクリプトを再生できます。
- **コード** ウィンドウでスクリプトを開くには、**.NET スクリプトへ移動** をクリックします。
- スクリプトを保存して、**記録完了** ダイアログ ボックスを終了するには、**保存** をクリックします。
スクリプトが **コード** ウィンドウに表示されます。

記録されたテストが **コード** ウィンドウに表示されているときには、いつでもそのテストを再生できます。

複数のテスト アプリケーションをテストするスクリプトの記録

複数のテスト アプリケーションをテストするスクリプトを記録できます。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

1. テストする主要なアプリケーション用に、スクリプトを記録するか手動で作成します。
2. **プロパティ** ペインで、**アプリケーション構成の追加** を右クリックして選択します。**アプリケーションの選択** ダイアログ ボックスが開きます。
3. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
4. 標準アプリケーションをテストするには、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、リストからアプリケーションを選択します。
5. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
 - a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
6. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
 - a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。
このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
Silk Test Workbench は、モバイル デバイスまたはエミュレータ上に指定したアプリをインストールします。
7. **OK** をクリックします。
8. 新しいアプリケーション構成を使用して、スクリプトの追加ステップを記録します。

Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するスクリプトの構成

Java Network Launching Protocol (JNLP) を使用して起動するアプリケーションでは、Silk Test Workbench に追加の構成が必要です。これらのアプリケーションは Web から起動されるため、実際のアプリケーションと、「Web Start」を起動するアプリケーションを起動するように、アプリケーション構成

を手動で構成する必要があります。このようにしない場合、アプリケーションがすでに実行されていないかぎり、ビジュアル テストを再生すると、失敗します。

1. Silk Test Workbench がアプリケーションを起動できずにテストが失敗する場合は、アプリケーション構成を編集します。
2. アプリケーションを起動するコードを含めるスクリプトを開きます。
3. プロパティ ペインで、**アプリケーション構成** カテゴリに移動します。
4. 変更するアプリケーション構成をダブルクリックします。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
5. 基本状態を編集して、再生中に Web Start が起動されるようにします。
 - a) **編集** をクリックします。
 - b) **実行可能ファイル パターン** テキスト ボックスに、javaws.exe への絶対パスを入力します。たとえば、以下のように入力します。

```
%ProgramFiles%\Java\jre6\bin\javaws.exe
```
 - c) **コマンドライン パターン** テキスト ボックスに、Web Start への URL を含むコマンドライン パターンを入力します。

```
"<url-to-jnlp-file>"
```

たとえば、SwingSet3 アプリケーションの場合、以下のように入力します。

```
"http://download.java.net/javadesktop/swingset3/SwingSet3.jnlp"
```
 - d) **OK** をクリックします。
6. **OK** をクリックします。テストは、基本状態を使用し、Web 起動アプリケーションとアプリケーション構成の実行可能パターンを開始して javaw.exe に接続し、テストを実行します。

テストを実行すると、アプリケーション構成の EXE ファイルが基本状態の EXE ファイルと一致しないという警告が表示されます。テストは予想したとおりに実行されているため、このメッセージは無視できません。

新しいブラウザ ウィンドウで開くスクリプトの構成

スクリプト用に既存のブラウザ ウィンドウを使用するのではなく、別のブラウザ ウィンドウを開く場合があります。たとえば、Silk Central Test Manager から Silk Test Workbench スクリプトを実行する場合には、スクリプトは既存の Silk Central ウィンドウを使用します。Silk Central に戻るには、ユーザーはブラウザ ウィンドウで **戻る** をクリックする必要があります。Silk Central から実行するテスト用に別のブラウザ ウィンドウを開くスクリプトを構成すると、この動作を回避できます。

1. アプリケーションを起動するコードを含めるスクリプトを開きます。
2. スクリプトで基本状態を使用する場合は、以下のステップを実行します。
 - a) プロパティ ペインで、**アプリケーション構成** カテゴリに移動します。
 - b) 変更するアプリケーション構成をダブルクリックします。**アプリケーション構成の編集** ダイアログ ボックスが開きます。
 - c) **基本状態の実行** チェック ボックスをオフにします。
3. スクリプトの先頭に Imports System.Diagnostics と入力します。
4. アプリケーションを新しいプロセスとして定義するには、以下のように入力します。

```
Dim applicationProcess As New Process()
```

`applicationProcess` は、起動するアプリケーションの名前です。

たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim ieProcess As New Process()
```

5. 以下のコードを入力して、アプリケーションのフルパスを定義します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "applicationexecutable.exe"
```

`applicationexecutable.exe` は、テストアプリケーションを起動する実行可能ファイルの名前です。たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "C:\Program Files\Internet Explorer\iexplore.exe"
```



注: テストするアプリケーションの実行可能ファイルが Silk Test Workbench と同じディレクトリにある場合は、フルパスを指定する必要はありません。

6. テストする Web アプリケーションを開くには、`ProcessStartInfo.Arguments` プロパティを URL に設定します。

たとえば、Internet Explorer を起動してサンプル Web アプリケーションをテストするには、以下の行を `Filename` 行のあとに追加します。

```
psi.Arguments = "http://demo.borland.com/InsuranceWebExtJS/index.jsf"
```

7. プロセスを起動するコードを指定します。
たとえば、`ieProcess` を起動するには、以下のように入力します。

```
ieProcess.StartInfo = psi  
ieProcess.Start()
```

8. スクリプトを保存します。

テストアプリケーションでのコントロールの識別 (.NET スクリプト)

.NET スクリプトを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。既存のコントロールを変更したり、手動で操作を挿入したりするには、テストするアプリケーションでコントロール名を識別します。

1. ツールバーの **オブジェクトの識別** をクリックします。**オブジェクトの識別** ダイアログ ボックスが開きます。
2. **識別の開始** をクリックします。
3. テストしているアプリケーションにカーソルを移動します。カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが強調表示されます。関連するロケータ文字列またはオブジェクト マップ項目は、**選択済みロケータ** テキスト ボックスに表示されます。
4. 使用するコントロールが強調表示されているときに、コントロールをクリックします。
5. 省略可能：**ロケータの詳細の表示** をクリックすると、ロケータ ツリーや関係する属性のすべてが **ロケータ属性** テーブルに表示されます。
6. 省略可能：記録したロケータ属性は、**ロケータ属性** テーブルの別のロケータ属性で置き換えることができます。

たとえば、記録したロケータは以下のように表示されます。

```
/BrowserApplication//BrowserWindow//input[@id='loginButton']
```

ロケータ属性 テーブルに `textContent Login` がリストされている場合、以下のようにしてロケータを手動で変更できます。

```
/BrowserApplication//BrowserWindow//input[@textContent='Login']
```

新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。

7. **テスト** をクリックして、ロケータがコントロールを正しく認識することを確認します。
8. ロケータをスクリプトにコピーするには、**貼り付け** をクリックします。
9. **閉じる** をクリックします。

スクリプト内からのアプリケーションの起動

通常、スクリプトを記録すると、テストするアプリケーションを起動する基本状態が自動的に記録されます。ただし、スクリプトで複数のアプリケーションをテストするか、何らかの理由で **基本状態の実行** 機能をオフにする場合は、スクリプト内からテスト アプリケーションを起動する必要があります。

1. アプリケーションを起動するコードを含めるスクリプトを開きます。
2. スクリプトの先頭に Imports System.Diagnostics と入力します。
3. アプリケーションを新しいプロセスとして定義するには、以下のように入力します。

```
Dim applicationProcess As New Process()
```

`applicationProcess` は、起動するアプリケーションの名前です。
たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim ieProcess As New Process()
```

4. 以下のコードを入力して、アプリケーションのフルパスを定義します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "applicationexecutable.exe"
```

`applicationexecutable.exe` は、テスト アプリケーションを起動する実行可能ファイルの名前です。たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "C:¥Program Files¥Internet Explorer¥iexplore.exe"
```



注: テストするアプリケーションの実行可能ファイルが Silk Test Workbench と同じディレクトリにある場合は、フルパスを指定する必要はありません。

5. 初期化時に特定の Web ページに移動するには、URL に ProcessStartInfo.Arguments プロパティを設定します。
たとえば、Internet Explorer を起動して Google.com に移動するには、Filename 行のあとに以下の行を追加します。

```
psi.Arguments = "http://www.google.com"
```

6. プロセスを起動するコードを指定します。
たとえば、ステップ 1 で定義した `ieProcess` を起動するには、以下のように入力します。

```
ieProcess.StartInfo = psi  
ieProcess.Start()
```

7. スクリプトを保存します。

Internet Explorer を起動して Google.com に移動するスクリプト全体は、以下のようになります。

```
Imports System  
Imports System.Diagnostics
```

```
Public Module Main
```

```
    Public Sub Main()  
        Dim _desktop As Desktop = Agent.Desktop  
        Dim ieProcess As New Process()  
        Dim psi As New ProcessStartInfo()  
        psi.FileName = "C:¥Program Files¥Internet Explorer¥iexplore.exe"  
        psi.Arguments = "http://www.google.com"  
        ieProcess.StartInfo = psi  
        ieProcess.Start()  
    End Sub  
End Module
```


スクリプトのオブジェクト マップ項目またはロケータの手動による記録

開始する前に、テストするアプリケーションが実行中であることを確認します。

オブジェクトの識別 ダイアログ ボックスを使用して、オブジェクト マップ項目またはロケータを手動でキャプチャします。**オブジェクトの識別** ダイアログ ボックスを使用すると、テストするオブジェクトを簡単かつ確実に識別することができます。

1. オブジェクトを含めるスクリプトを開き、オブジェクトの挿入先となるスクリプト内の場所をクリックします。
2. ツール > **オブジェクトの識別** を選択します。**オブジェクトの識別** ダイアログ ボックスが開きます。
3. **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケータを識別します。
- **ホットキー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャします。

4. **識別の開始** をクリックします。

Google Chrome でオブジェクト マップ項目またはロケータをキャプチャする場合は、Google Chrome の選択されたインスタンスが適切なオートメーション パラメータを使用して起動されているかどうかを Silk Test Workbench が検出します。適切なパラメータを使用して起動されていない場合は、Silk Test Workbench が Google Chrome を終了し、オートメーション パラメータ セットを使用して再起動します。

5. 省略可能 : ロケータを記録する前に、テスト対象アプリケーションを特定の状態にするには、**識別の停止** をクリックします。これ以降、テスト対象アプリケーションで実行する操作は、記録されなくなります。ロケータの記録を再開する場合は、**識別の開始** をクリックします。
6. 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホットキー** モードでオブジェクトをキャプチャします。

デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキスト ボックスに表示されます。

7. **ロケータの詳細** テーブルでのロケータの表示方法を絞り込むには、以下のいずれかの操作を実行します。

- **ロケータの詳細の非表示** : **ロケータの詳細** テーブルを非表示にするには、このリンクをクリックします。
- **オブジェクト マップの表示** : オブジェクト マップ項目名を **ロケータ** 列に表示するには、このチェック ボックスをオンにします。オブジェクト マップ項目名は、コントロールまたはウィンドウに対して、コントロールやウィンドウのロケータではなく論理名 (エイリアス) を関連付けます。デフォルトでは、オブジェクト マップ項目名が表示されます。ロケータを使用するには、このチェック ボックスをオフにします。
- **完全ロケータの表示** : 完全ロケータ名を表示するには、このチェック ボックスをオンにします。オブジェクトに関連付けられた属性のみを表示するには、このチェック ボックスをオフにします。
- **プロパティの表示** : **ロケータの詳細** テーブルで選択したオブジェクトの属性および属性値を表示するには、このチェック ボックスをオンにします。このテーブルの属性を選択して、ロケータの

識別に使用できます。[プロパティ] サブツリーを非表示にし、ロケータの詳細のみを表示するには、このチェックボックスをオフにします。

8. **選択済みロケータ** テキスト ボックスに表示されるオブジェクトが使用するオブジェクトであることをテストするには、**テスト** をクリックします。Silk Test Workbench により、ロケータに対応するオブジェクトが、テストするアプリケーションで強調表示されます。
9. 記録したロケータを置換するには、**ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。
10. ロケータをスクリプトにコピーするには、**貼り付け** をクリックします。Silk Test Workbench によって、ロケータがスクリプトに追加されます。
11. スクリプトで正しくコンパイルされ、実行する操作が含まれるように、ロケータを変更します。たとえば、以下の完全ロケータをスクリプトに貼り付けます。

```
/WPFWindow[@caption='Basic Controls'][1]//  
WPFRichTextBox[@automationId='editableTextBox']
```

With ステートメントに /WPFWindow 宣言がすでに含まれている場合は、以下のロケータを使用し、そのあとにメソッドが続くようにスクリプトを変更します。

```
.WPFRichTextBox("@automationId='editableTextBox']").Select(1,10)
```

既存のテストへの追加操作の記録

テストを作成したあと、テストを開き、テストの任意の場所から追加操作を記録できます。これにより、既存のテストを追加操作で更新できます。

1. 既存のテスト スクリプトを開きます。
2. 追加操作を記録するテスト スクリプトの場所を選択します。



注: 記録した操作は、選択した場所の後に挿入されます。テスト対象アプリケーション (AUT) は基本状態に戻りません。代わりに、テスト スクリプトの直前の操作が記録された範囲で AUT を開いておきます。

3. **操作 > 記録** をクリックします。
Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
4. AUT に対して実行したい追加操作を記録します。
記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
5. 記録を停止するには、**記録中** ウィンドウの **停止** をクリックします。
記録を停止するキーの組み合わせ (デフォルトでは **Alt+F10**) を押すか、タスクバーの Silk Test Workbench アイコンをクリックして、記録を停止することもできます。

既存のスクリプトを開く

既存のスクリプトを開いて、再生、確認、または変更を行います。

1. **ファイル > 開く** を選択します。**アセットブラウザ** が開きます。
2. 左側のペインで **.NET スクリプト** を選択し、スクリプトのリストを表示します。
3. 右側のペインで、開く資産をダブルクリックします。

スクリプトの編集

スクリプトを作成して保存したあとは、いつでもそれを編集できます。

1. スクリプトを開きます。
2. テキスト エディタの場合と同様にスクリプトを編集します。

テキストの選択、削除、切り取り、コピー、貼り付けなどが可能です。スクリプトウィンドウには、オートコンプリートと構文の関連付けテクノロジーも備えられており、少ないエラーでコードを完成するのに役立ちます。

たとえば、*Window* のあとにピリオドを入力すると、*Window* オブジェクトに有効なメソッドとプロパティのリストがすべて表示されます。

同様に、有効な関数のあとにスペース、カンマ、または左かっこを入力すると、構文に関するヒントが表示されます。

3. ツールバーの **保存** をクリックするか、または **ファイル > 保存** を選択して、説明を入力し、スクリプトを保存します。

スクリプトの保存

新規スクリプトまたは変更したスクリプトを保存できます。更新したスクリプトを閉じると、変更内容を保存するかどうかを確認するプロンプトが表示されます。

1. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開きます。
2. 省略可能：スクリプト名を変更し、説明を入力します。
3. **OK** をクリックします。

スクリプトでの要素の定義への移動

VB .NET スクリプトでメソッドやクラスの定義に移動するには、次のアクションを実行します。

1. スクリプトを開きます。
2. 定義に移動する要素を右クリックします。
代わりに、**F12** を押すこともできます。
3. **定義へ移動** を選択します。

アプリケーション構成

アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続する方法を定義します。Silk Test Workbench は、基本状態を作成するときに、アプリケーション構成を自動的に作成します。しかし、アプリケーション構成を追加したり、変更や削除をすることが必要になる場合があります。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

- Windows アプリケーションの場合、アプリケーション構成には以下が含まれます。

- 実行可能ファイル パターン


このパターンに一致するすべてのプロセスは、テストに対して有効化されます。たとえば、Internet Explorer の実行可能パターンは *%IEXPLORE.EXE です。実行可能ファイルの名前が IEXPLORE.EXE で、任意のディレクトリに置かれているプロセスはすべて有効化されます。

- コマンドライン パターン


コマンドライン パターンは、テストを行うために有効化されるプロセスの制約に使用される補足パターンで、コマンドライン引数の一部 (実行可能ファイル名の後ろ部分) をマッピングすることにより行います。コマンドラインパターンを含むアプリケーション構成では、実行可能パターンとコマンドラインパターンの両方に一致するプロセスのみが、テストに対して有効化されます。コマンドラインパターンが定義されていない場合は、指定された実行可能ファイルパターンを持つすべてのプロセスが有効化されます。コマンドラインの使用は、Java アプリケーションに対して特に有益です。これは、ほとんどの Java プログラムが javaw.exe を使用して実行されるためです。つまり、典型的な Java アプリケーションに対してアプリケーション構成を作成する場合、実行可能パターン

には *%javaw.exe が使用され、このパターンはすべての Java プロセスに一致します。このような場合、コマンドラインパターンを使用して、該当するアプリケーションのみがテストに対して有効化されるようにします。たとえば、アプリケーションのコマンドラインが **com.example.MyMainClass** で終わる場合には、コマンドラインパターンに ***com.example.MyMainClass** を使用します。

- ローカルマシン上のデスクトップブラウザの Web アプリケーションの場合、アプリケーション構成にはブラウザの種類だけが含まれます。

 **注:** コマンドライン引数を指定してブラウザを開始したり、ブラウザの作業ディレクトリや実行可能ファイルを指定するには、ブラウザの種類でカスタムを選択します。詳細については、「基本状態の変更」を参照してください。


- リモートマシン上の Apple Safari や Microsoft Edge の Web アプリケーションの場合、アプリケーション構成には以下が含まれます。
 - ブラウザの種類
 - 接続文字列
- モバイルブラウザの Web アプリケーションの場合、アプリケーション構成には以下が含まれます。
 - ブラウザの種類
 - 接続文字列
- ネイティブモバイルアプリケーションの場合、アプリケーション構成には以下が含まれます。
 - 接続文字列
 - シンプルなアプリケーション名。モバイルデバイス上で複数のアプリケーションが同じ名前を持つ場合は、アプリケーションの完全修飾名が使用されます。

 **注:** 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザーアプリケーション構成を複数追加しないでください。

アプリケーション構成の編集

アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続する方法を定義します。Silk Test Workbench は、基本状態を作成するときに、アプリケーション構成を自動的に作成します。しかし、アプリケーション構成を追加したり、変更や削除をすることが必要になる場合があります。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

- Silk Test Workbench メニューで、**ツール > アプリケーション構成の編集** をクリックします。プロジェクトが選択されていない場合、**プロジェクトの選択** ダイアログボックスが表示されます。
- リストからプロジェクトを選択して、**編集** をクリックします。**アプリケーション構成の編集** ダイアログボックスが開き、既存のアプリケーション構成が一覧表示されます。
- アプリケーション構成をさらに追加するには、**アプリケーション構成の追加** をクリックします。
 - アプリケーションの選択** ダイアログボックスで、テストするアプリケーションの種類に対応するタブを選択します。
 - テストするアプリケーションを選択します。
 - OK** をクリックします。

 **注:** 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザーアプリケーション構成を複数追加しないでください。

- アプリケーション構成を削除するには、**削除** をクリックします。
- アプリケーション構成を編集するには、**編集** をクリックします。
- OK** をクリックします。


アプリケーション構成の変更

アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続し、構成する方法を定義します。

1. ビジュアル テストまたは VB .NET スクリプトを開きます。
2. ビジュアル テストの場合は、タスク ペインで <<開始>> ステップをクリックします。
3. プロパティ ペインで、**アプリケーション構成** カテゴリに移動します。
4. 変更するアプリケーション構成を選択します。
 - ビジュアル テストの場合は、変更するアプリケーション構成をクリックし、**アプリケーション構成の編集** をクリックします。
 - VB .NET スクリプトの場合は、変更するアプリケーション構成をダブルクリックします。


アプリケーション構成の編集 ダイアログ ボックスが開きます。

5. 実行可能ファイルを指定するには、**実行可能ファイル** フィールドに実行可能ファイルへの完全パスを入力します。

 **注:** Web アプリケーションをテストする場合に、ブラウザの実行可能ファイルを指定する場合は、ブラウザの種類からカスタムを選択します。

たとえば、Mozilla Firefox を起動するには、「C:\Program Files (x86)\Mozilla Firefox\firefox.exe」を入力します。

6. デスクトップ アプリケーションをテストし、実行可能ファイルと一緒にコマンド ライン パターンを使用したい場合には、コマンド ライン パターンを **コマンド ライン パターン** テキスト ボックスに入力します。
7. Web アプリケーションをテストする場合は、**移動する URL** テキスト ボックスに、Web アプリケーションのアドレスを入力します。
8. 省略可能 : あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
9. 省略可能 : ブラウザー ウィンドウの **向き** を選択します。
10. 省略可能 : **ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
11. モバイル ネイティブ アプリケーションのテストを行う場合は、**モバイル アプリ** テキスト ボックスに、テストするモバイル アプリケーションの名前を入力します。
12. **OK** をクリックします。

 **注:** 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザ アプリケーション構成を複数追加しないでください。

アプリケーション構成の削除

ビジュアル テストまたは VB .NET スクリプトでテストしなくなったアプリケーションに対応するアプリケーション構成をビジュアル テストまたは VB .NET スクリプトから削除します。

1. ビジュアル テストまたは VB .NET スクリプトを開きます。
2. ビジュアル テストの場合は、**タスク** ペインで <<開始>> ステップをクリックします。
3. プロパティ ペインで、**アプリケーション構成** カテゴリに移動します。
4. 削除するアプリケーション構成を選択します。
 - ビジュアル テストの場合、アプリケーション構成をクリックします。

- VB .NET スクリプトの場合、アプリケーション構成を右クリックします。

5. **アプリケーション構成の削除** をクリックします。

6. **はい** をクリックして、アプリケーション構成を削除することを確認します。

[アプリケーションの選択] ダイアログ ボックス

アプリケーションの選択 ダイアログ ボックスを使用して、テストしたアプリケーションを選択し、アプリケーションとオブジェクト マップを関連付けたり、アプリケーション構成をテストに追加したりします。アプリケーションの種類は、ダイアログ ボックスのタブとしてリストされます。使用したいアプリケーションの種類に対応したタブを選択します。

Windows システムで実行中のすべての Microsoft Windows アプリケーションの一覧が表示されます。リストから項目を選択して、**OK** をクリックします。

キャプションを持たないプロセスを表示しない チェック ボックスを使用して、キャプションを持たないアプリケーションを一覧から除去します。

Web

利用可能なすべてのブラウザーの一覧が表示されます (任意の接続済みモバイル デバイス上のモバイル ブラウザーや Mac 上の Apple Safari を含む)。**移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザーのインスタンスが既に実行されている場合、**実行中のブラウザーの URL を使用する** をクリックして、実行中のブラウザー インスタンスに現在表示されている URL の記録を行うことができます。あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。ブラウザー サイズを選択する場合は、選択したブラウザーの **向き** も選択できます。



注: 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザー アプリケーション構成を複数追加しないでください。

モバイル

利用可能なすべてのモバイル デバイスとすべての実行中の Android エミュレーターの一覧が表示されます (リモート ロケーションに接続されたデバイスを含む)。モバイル ネイティブ アプリケーションをテストする場合は、このタブを選択します。選択したモバイル デバイス上で現在実行中のモバイル アプリケーション (アプリ) をテストするために選択したり、テストするアプリの名前やファイルを手動で参照したり指定することができます。

リモート ロケーションの編集

リモート ロケーション ダイアログ ボックスを使用すると、リモート ロケーション上のブラウザーやモバイル デバイスを、テストするアプリケーションのセットに追加できます。

1. **ツール > リモート ロケーションの編集** をクリックします。**リモート ロケーション** ダイアログ ボックスが表示されます。

2. リモート ロケーションを追加するには、次の操作を実行します。

a) **ロケーションの追加** の右側にある矢印をクリックして、Silk Test Information Service を使用してリモート ロケーションを追加するのか、Silk Central を追加するのかを指定します。



注: Silk Central は 1 つだけリモート ロケーションとして設定できます。Silk Central との統合が既に設定されている場合は、リモート ロケーション リストに Silk Central が表示されません。

b) **ロケーションの追加** をクリックします。**ロケーションの追加** ダイアログ ボックスが表示されます。

c) Silk Test Workbench がリモート マシン上の Information Service に接続するためのリモート ロケーションの URL とポートを、**ホスト** フィールドに入力します。

デフォルトのポートは 22901 です。

d) 省略可能 : **名前** フィールドにリモート ロケーションの名前を入力します。

3. 既存のリモート ロケーションを編集するには、**編集** をクリックします。

4. リモート ロケーションを削除するには、**削除** をクリックします。
5. 省略可能：**アプリケーションの選択** ダイアログに表示されるブラウザやデバイスの数を減らすには、**このロケーション上のデバイスとブラウザを非表示** をクリックします。そのリモート ロケーションにインストールされたブラウザや接続されたデバイスが、**アプリケーションの選択** ダイアログに表示されなくなります。デフォルトでは、すべてのリモート ロケーションにインストールされたブラウザや接続されたデバイスのすべてが、**アプリケーションの選択** ダイアログに表示されます。
6. **OK** をクリックします。

リモート ロケーションを追加すると、リモート ロケーション上にインストールされているブラウザ (Mac 上の Apple Safari など) が、**アプリケーションの選択** ダイアログ ボックスの **Web** タブで利用可能になり、リモート ロケーションに接続されているモバイル デバイスが、**アプリケーションの選択** ダイアログ ボックスの **モバイル** タブで利用可能になります。

ユーザー アカウント制御

ユーザー アカウント制御 (UAC) を設定して Silk Test Workbench を使用できます。アプリケーションのセキュリティレベルが Silk Test Workbench と同じである場合、UAC が有効な状態でアプリケーションをテストできます。

システムに Silk Test Workbench をインストールするには管理者権限が必要です。Silk Test Workbench がインストールされている場合は、Silk Test Workbench を使用するための管理者権限は必要ありません。



注: UAC が有効な状態で、Silk Test Workbench を使用する場合は、Silk Test Workbench にインストール ディレクトリへの書き込み権限が必要となるため、Silk Test Workbench の Program Files ディレクトリへのインストールは推奨しません。Silk Test Workbench を Program Files ディレクトリへインストールした場合は、Silk Test Workbench のデータベースを構成するために管理者権限が必要です。

特定のテクノロジ ドメインの無効化

あるテストにおいて、特定のテクノロジ ドメインが不要で、問題を起こすことがわかっている場合 (テストのパフォーマンスを低下させる場合など)、そのテクノロジ ドメインをロードしないように無効化できます。



注: そのテクノロジ ドメインがテストに必要なことが確実な場合にのみ、テクノロジ ドメインを無効化してください。

1. ビジュアル テストの **テクノロジ ドメイン** ダイアログ ボックスを開くには：
 - a) ビジュアル テストの **<<開始>>** ステップを選択します。
 - b) **プロパティ** タブを開きます。
 - c) **テクノロジ ドメイン** フィールドをクリックします。
 - d) **開く** をクリックします。
2. VB .NET スクリプトの **テクノロジ ドメイン** ダイアログ ボックスを開くには：
 - a) **プロパティ** タブを開きます。
 - b) プロパティ ツリーで、テクノロジ ドメインを無効化するアプリケーション構成を右クリックします。
 - c) **テクノロジ ドメインの選択** をクリックします。
3. **テクノロジ ドメイン** ダイアログ ボックスで、**テクノロジ ドメインを自動的にロードする** チェック ボックスをオフにします。
4. **テクノロジ ドメイン** リストから、そのテストでロードする必要の無いテクノロジ ドメインをオフにします。
5. **OK** をクリックします。

テストの以降の実行では、有効なテクノロジ ドメインだけがロードされます。

アプリケーション構成エラー

アプリケーションにアタッチできない場合、以下のエラーメッセージが表示されます。アプリケーション <アプリケーション名> にアタッチするのに失敗しました。詳細については、ヘルプを参照してください。

この場合、以下の表に示されている 1 つ以上の問題が原因である可能性があります。

問題	原因	解決策
タイムアウト	<ul style="list-style-type: none">システムが遅すぎます。システムのメモリ サイズが小さすぎます。	速いシステムを使用するか、現在使用しているシステムのメモリ使用量を減らします。
ユーザー アカウント制御 (UAC) の失敗	システムの管理者権限がありません。	管理者権限を持つユーザー アカウントでログインします。
コマンド ライン パターン	コマンド ライン パターンが固有すぎます。この問題は特に Java の場合に発生します。再生が意図したとおりに機能しないことがあります。	パターンから不明瞭なコマンドを削除します。
<ul style="list-style-type: none">ブラウザーの選択 ダイアログ ボックスが、Web アプリケーションに対してテストを実行しているときに表示されないWeb アプリケーションに対してテストを実行しているときに、複数のブラウザー インスタンスが開始される開いているブラウザー インスタンスを使用して Web アプリケーションに対してテストを実行しているときに、Silk Test Workbench が動作を停止することがある	基本状態と複数のブラウザー アプリケーション構成がテストケースに対して定義されています。	1 つを除くすべてのブラウザー アプリケーション構成をテストケースから削除します。
テスト実行時の再生エラー	テストに対してアプリケーション構成が定義されていません。 次の例外が表示される可能性があります： アプリケーション構成が存在しません。	<ul style="list-style-type: none">キーワード駆動テストを実行するときに、基本状態を実行するキーワードがテストに含まれていることを確認します。アプリケーション構成が現在のプロジェクトに対して設定されていることを確認します。

アプリケーション構成のトラブルシューティング

アプリケーションの選択 ダイアログ ボックスにアプリケーションが表示されない理由

- キャプションを持たないプロセスを表示しない チェック ボックスをオフにします。このチェック ボックスは、デフォルトではオンになっており、キャプションを持たないアプリケーションはダイアログ ボックスに表示されません。
- Silk Test Workbench をシステム特権で実行します。

1. Silk Test Workbench を閉じます。

2. Open Agent を停止します。
 3. Silk Test Workbench を管理者として実行します。
- **タスク マネージャー** を使用して、アプリケーションが他のユーザー アカウントで実行されていないか確認します。
 - アプリケーションが runas コマンド、または同等のコマンドで起動されていないか確認します。

アプリケーション構成での環境変数

以下の環境変数は、アプリケーション構成で使用できます。

環境変数	説明
TMP または TEMP	Windows 一時ディレクトリ。
USERPROFILE	ユーザー プロファイル ディレクトリ (C:\%Documents and Settings\%myUserName)。
APPDATA	アプリケーション データ ディレクトリ。
ProgramFiles	Program Files ディレクトリ。
ProgramFiles(x86)	32 ビット アプリケーションの Program Files ディレクトリ (64 ビット環境でのみ使用可能)。

基本状態

アプリケーションの基本状態とは、各テストの実行開始前にアプリケーションに想定される既知の安定した状態です。大抵の場合この状態は、アプリケーションを最初に起動したときの状態になります。基本状態は、記録するアプリケーションを選択すると自動的に生成されます。

ビジュアル テストまたはスクリプトを記録すると、Silk Test Workbench により基本状態が自動的に作成されます。テストの必要性に応じて基本状態を有効にしたり無効にしたりすることができます。たとえば、アプリケーション ウィンドウを手動で開く場合があります。このような場合には、基本状態を無効にします。基本状態の情報は、アプリケーション構成の一部です。

基本状態はテストの整合性を保障するための重要な一因です。各テストが安定した基本状態から開始できることを保障することによって、あるテストのエラーによって、後続のテストが失敗しないことを保障することができます。

Silk Test Workbench は、ビジュアル テストまたはスクリプトを実行する前に、アプリケーションがその基本状態にあることを保障します。

エラーが発生すると、Silk Test Workbench はテストの実行を停止します。

基本状態のオン/オフの切り替え

ビジュアル テストまたはスクリプトを記録すると、Silk Test Workbench により基本状態が自動的に作成されます。テストの必要性に応じて基本状態を有効にしたり無効にしたりすることができます。たとえば、アプリケーション ウィンドウを手動で開く場合があります。このような場合には、基本状態を無効にします。

1. ビジュアル テストで、以下のステップを実行します。
 - a) **テスト ステップ** ペインで、<<開始>> ステップをクリックします。
 - b) **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
 - c) アプリケーション構成フィールドをクリックし、**アプリケーション構成の編集** をクリックします。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
 - d) 次のいずれか 1 つのステップを行います：

- 基本状態を使用するには、**基本状態の実行** チェック ボックスをオンにします。
 - 基本状態をオフにするには、**基本状態の実行** チェック ボックスをオフにします。
- e) **OK** をクリックします。
2. スクリプトで、以下のステップを実行します。
- a) **プロパティ** ペインで、変更するアプリケーション構成を右クリックし、**アプリケーションの実行可能ファイル名の編集** を選択します。
たとえば、Internet Explorer で Web アプリケーションをテストする場合は、メニューから **ieexplore.exe の編集** を選択します。
アプリケーション構成の編集 ダイアログ ボックスが表示されます。
- b) 次のいずれか 1 つのステップを行います：
- 基本状態を使用するには、**基本状態の実行** チェック ボックスをオンにします。
 - 基本状態をオフにするには、**基本状態の実行** チェック ボックスをオフにします。
- c) **OK** をクリックします。

基本状態の実行

アプリケーションに対してテストの記録を開始する前に基本状態を実行することで、記録するすべてのアプリケーションを起動して、記録に適した状態にすることができます。

基本状態を実行すると、アクティブなビジュアル テストまたは .NET スクリプトのすべてのアプリケーション構成のうち、**基本状態の実行** が設定されているものが実行されます。

基本状態を実行するには：


操作 > 基本状態の実行 をクリックします。

アプリケーションの種類に応じて、Silk Test Workbench は次のアクションを実行します。

- 現在のプロジェクトで定義されているアプリケーション構成に対応するすべてのアプリケーションのアプリケーション構成を実行します。
- デスクトップ アプリケーションの場合、Silk Test Workbench は指定したインストール ディレクトリからアプリケーションを開きます。%ProgramFiles% 環境変数を含むディレクトリへのパスが指定された場合、Silk Test Workbench は Program Files ディレクトリからアプリケーションを検索し、見つからなかった場合に、Silk Test Workbench は Program Files (x86) ディレクトリを検索します。%ProgramFiles(x86)% 環境変数を含むディレクトリへのパスが指定された場合、Silk Test Workbench は Program Files (x86) ディレクトリからアプリケーションを検索し、見つからなかった場合に、Silk Test Workbench は Program Files ディレクトリを検索します。
- Web アプリケーションの場合、Silk Test Workbench は指定したブラウザで指定した URL の Web アプリケーションを開きます。
- モバイル Web アプリケーションの場合、Silk Test Workbench は指定したモバイル デバイスまたはエミュレータ上の指定したブラウザで指定した URL を開きます。
- モバイル ネイティブ アプリケーションの場合、Silk Test Workbench は指定したモバイル デバイスまたはエミュレータ上で指定したアプリを開きます。指定したアプリが指定したモバイル デバイスまたはエミュレータ上にインストールされていない場合、Silk Test Workbench はアプリをインストールしてから開きます。

ユーザー インターフェイスからの基本状態の変更

Silk Test Workbench が記録や再生時にテスト対象アプリケーションを開始する方法を指定する基本状態を、ユーザー インターフェイスから編集できます。基本状態では、AUT の実行可能ファイルの場所、作業ディレクトリや、Web アプリケーションの URL や接続文字列などを指定できます。たとえば、ステージング Web サイトで既に実行したテストを、プロダクション Web サイトで実行する場合には、基本状態の URL を変更すれば、新しい Web サイトに対してテストを実行することができます。


 **注:** 再生時に特定のテストに対してのみ Silk Test Workbench がテスト対象アプリケーション (AUT) を開始するように指定する場合は、テストを含むスクリプトで基本状態を編集します。詳細については、「[スクリプトでの基本状態の変更](#)」を参照してください。

ユーザー インターフェイスから基本状態を編集するには :

1. ビジュアル テストまたは VB .NET スクリプトを開きます。
2. ビジュアル テストの場合は、タスク ペインで **<<開始>>** ステップをクリックします。
3. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
4. 変更するアプリケーション構成を選択します。
 - ビジュアル テストの場合は、変更するアプリケーション構成をクリックし、**アプリケーション構成の編集** をクリックします。
 - VB .NET スクリプトの場合は、変更するアプリケーション構成をダブルクリックします。


アプリケーション構成の編集 ダイアログ ボックスが開きます。

5. **基本状態の実行** チェック ボックスをオンにし、アプリケーション構成で基本状態を使用します。
6. 実行可能ファイルを指定するには、**実行可能ファイル** フィールドに実行可能ファイルへの完全パスを入力します。

 **注:** Web アプリケーションをテストする場合に、ブラウザの実行可能ファイルを指定する場合は、ブラウザの種類からカスタムを選択します。

たとえば、Mozilla Firefox を起動するには、「C:\Program Files (x86)\Mozilla Firefox\firefox.exe」を入力します。

7. コマンド ライン引数を指定するには、**コマンド ライン引数** フィールドに引数を入力します。

 **注:** Web アプリケーションをテストする場合に、コマンド ライン引数を指定してブラウザを開始する場合は、ブラウザの種類からカスタムを選択します。

たとえば、Mozilla Firefox を *myProfile* プロファイルを指定して起動するには、「-p myProfile」を入力します。

8. テストするアプリケーションがディレクトリに依存する場合は、**作業ディレクトリ** フィールドにディレクトリへのパスを指定します。

たとえば、Java アプリケーションを起動するバッチ ファイルを使用する場合には、バッチ ファイルは JAR ファイルを相対パスで参照することができます。この場合、正しく相対パスが処理されるように作業ディレクトリを指定します。
9. デスクトップ アプリケーションをテストする場合は、**ロケーター** フィールドにアプリケーションのメイン ウィンドウを指定します。

たとえば、ロケーターは、/Shell[@caption='Swt Test Application'] のようになります。
- 10 実行可能ファイル パターンを指定して、デスクトップ アプリケーションをテストする場合は、**実行可能ファイル パターン** テキスト ボックスに、テストするデスクトップ アプリケーションの実行可能ファイルの名前とファイルへのパスを入力します。

たとえば、電卓を指定する場合には、*%calc.exe と入力します。

- 11 デスクトップ アプリケーションをテストし、実行可能ファイルと一緒にコマンド ライン パターンを使用したい場合には、コマンド ライン パターンを **コマンド ライン パターン** テキスト ボックスに入力します。
- 12 Web アプリケーションをテストする場合は、**移動する URL** テキスト ボックスに、Web アプリケーションのアドレスを入力します。
- 13 省略可能 : あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。

たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
- 14 省略可能 : ブラウザー ウィンドウの **向き** を選択します。
- 15 省略可能 : **ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。

16 リモート ロケーション上の Web アプリケーションまたはモバイル ネイティブ アプリケーション (Mac に接続されているモバイル デバイスなど) をテストする場合に、リモート ロケーションを変更する場合は、**変更** をクリックして **アプリケーションの選択** ダイアログ ボックスを開き、**リモート ロケーションの編集** をクリックします。

17 モバイル アプリケーションや Apple Safari 上の Web アプリケーションをテストする場合は、**接続文字列** テキスト ボックスに接続文字列を入力します。

詳細については、「[接続文字列](#)」を参照してください。

18 WebDriver ベースのブラウザのケイパビリティを編集する場合は、**接続文字列** テキスト ボックスを使用できます。

たとえば、Google Chrome のブラウザ ウィンドウを最大化して開始する場合には、**接続文字列** テキスト ボックスに次のように入力します。

```
"chromeOptions="
+ "{"
+ " ¥"args¥":[ ¥"--start-maximized¥"]"
+ "};"
```

詳細については、「[WebDriver ベースのブラウザのケイパビリティの設定](#)」を参照してください。

19 モバイル ネイティブ アプリケーションのテストを行う場合は、**モバイル アプリ** テキスト ボックスに、テストするモバイル アプリケーションの名前を入力します。


20 **タイムアウト** を変更すると、基本状態が実行されたときにアプリケーションの準備ができ実行されるまで Silk Test Workbench が待機する時間 (ミリ秒) を指定できます。アプリケーションの準備が整っておらず、実行中でない場合に、タイムアウト時間が経過すると、プロセスはタイムアウトします。

21 **OK** をクリックします。

22 テスト対象アプリケーションの起動に時間がかかる場合は、再生オプションのアプリケーション準備完了タイムアウトの値を増やしてください。


アプリケーションが実行していない場合は、基本状態を実行することで開始されます。アプリケーションがすでに実行中の場合には、Silk Test Workbench によってアプリケーションの別のインスタンスは開始されません。

複数のアプリケーション構成がテストに含まれており、基本状態に関連付けられているオブジェクト以外のアプリケーションまたは Web ページを変更する場合は、基本状態をオフにすることができます。このことは、変更を記録したり再生したりするときに基本状態が使用されないことを意味します。このため、アプリケーションまたは Web ページを起動するステップをテスト内に記録する必要があります。たとえば、Web ページをテストする場合は、テスト内で Internet Explorer を起動します。

 **注:** 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザ アプリケーション構成を複数追加しないでください。

スクリプトでの基本状態の変更

Silk Test Workbench が再生時にテスト対象アプリケーションを開始する方法を指定する基本状態を、スクリプトで編集できます。基本状態では、AUT の実行可能ファイルの場所、作業ディレクトリ、Web アプリケーションの URL や接続文字列などを指定できます。たとえば、ステージング Web サイトで既に実行したテストを、プロダクション Web サイトで実行する場合には、基本状態の URL を変更すれば、新しい Web サイトに対してテストを実行することができます。

 **注:** Silk Test Workbench が記録や再生時にテスト対象アプリケーション (AUT) を開始する方法を指定するには、ユーザー インターフェイスから基本状態を編集します。詳細については、「[基本状態の変更](#)」を参照してください。

スクリプトで基本状態を編集するには：

1. スクリプトを開きます。
2. baseState メソッドを変更します。

基本状態の作成と実行との間にコードを追加できます。

```
' VB code
' Go to web page 'demo.borland.com/InsuranceWebExtJS'
BrowserBaseState baseState = new BrowserBaseState()
' <-- Insert your changes here!
baseState.Execute()
```

3. テストするアプリケーションの実行可能ファイルの名前とファイルへのパスを指定する場合は、次のコードを使用します。

```
' VB code
baseState.Executable=executable
```

たとえば、電卓を指定する場合は、次のように入力します。

```
' VB code
baseState.Executable = "C:\Windows\SysWOW64\calc.exe"
```

Mozilla Firefox を指定する場合は、次のように入力します。

```
' VB code
baseState.Executable = "C:\Program Files (x86)\Mozilla Firefox\firefox.exe"
```

4. コマンドライン引数を指定する場合は、次のコードを使用します。

```
' VB code
baseState.CommandLineArguments = commandLineArguments
```

たとえば、Mozilla Firefox を *myProfile* プロファイルを指定して起動する場合は、次のように入力します。

```
' VB code
baseState.CommandLineArguments = "-p myProfile"
```

5. 他の作業ディレクトリを指定する場合は、次のコードを使用します。

```
' VB code
baseState.WorkingDirectory = workingDirectory
```

6. 実行可能ファイルパターンを使用する場合には、次のコードを使用します。

```
' VB code
baseState.ExecutablePattern = executablePattern
```

たとえば、電卓に対して実行可能ファイルパターンを指定する場合は、次のように入力します。

```
' VB code
baseState.ExecutablePattern = "*\calc.exe"
```

7. 実行可能ファイルと一緒にコマンドラインパターンを使用する場合は、次のコードを使用します。

```
' VB code
baseState.CommandLinePattern = commandLinePattern
```

たとえば、アプリケーションのコマンドラインが **com.example.MyMainClass** で終わる場合には、コマンドラインパターンに ***com.example.MyMainClass** を使用します。

```
' VB code
baseState.CommandLinePattern = "*com.example.MyMainClass"
```

8. Web アプリケーションまたはモバイル Web アプリケーションをテストする場合、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、インストール済みのブラウザまたはモバイルブラウザから 1 つを指定します。

たとえば、Google Chrome を指定する場合は、次のように入力します。

```
' VB code
baseState.BrowserType = BrowserType.GoogleChrome
```

9. Web アプリケーションまたはモバイル Web アプリケーションをテストする場合、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、テストする Web アプリケーションのアドレスを指定します。

```
' VB code
baseState.Url = url
```

たとえば、次のように入力します。

```
' VB code
baseState.Url = "demo.borland.com/InsuranceWebExtJS/"
```

- 10 デスクトップ ブラウザー上の Web アプリケーションをテストする場合は、ブラウザー ウィンドウの幅と高さを指定できます。

```
' VB code
baseState.ViewportHeight = viewportHeight
baseState.ViewportWidth = viewportWidth
```

- 11 リモート ロケーション上の Web アプリケーション、またはモバイル ネイティブ アプリケーションをテストする場合は、接続文字列を指定します。

```
' VB code
New MobileBaseState(connectionString)
```

接続文字列についての詳細は、「[リモート デスクトップ ブラウザーの接続文字列](#)」または「[モバイル デバイスの接続文字列](#)」を参照してください。

- 12 Mozilla Firefox または Google Chrome のケイパビリティを編集する場合にも、接続文字列を使用できます。

たとえば、Mozilla Firefox のダウンロード フォルダーを設定する場合は、次のように入力します。

```
' VB code
baseState.ConnectionString = "moz:firefoxOptions= {¥"prefs¥": { ¥"browser.download.dir¥":¥"C:/Download/¥"}}};"
```

詳細については、「[WebDriver ベースのブラウザーのケイパビリティの設定](#)」を参照してください。

記録の概要

新しいビジュアル テストまたはスクリプトを記録すると、**アプリケーションの選択** ダイアログ ボックスが開きます。テストするアプリケーションをダブルクリックします。Silk Test Workbench により、テストアプリケーションを識別するアプリケーション構成と基本状態が自動的に作成されます。アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続し、構成する方法を定義します。アプリケーションの基本状態とは、各テストの実行開始前にアプリケーションに想定される既知の安定した状態です。

ビジュアル テストまたはスクリプトを記録する前に **オブジェクトの識別** ダイアログ ボックスを開いた場合にも、**アプリケーションの選択** ダイアログ ボックスが開きます。

テストするアプリケーションを選択したら、以下のステップのいずれかを実行します。

- ビジュアル テストを記録する。
- スクリプトを記録する。
- 識別するオブジェクトを選択する。

必要に応じて、アプリケーション構成を変更し、基本状態をオフにすることができます。

記録中のオブジェクトのハイライト

記録中に AUT のアクティブ オブジェクトは緑色の矩形でハイライトされます。新しいオブジェクトがアクティブになるとすぐに、この新しいオブジェクトがハイライトされます。同じオブジェクトがアクティブなまま 0.5 秒が経過すると、ツールチップが表示されます。ツールチップには、アクティブ オブジェクトのクラス名とアクティブ オブジェクトに相対的なマウスの現在位置が表示されます。このツールチッ

ブは、新しいオブジェクトがアクティブになったり、ユーザーがマウスを押したり、表示されてから 2 秒が経過すると、表示されなくなります。

記録中および再生中に除外される文字

記録および再生中に Silk Test が無視する文字を以下に示します。

文字	コントロール
...	MenuItem
タブ	MenuItem
&	すべてのコントロール。アンパサンド (&) はアクセラレータとして使用されるため、記録されません。

記録中に利用可能なアクション

記録中に次のアクションを **記録中** ウィンドウで実行できます。

アクション	ステップ
記録の一時停止	一時停止 をクリックして、操作を記録せずに AUT を特定の状態にしてから、 記録 をクリックして記録を再開できます。
記録した操作の順番の変更	記録中 ウィンドウで記録した操作の順番を変更するには、移動したい操作を選択して新しい場所にそれらをドラッグします。
複数の操作の選択	複数の操作を選択するには、Ctrl を押しながら操作をクリックするか、もしくは選択する最初の操作をクリックしてから、 Shift を押しながら最後のアクションをクリックします。
記録した操作の再生	記録中 ウィンドウから記録した操作を再生するには、操作を選択してから 再生 をクリックします。すべての記録した操作を再生するには、 記録した操作 をクリックしてから 再生 をクリックします。
記録した操作の削除	誤って記録した操作を 記録中 ウィンドウから削除するには、マウス カーソルをその操作上にポイントすると表示される 削除 をクリックします。
イメージまたはコントロールのプロパティの検証	検証するオブジェクトの上にマウス カーソルを移動して、 Ctrl+Alt を押します。
オブジェクト マップ エントリの変更	Web アプリケーション、またはモバイル Web アプリを記録しているときに、記録したオブジェクトに対して自動的に生成されたオブジェクト マップ エントリが分かりにくい場合や、特殊な文字が含まれてしまっている場合、分かりやすいオブジェクト マップ エントリに変更したいことがあります。このような場合は、記録中にオブジェクトを右クリックして、 操作の選択 ダイアログの オブジェクトの識別 領域を展開します。そして、 オブジェクト マップ ID フィールドでオブジェクト マップ エントリを編集できます。たとえば、デモアプリケーションの画像に対して自動的に生成されるオブジェクト マップ エントリは、「http demo borland」になります。このエントリが何を参照しているのかを理解するのは、オブジェクト マップを見ただけでは困難です。このオブジェクト マップ エントリを「InsuranceWebHomePageBanner」のように変更すれば、より分かりやすくなります。この機能は、Internet Explorer 以外のすべてのサポートするデスクトップおよびモバイル ブラウザで利用できます。
別のロケータの選択	Web アプリケーション、またはモバイル Web アプリに対して記録しているときに、記録したオブジェクトに対して自動的に生成されたロケータが要求を満たしていない場合、 ロケータ フィールドにある矢印をクリックして、Silk Test Workbench に他のロケータを提示させることができます。提示される

アクション	ステップ
	<p>ロケータはすべて、オブジェクトを一意に識別できるものです。この機能は、Internet Explorer 以外のすべてのサポートするデスクトップおよびモバイルブラウザで利用できます。</p>

テストの拡張

ビジュアルテストまたはスクリプトを拡張するさまざまな方法について説明します。

アクティブ データ


ビジュアルテストとスクリプトには、通常、定数またはリテラルデータの値が含まれていて、アプリケーションのオープンフィールドや複数選択のコントロールに自動的にデータが入力されます。このようなアプリケーションをテストする場合、ビジュアルテストまたはスクリプトでは、各トランザクションに対し同じデータフィールド名およびデータ値が使用されます。

リテラルデータ以外のデータを使用してビジュアルテストやスクリプトを繰り返し再生するには、別のデータ値が使用されるようにビジュアルテストやスクリプトを変更する必要があります。

アクティブデータテストでは、強力で包括的なアプリケーションテストソリューションの入力として既存のデータを利用できます。アクティブデータを使用して、トランザクションごとに異なるデータセットを使用することにより、複雑なコードを記述したり、既存のデータを損なうことなく、テストアプリケーションで複数のトランザクションを実行できます。

既存のデータファイルを手動で編集することなく、既存のデータファイルに新しいデータを書き込むこともできます。

アクティブデータを使用すると、テスト担当者は、有効なデータや無効なデータをテストアプリケーションに入力して、その結果を検証できます。つまり、一連のテスト条件を実行するためのビジュアルテストやスクリプトの数が少なくて済み、メンテナンスのためのオーバーヘッドが減ります。

 **注:** アクティブデータファイルを開くときにエラーが発生した場合は、エラー処理を使用してテスト実行を続行できます。

ビジュアルテスト

ビジュアルテストでは、ローカル変数を使用する場所にアクティブデータを使用できます。入力をアクティブデータファイルのデータに置き換えたり、一連の入力ステップを繰り返す繰り返しロジックを作成し、記録されたデータをアクティブデータファイルのデータに置き換えたりできます。式の内容など、ビジュアルテストで作成されたデータはすべて、アクティブデータファイルに書き込んでテストに利用することもできます。

ビジュアルテストで使用するアクティブデータファイルを選択するには、ビジュアルテストの <<開始>> ステップのアクティブデータプロパティで設定するか、または任意のステップを選択して **挿入 > アクティブデータ** を選択します。

スクリプト

スクリプトでは、ローカル変数を使用する場所にアクティブデータを挿入できます。ただし、ビジュアルテストとは異なり、ウィザードは使用できません。手動でデータを挿入する必要があります。

キーワード駆動テスト

キーワード駆動テストでは、テスト全体に対してアクティブデータを簡単に使用することはできません。しかし、個々のキーワードを実装しているビジュアルテストや.NETスクリプト内では、アクティブデータを使用することができます。このようにキーワード内でアクティブデータを使用するには、アクティブデータを使用したキーワードの種類の実装について説明するトピックを参照してください。

アクティブ データ テストの利点

アプリケーション テストでは、テスト対象アプリケーション (AUT) のコントロールにデータを入力して、そのデータを処理し、最終的に結果のレポートを得ることで、AUT が期待通り機能していることを確認します。テストの記録時に入力したデータを変更せずに、テストを再生すると、ある特定のデータ セットに対して AUT が機能することだけを検証することになります。

サポートするシナリオに対して個々にテストを記録するのではなく、エッジケースを含む様々なシナリオに対して AUT をテストできるようなテストを作成できるようにするために、Silk Test Workbench にはアクティブ データテスト機能が備わっており、次のメリットを提供します。

- テストの実行時に様々なデータ セットを使って、単一のテストで様々なサポートするシナリオやエッジケースをテストする
- 実際のテストを変更することなく、テストデータをアクティブ データ ファイルで直接変更したり、Silk Test Workbench 内から変更する


アクティブ データ資産

アクティブ データは、Excel スプレッドシートのようなファイルに格納され、これをアクティブ データ ファイルと呼びます。アクティブ データ ファイルは、Silk Test Workbench データベース内部に格納することも、ファイル システムなどの外部に格納することも可能です。

アクティブ データ資産を使うと、アクティブ データ ファイル内のデータに Silk Test Workbench からアクセスできます。また、アクティブ データ資産を使って、Silk Test Workbench から直接アクティブ データ ファイル内のデータを管理することもできます。アクティブ データ資産は、ビジュアル テスト、VB .NET スクリプト、キーワード駆動テストのすべてから使用することができ、いつでも編集できます。

アクティブ データ資産を作成する場合、資産で使用する外部ファイルを選択します。次の項目を指定できます。

- アクティブ データ資産内で使用するアクティブ データ ファイルのデータ
- アクティブ データ ファイルの最初の行が列ヘッダーかどうか
- アクティブ データ ファイルを書き込み可能にするか、読み取り専用にするか


 **注:** Silk Test Workbench が開いているアクティブ データ ファイルにアクセスしようとすると、実行時エラーが発生します。このような場合にテスト実行を続行するには、エラー処理を使用します。

アクティブ データ ファイルは事前に作成しておくことも、Silk Test Workbench 内で作成して編集することも可能です。アクティブ データ資産は、**アセット ブラウザ** を使用して作成、変更、および管理します。

アクティブ データ テスト資産の作成

新しいアクティブ データ資産を作成するには、以下を実行します。

1. **表示 > アセット ブラウザ** を選択します。
2. **資産の種類** で、**アクティブ データ** を右クリックして、**アクティブ データの新規作成** を選択します。**アクティブ データ** ウィンドウが開きます。
3. **名前** テキスト ボックスに、資産のわかりやすい名前を入力します。
アクティブ データ資産名は、Silk Test Workbench 資産の命名規則に従う必要があります。

 **ヒント:** アクティブ データ資産に、その資産に使用されるアクティブ データ ファイルと同じ名前を付けると、テストで使用する資産を特定しやすくなります。

4. 資産が使用するデータを含んだ新しいアクティブ データ ファイルを作成するか、既存のアクティブ データ ファイルを選択できます。
 - 新しいアクティブ データ ファイルをファイル システムに作成するには、**外部ファイル** を選択してから **新規作成** をクリックします。

- 既存のアクティブ データ ファイルのデータを使用するには、**外部ファイル** を選択してから **参照** をクリックし、ファイルを選択します。アクセスできるようにするため、ファイルを閉じてください。
- 新しいアクティブ データ資産を Silk Test Workbench データベースに作成するには、**内部データ** を選択してから **新規作成** をクリックします。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースに格納して、アクティブ データ資産を使って内部ファイルからデータを使用するには、**内部データ** を選択してから **アップロード** をクリックし、ファイルを選択します。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースからファイル システムにコピーするには、**内部データ** を選択してから **ダウンロード** をクリックし、ファイルを選択します。保存したファイルは、Excel などのアプリケーションを使って編集できます。

ファイル テキスト ボックスには、アクティブ データ ファイルの名前が表示されます。

5. 省略可能 : **説明** テキスト ボックスに、資産の説明を入力します。

説明を使用して、データ ファイルとして使用されるファイルの種類、資産を作成するアクティブ データ テストによって実行されるテストの種類、またはアクティブ データ テストでテストされるアプリケーションを示すことができます。**アセットブラウザ** では、説明が入力されているすべてのアクティブ データ資産に説明が表示されます。



注: 空のデータ ファイルが必要な場合は、**保存** および **閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** 設定ウィンドウを閉じます。

6. **アクティブ データ** ウィンドウで、**オプション** タブに切り替えます。

7. アクティブ データ ファイルに対して読み取りオプションを設定します。

8. **アクティブ データ** ウィンドウで、**詳細** タブに切り替えます。

9. アクティブ データ ファイル内のデータを編集します。

10 **保存** または **データの保存** をクリックしてすべての変更内容を保存するか、または **保存して閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** ウィンドウを閉じます。

アクティブ データ資産のデータ ファイルが外部の場合、つまりファイル システムに格納されており、Silk Test Workbench データベースではない場合、**データの保存** をクリックすると、データ ファイルだけが保存されます。この場合、アクティブ データ資産は保存されません。

新しいアクティブ データ資産が **アセットブラウザ** に表示されます。

アクティブ データ テストのためのテスト ファイルの準備

Silk Test Workbench では、アクティブ データ テストで以下の種類のファイルを使用できます。

- Microsoft Excel® スプレッドシート (.xls) ファイル



注: Microsoft Excel® がインストールされていない場合、読み取り専用モードで .xls ファイルを使用できます。

- Microsoft Excel® スプレッドシート (.xlsx) ファイル



注: Silk Test Workbench で .xlsx ファイルを使用するには、Microsoft Excel® 2007 以降をインストールしてください。Microsoft Excel® 2007 以降がインストールされていない場合、**2007 Office System** ドライバ : データ接続コンポーネント をインストールして、読み取り専用モードで .xlsx ファイルを使用できます。

- プレーン テキスト ファイル (.txt)
- カンマ区切り値ファイル (.csv)

カンマ区切り値ファイルおよびプレーン テキスト ファイルには、1 セットのデータしか含めることができませんが、Excel スプレッドシート ファイルには複数のシートを含めることができます。各シートには独自の表が含まれ、アクティブ データ テストでは別のエンティティとして扱われます。

カンマ区切り値ファイルまたはプレーン テキスト ファイルを使用して新しいテスト データ資産を作成する場合は、ファイルだけを選択すれば済みます。Excel スプレッドシート ファイルを使用して新しいテスト データ資産を作成する場合は、使用するシートも選択する必要があります。

アクティブ データ テストのためのファイル準備のヒント

- 可能な場合、データ ファイルのデータは、アプリケーション内と同じデータ カテゴリにグループ化する必要があります。たとえば、データ ファイルにアクティブ データとして使用される姓と名が含まれている場合、テスト対象アプリケーションには姓と名の別個のフィールドがなければなりません。アプリケーションで姓と名に別のフィールドが使用され、データ ファイルでは姓と名を組み合わせで 1 つのデータ フィールドになっている場合、テスト データと、アプリケーションのテストに必要なデータが 1 対 1 の関係になりません。アクティブ データ ファイル内の結合されたデータをテスト アプリケーションの姓フィールドまたは名フィールドのテストに使用した場合は、期待した結果が得られない場合があります。
- アクティブ データ テストに使用するファイルは、ビジュアル テストまたはスクリプトで使用する前に作成しておく必要があります。
- オブジェクト名テスト データの場合、データ ファイル内の名前エントリにはかっこや引用符を含めないでください。

たとえば、あるフォーム上の複数のエディット ボックスで同じテストを実行し、それらの各エディット ボックスは Index プロパティのみを使用して一意に識別される場合、各エディット ボックスの名前は以下のように表されます。

```
TextField("[1]")
TextField("[2]")
TextField("[3]")
TextField("[4]")
```

アクティブ データ ファイル内のオブジェクト名のテスト データの各行には、フォーム上のエディット ボックスを表すそれぞれのインデックスを含めることができます。ただし、アクティブ データ ファイルの各行のデータでは、以下のようにかっこまたは引用符を除いてください。

```
Index=1
Index=2
Index=3
Index=4
```

- アクティブ データ ファイルの最初の行は、データの最初の行にすることも列見出しにすることもできます。アクティブ データ テストを設定するとき、アクティブ データ資産では、アクティブ データ ファイルの最初の行のタイプを適切に設定する必要があります。
- ユーザーがアクティブ データ資産を読み取り専用として開くと、読み取り権限だけが付与されます。アクティブ データ資産が読み書き可能モードで開かれると、資産に一度も書き込みが行われなくても、読み取りと書き込みの権限が付与されます。これは、テストがあとでアクティブ データ ファイルに書き込みをするように変更された場合でも、正しく再生されるようにするためです。読み取りおよび書き込み権限で開かれたアクティブ データ資産は、排他的ファイル ロック機能を持ち、資産と関連付けられたアクティブ データ ファイルが他のプログラムから開かれられないようにすることができます。

アクティブ データ テスト用の新しいデータ ファイルの作成

アクティブ データ資産には、一緒に使用するデータ ファイルを関連付ける必要があります。Silk Test Workbench を使用してファイルを作成するか、またはテキスト エディタや Microsoft Excel® などのツールを使用してファイルを作成します。

アクティブ データ資産の作成時に、Silk Test Workbench で、資産に関連付けるアクティブ データを含む新しいアクティブ データ ファイルを作成することもできます。

1. **表示 > アセット ブラウザ** を選択します。
2. **資産の種類** で、**アクティブ データ** を右クリックして、**アクティブ データの新規作成** を選択します。**アクティブ データ** ウィンドウが開きます。
3. **名前** テキスト ボックスに、資産のわかりやすい名前を入力します。

アクティブ データ資産名は、Silk Test Workbench 資産の命名規則に従う必要があります。



ヒント: アクティブ データ資産に、その資産に使用されるアクティブ データ ファイルと同じ名前を付けると、テストで使用する資産を特定しやすくなります。

4. **新規作成** をクリックして、アクティブ データ ファイルとして使用する新しいファイルを作成します。**アクティブ データの新規作成** ダイアログ ボックスが開きます。
5. **名前** テキスト ボックスに、新しいファイルのわかりやすい名前を入力します。
このファイルに含まれるデータを示す名前か、このファイルを使用するテストの対象アプリケーションの種類を示す名前を付けることをお勧めします。
6. 新しいアクティブ データ ファイルの種類を選択します。
ファイルの種類の詳細については、「アクティブ データ テスト資産の作成」を参照してください。
7. **参照** をクリックし、**フォルダの参照** ダイアログ ボックスを使用して、新しいアクティブ データ ファイルを保存する場所を選択します。**OK** をクリックして、**アクティブ データの新規作成** ダイアログ ボックスに戻ります。新しいアクティブ データ ファイルの場所が、**場所** テキスト ボックスに表示されます。
8. **OK** をクリックして、新しいアクティブ データ ファイルを作成します。
9. アクティブ データ資産の作成を続行します。


アクティブ データ テスト ファイルに対する読み取りオプションの設定

アクティブ データ 設定ウィンドウの **オプション** タブを使用して、アクティブ データ ファイルに対する読み取り/書き込みオプションを設定します。また、このタブでは、アクティブ データ ファイルの最初の行を列見出しとして扱うかどうかを設定したり、アクティブ データ テストに使用するスプレッドシート データ ファイル中のシートを設定したりもします。これらのオプションの設定手順は、スプレッドシート データ ファイルに対するものとテキスト (.txt) またはカンマ区切り (.csv) データ ファイルに対するものとで多少異なります。

1. Silk Test Workbench によってファイルの最初のデータ行が列見出しとして扱われるようにするには、**アクティブ データ** ウィンドウの **オプション** タブで **最初の行を見出しとして使用** チェック ボックスをオンにします。
このチェック ボックスをオンにすると、アクティブ データ テストの作成時に、データ ファイル内の各列の行に含まれるレコードに対してデータをマップできます。**最初の行を見出しとして使用** チェック ボックスがオフの場合、ファイルの最初の行のデータは実際のアクティブ データ項目として解釈されます。
2. ファイルが実際に読み取り専用設定されているかどうかにかかわらず、Silk Test Workbench がアクティブ データ ファイルを読み取り専用ファイルとして扱うよう強制するには、**読み取り専用を適用** チェック ボックスをオンにします。
このオプションを選択しても、アクティブ データ ファイルの読み取り/書き込み設定は変更されません。
読み取り専用を適用 チェック ボックスをオンにすると、**アクティブ データ** ウィンドウの **詳細** タブの行と列の **挿入** および **削除** ボタンも無効になります。そのため、Silk Test Workbench 内からアクティブ データ ファイルを変更できなくなります。
3. スプレッドシートでは、**シート** リストで、アクティブ データ資産で使用するデータを含む Excel スプレッドシート内のシートを選択します。
4. カンマ区切り値ファイルおよびテキスト ファイルでは、アクティブ データ ファイル内でデータ項目を区切るために使用される文字を指定するオプションを選択します。
カスタム文字を使用する場合は、**その他** オプションを選択して、**その他** テキスト ボックスにカスタム文字を入力します。
5. アクティブ データ資産の作成を続行します。

アクティブ データ テストのためのアクティブ データ ファイルの編集


アクティブ データ ウィンドウの **詳細** タブを使用して、アクティブ データ資産で使用されるアクティブ データ ファイル内のデータを編集します。 **全般** タブでアクティブ データ ファイルの名前が指定されるまでは、**詳細** タブの機能は無効になっています。 **読み取り専用を適用** チェック ボックスがオフの場合は、列と行をデータ ファイルに追加したり、データ ファイルから削除したりできます。また、データをリストからコピーしたり、リストに貼り付けたりすることもできます。


 **ヒント:** ファイルの元のアプリケーションでアクティブ データ ファイルを編集することもできます。元のアプリケーションを使用してアクティブ データ ファイルを変更した場合、その変更内容は Silk Test Workbench で次回そのファイルを開いたときに表示されます。

1. アクティブ データ ファイルの内容を必要に応じて更新します。
 - a) 該当するボタンをクリックして行または列を挿入します。行と列は、最後にある行または列のあとに自動的に挿入されます。
 - b) 行または列内のセルを 1 つ選択し、該当する **削除** ボタンをクリックして行または列を削除します。次に、メッセージ ボックスで **はい** をクリックして削除を確認します。その行または列内のすべてのセルの、すべてのデータも削除されます。
 - c) データを入力または編集するセル内をダブルクリックします。データは入力することも、外部ソースからコピーしてセルへ貼り付けることもできます。
 - d) 列のセルを選択し、**名前の変更** をクリックして、その列の見出しの名前を変更します。
 - e) 任意の時点で **データの保存** をクリックして、すべての更新内容を保存します。 **データの保存** をクリックすると、変更内容が元のアクティブ データ ファイルに保存されます。特定のテスト実行に対してのみデータを変更する場合は、最初に元のアクティブ データ ファイルのコピーを作成します。

2. アクティブ データ資産の作成を続行します。

再生中に、アクティブ データ テストに使用されるアクティブ データ ファイルのデータを更新することができます。 ビジュアル テストの場合は、更新内容が元のアクティブ データ ファイルに自動的に保存されますが、データの更新を元のアクティブ データ ファイルに保存する方法を制御するステップを作成することもできます。詳細については、「**詳細設定オプションの変更**」を参照してください。 スクリプトの場合は、再生中に .NET コマンドを使用してアクティブ データ ファイルを編集することができます。

 **ヒント:** **詳細** タブを使用してアクティブ データ ファイルのデータを更新して保存すると、Silk Test Workbench によって対応するアクティブ データ資産のバージョン番号が更新されます。ただし、アクティブ データ ファイル内のデータを Microsoft Excel などの外部ツールを使用して更新すると、資産のバージョン番号は更新されません。

 **ヒント:** データは、アクティブ データ テストに必要なとされる引用符または後続スペースを含めて追加する必要があります。

アクティブ データに使用するデータの指定

アクティブ データ テストにアクティブ データ ファイルのデータを使用する場合、使用するファイル内のデータ行および使用順序を指定することができます。 **開始行**、**終了行** および **ランダム数** の値によって、使用される行および行の使用順序が決定されます。

ランダム数 に指定する値により、資産に関連付けられているアクティブ データ ファイルのレコードをアクティブ データ テストで使用するかどうか、またどのように使用するかが決まります。

ビジュアル テストでは、**開始行**、**終了行**、および **ランダム数** の値は、**テスト ロジック (繰り返し) デザイン** の **使用するアクティブ データ資産の定義** ページで設定します。これらの値は、設定後、「アクティブ データを使用して繰り返します」ステップの **開始行**、**終了行**、および **ランダム数** プロパティで更新できます。VB .NET テスト スクリプトの場合、**開始行**、**終了行**、および **ランダム数** の値を LoadActiveData メソッドの StartRow、EndRow、および RandomCount パラメータを使って設定および更新します。

以下の表は、さまざまな **開始行** と **終了行** の値を許容される **ランダム数** の値と共に使用した結果を示しています。

[開始行]の値	[終了行]の値	[ランダム数]の値 (ビジュアルテスト)	[ランダム数]の値 (VB .NET スクリプト)	結果
1	-1 ^b	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	すべての行を、インデックスの順序で読み込みます。
1	5	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	行を 1、2、3、4、5 の順で読み込みます。
5	-1 ^b	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	アクティブ データ ファイルの行 5 から最後の行までを順番に読み込みます。
5	1	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	行を 5、4、3、2、1 の順で読み込みます。
-1 ^a	1	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	すべての行を、逆の順序で読み込みます。
-1 ^a	5	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	アクティブ データ ファイルの最後の行から行 5 までを逆の順番で読み込みます。
-1 ^a	-1 ^b	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	アクティブ データ ファイルの最後の行を読み込みます。
1	1	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	行 1 を読み込みます。
1	-1 ^b	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	すべての行をランダムな順序で読み込みます。
1	5	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	行 1 から 5 をランダムな順序で読み込みます。
5	-1 ^b	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	アクティブ データ ファイルの行 5 から最後の行までをランダムな順序で読み込みます。
5	1	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	行 5 から 1 をランダムな順序で読み込みます。
-1 ^a	1	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	すべての行をランダムな順序で読み込みます。
-1 ^a	5	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	アクティブ データ ファイルの行 5 から最後の行までをランダムな順番で読み込みます。
1	-1 ^b	5	5	アクティブ データ ファイルから 5 行をランダムに読み込みます。
1	10	5	5	アクティブ データ ファイルの 1 番目の行から 10 番目の行までのうち、5 行をランダムに読み込みます。
10	-1 ^b	5	5	アクティブ データ ファイルの 10 番目の行から最後の行までのうち、5 行をランダムに読み込みます。
1	5	10	10	範囲外実行時エラーが発生します。

^a ビジュアルテストでは、**開始行**の値に対して**データを含む最後の行から開始** チェックボックスをオンにすることに相当します。

^b ビジュアルテストでは、**終了行**の値に対して**データを含む最後の行で終了** チェックボックスをオンにすることに相当します。


アクティブデータファイルの行数を超える値を **StartRow**、**EndRow**、または **RandomCount** パラメータに指定すると、範囲外実行時エラーが生成されます。


LoadActiveData() が、別のアプリケーションで開かれているアクティブデータファイルを読み込もうとすると、実行時エラーが発生します。

ビジュアルテストでのアクティブデータの使用

ビジュアルテストでアクティブデータを使用するには、使用するすべてのデータファイルを作成する必要があります。また、使用する各データファイルをアクティブデータ資産に関連付ける必要があります。

Silk Test Workbench データベースの外部からのテストデータの格納と取得は、外部ファイルを使用して行われます。テストデータは、Excel スプレッドシートファイル (.xls、.xlsx)、テキストファイル (.txt)、またはカンマ区切り値ファイル (.csv) に格納できます。テストデータを外部ファイルに格納すると、複数のアクティブデータ資産、および任意のビジュアルテストまたはスクリプトで、テストデータを再利用できるようになります。


 **注:** Microsoft Excel® がインストールされていない場合、すべての .xls ファイルは読み取り専用モードになります。Silk Test Workbench で .xlsx ファイルを使用するには、Microsoft Excel® 2007 以降をインストールしてください。Microsoft Excel® 2007 以降がインストールされていない場合、**2007 Office System** ドライバ：データ接続コンポーネントをインストールして、読み取り専用モードで .xlsx ファイルを使用できます。

 **注:** アクティブデータファイルを開くときにエラーが発生した場合は、エラー処理を使用してテスト実行を続行できます。

ビジュアルテストでは、どのアクティブデータ資産がビジュアルテストに使用できるかを指定する必要があります。アクティブデータを使用するビジュアルテストの各ステップのプロパティは、正しいデータを指すように変更する必要があります。これはマッピングと呼ばれます。

アクティブデータ資産のビジュアルテストへの関連付け


アクティブデータ資産をビジュアルテストで使用するには、アクティブデータ資産をビジュアルテストに関連付ける必要があります。1つのビジュアルテストにアクティブデータ資産をいくつでも関連付けることができます。

 **注:** アクティブデータ資産は、ビジュアルテストの任意のテストステップが選択されているときに挿入/作成できます。ただし、プロパティは、<<開始>>ステップの**アクティブデータ**プロパティに表示されます。


1. アクティブデータを使用するビジュアルテストを開きます。
2. 新しいアクティブデータ資産を作成するには、**挿入 > アクティブデータ > 新規作成** を選択します。
3. アクティブデータ資産を参照するには、**挿入 > アクティブデータ > 既存の関連付け** を選択します。

アクティブデータの参照 ダイアログボックスが開きます。

- a) 資産リストにすべての資産を表示するか、自分で作成したものだけを表示するかを選択します。
- b) リストから資産を選択します。

 **注:** 列見出し (**名前**、**プロジェクト**、**説明**、**更新日**) をクリックして、その内容を基準にアクティブデータ資産のリストをアルファベット順に並べ替えることができます。

- c) **OK** をクリックします。

 **ヒント:** 関連付けられたアクティブデータファイルにデータ行またはデータ列が含まれていない場合、Silk Test Workbench によりメッセージが表示されます。アクティブデータ資産を

ビジュアルテストに関連付けるには **はい** をクリックし、取り消すには **いいえ** をクリックします。

4. 以上のステップを繰り返して、他のアクティブ データ資産のアクティブ データ ファイルを関連付けます。

アクティブ データ資産がビジュアルテストに関連付けられるたびに、項目 [n] という名前の追加プロパティがアクティブ データ プロパティに表示されます。ここで、[n] はビジュアルテストに関連付けられたアクティブ データ資産の合計数のうちの連続番号を示します。選択したアクティブ データ資産の名前は、このプロパティの値として表示されます。**アクティブ データ** プロパティは、**全般** プロパティ カテゴリ内にあります。

5. <<開始>> ステップをクリックして、ビジュアルテストにおけるアクティブ データの関連付けを編集します。

ステップのプロパティが **プロパティ** ペインに表示されます。カテゴリ別またはアルファベット順にプロパティを並び替えます。関連付けを編集するには、**プロパティ** ペインの適切な **項目** プロパティをクリックします。

- **アクティブ データの参照** ダイアログ ボックスを使用して、関連付けられているアクティブ データ資産を置き換えるには、値領域の **アクティブ データの参照** ツールバー ボタンをクリックします。
- ビジュアルテストへのアクティブ データ資産の関連付けを削除するには、**削除** ツールバー ボタンをクリックします。

ビジュアルテストでのアクティブ データに対する繰り返しロジックの作成

アクティブ データに対して繰り返しロジックを作成すると、ビジュアルテストでアクティブ データ ファイルのデータを検索し、ステップ内のリテラル データをアクティブ データ ファイルのデータと置き換えることができます。

ビジュアルテストでアクティブ データの繰り返しロジックを作成する場合、基本的にはアクティブ データテストに使用されるテスト ステップの連続ブロックを定義します。リテラル データをアクティブ データ ファイルのデータと置き換えるよう定義されたブロックのステップは、再生時に繰り返されます。ステップを繰り返す回数は繰り返しロジックで定義され、アクティブ データ ファイルのすべての行または一部の行を対象にすることができます。

ビジュアルテストでアクティブ データの繰り返しロジックを作成しても、ビジュアルテストでアクティブ データが使用されるとは限りません。この処理部分によってテストにステップが設定されますが、ステップ内のデータそのものは、アクティブ データ ファイルのデータにマップする必要があります。

1. ビジュアルテストで、置き換えるデータを含む最初のステップの前にあるステップを選択します。
2. **挿入 > テスト ロジック > 繰り返し** をクリックします。 **テスト ロジック デザイナ** ウィザードが開きます。
3. ウィザードの [ようこそ] 画面を表示する **全般** オプションが **はい** に設定されている場合は、**ようこそ** ページが開きます。 **次へ** をクリックして、**ロジックの種類を選択 (繰り返し)** ページを表示します。



ヒント: ウィザードの [ようこそ] 画面を表示する **全般** オプションが **いいえ** に設定されている場合は、**テスト ロジック デザイナ** ウィザードの最初のページとして **ロジックの種類を選択 (繰り返し)** ページが表示されます。

4. **ロジックの種類を選択 (繰り返し)** ページで、**アクティブ データ ファイル** を使用して **ステップ シーケンスを繰り返す** オプションを選択して、**次へ** をクリックします。 **使用するアクティブ データ資産の定義** ページが表示されます。
5. 使用するデータ ファイルを含むアクティブ データ資産を選択します。
6. 省略可能： **シート名** リストからデータ ファイル中の使用するシートを選択します。デフォルトでは、アクティブ データ資産で指定されているシートが使用されます。
7. **開始行** と **終了行** を選択して、定義したシートのどの行が繰り返し中に使用されるかを定義します。詳細については、「アクティブ データに使用するデータの指定」を参照してください。

- すべての指定したデータ行を順番に繰り返すか、ランダムな順序で繰り返すか、あるいは指定した数だけランダムにデータ行を繰り返すかを定義するには、**ランダム数** から対応するオプションを選択します。

詳細については、「アクティブ データに使用するデータの指定」を参照してください。

- 次へ** をクリックします。



ヒント: 繰り返しの構築 ページを使用して、ビジュアル テストで繰り返すステップのブロックを定義します。**繰り返しの構築** ページを使用してビジュアル テストで繰り返すステップのブロックを定義することは、**アクティブ データ領域の定義** ウィザード ページを使用してテスト スクリプトで繰り返すコードのブロックを定義することと似ています。

繰り返しの構築 ページが開きます。

- アクティブ データ繰り返しロジックに対して繰り返すステップを定義したら、**次へ** をクリックします。**要約** ページが開きます。

- 完了** をクリックします。Silk Test Workbench によって、**テスト ロジック デザイナ** ウィザードの **繰り返しの構築** ページで選択された最初のステップの前、および最後のステップの後ろに、繰り返しロジック ステップが挿入されます。繰り返しロジックを開始する、挿入されたステップのステップ テキストは次のようになります: アクティブ データ '[資産名]' を使用して繰り返します

繰り返しロジックを終了する、挿入されたステップのステップ テキストは次のようになります: 繰り返しの終了

アクティブ データ ファイルのデータのビジュアル テスト データへのマップ

ビジュアル テストでアクティブ データ ファイル内のデータを使用するには、該当するテスト ステップのデータをアクティブ データ ファイル内のデータ列にマップする必要があります。これには、まず目的のアクティブ データ ファイルを含むアクティブ データ資産をビジュアル テストに関連付けます。次に、アクティブ データ ファイルのデータを使用する各ステップを、該当するデータを含むアクティブ データ ファイルの列にマップする必要があります。この手順を使用して、テスト ステップ内のデータをアクティブ データ ファイルのデータにマップします。



注: マップされたデータに対してアクティブ データ ファイルのデータを使用した繰り返しロジックが作成されていない場合は、データをアクティブ データ ファイルのデータ列にマップしても、再生中に列の最初のレコードしか使用されません。

- ビジュアル テストで、ビジュアル テストに関連付けられたアクティブ データ ファイルのデータと置き換えるデータを含む最初のステップを選択します。

ステップのプロパティが **プロパティ** ペインに表示されます。プロパティは、カテゴリ別またはアルファベット順に並び替えることができます。アクティブ データ ファイルのデータを使用した繰り返しロジックがビジュアル テストにすでに作成されている場合は、マップされるデータを含むステップが「アクティブ データを使用して繰り返します」ステップと「繰り返しの終了」ステップの間にネストされている必要があります。

- ステップの **プロパティ** ペインで、置き換えるデータを含むプロパティの値領域をクリックします。
- 値領域の **選択** ボタンをクリックして、**アクティブ データ** をクリックします。**アクティブ データ列の選択** ダイアログ ボックスが開きます。**アクティブ データ資産** テキスト ボックスに、ビジュアル テストと関連付けられたアクティブ データ資産が表示されます。アルファベット順の最初の資産の名前がボックスに表示されます。
- アクティブ データ資産** テキスト ボックスで、マップするデータを含むアクティブ データ資産の名前を選択します。

選択したアクティブ データ資産のアクティブ データ ファイルの列名が、**列** リストに表示されます。

- 選択したアクティブ データ資産のアクティブ データ ファイルが現在元のアプリケーション (テキスト エディタまたは Microsoft Excel) で開かれていると、列名が **列** リストに表示されません。この場合は、元のアプリケーションを閉じ、マップするデータを含むアクティブ データ資産を再度選択します。

- アクティブ データ資産に対して **最初の行を見出しとして使用** チェック ボックスがオンになっている場合は、データ ファイルの最初の行のデータが列見出しとみなされます。各列の最初の行のデータが、**列** リストに表示される列名となります。
- アクティブ データ資産に対して **最初の行を見出しとして使用** チェック ボックスがオフになっている場合は、データ ファイルの最初の行のデータがデータとみなされます。この場合、Silk Test Workbench は **列** リスト内の名前に対して英字を順番に使用します (「A」、「B」、「C」)。



ヒント: アクティブ データ ファイルを作成する場合は、可能なかぎり、名前付きの見出しを使用します。**アクティブ データ列の選択** ダイアログ ボックスの **列** リストにわかりやすい列名が表示されると、アクティブ データ ファイルの正しい列にデータをマップしやすくなります。

5. **列** リストで、選択したプロパティで使用される実際のデータと置き換えるデータを含む列名を選択します。
6. **OK** をクリックして **アクティブ データ列の選択** ダイアログ ボックスを閉じ、データをマップします。Silk Test Workbench は、アクティブ データ ファイルの選択されたデータにマップする式を使用して実際のデータを置き換えます。式の構文は以下のとおりです: [[アクティブ データ資産名.データ型]("列名")]。ここで、[アクティブ データ資産名.データ型] は、選択されたアクティブ データ資産の名前および選択された列内のデータのデータ型です。ビジュアル テストの場合、アクティブ データ ファイルで使用されるデータは Silk Test Workbench により自動的に解釈されます。データに数字のみが含まれる場合、データ型は Long になります。データに 1 文字以上の英字が含まれる場合、データ型はテキストになります。

("列名") は、プロパティのデータがマップされる列の名前です。



ヒント: リテラル、変数、式、またはアクティブ データを選択して設定した値は、元の値が存在するときにビジュアル テストを保存していた場合にはリセットすることができます。プロパティの値をリセットするには、値領域で **選択** ボタンをクリックして、元のリテラル、変数、式の結果、またはアクティブ データのいずれかのタイプを割り当てます。データ型の元の値がプロパティ値として表示されます。

7. ビジュアル テストに関連付けられたアクティブ データ ファイルのデータと置き換えるデータを含む他のすべてのテスト ステップについて、以上のステップを繰り返します。
マップされたデータに対してアクティブ データ ファイルのデータを使用した繰り返しロジックが作成されていない場合は、データをアクティブ データ ファイルのデータ列にマップしても、再生中に列の最初のレコードしか使用されません。



注: 同じビジュアル テストで複数のアクティブ データ資産を使用することができますが、1 つの繰り返し文内で使用できるアクティブ データ資産は 1 つだけです。

アクティブ データによるビジュアル テストのステップ テキストの更新

ステップ内のデータをアクティブ データ ファイルのデータにマップすると、Silk Test Workbench によってステップ テキストが更新されます。ステップでは、**テスト ステップ** ペインの **ステップ** 列のリテラル データが、アクティブ データ資産、置き換えられるデータ型、および置き換えられるデータを含むアクティブ データ ファイルの列を示すテキストと置き換えられます。

以下の例に、Silk Test でステップ テキストのリテラル データがアクティブ データ ファイル内のマップされたデータとどのように置き換えられるかを示します。

例 1 : テキスト入力

以下の例は、テスト アプリケーションの TextField コントロールにリテラル値 Smith を入力する自動ステップを示しています。リテラル値は、ステップの **テキスト** プロパティに設定されています。

'Smith' を入力します

リテラル データが **contacts** という名前のアクティブ データ資産にある **Last Name** という名前の列のデータで置き換えられると、ステップ テキストは以下のように変わります。

'[[contacts].Text("Last Name")]' を入力します

この場合、置き換えられたデータには英字が含まれるため、Silk Test Workbench は置き換えられた値をテキストとして解釈し、ステップで指定されるデータ型が Text になります。

例 2 : 数値入力

以下の例は、ステップの Left および Top プロパティの座標値に基づいて画面上の特定の位置にウィンドウを移動するステップを示しています。

(235, 427) に移動します

Left および Top プロパティの値が **coordinates** という名前のアクティブ データ資産に関連付けられたファイル内の **Left Coordinate** および **Top Coordinate** という名前の列のデータに置き換えられると、ステップ テキストは以下のように変わります。

([[>coordinates].Long("Left Coordinate")], [[coordinates].Long("Top Coordinate")]) に移動します

この場合、置き換えられたデータには数値のみが含まれるため、Silk Test Workbench はアクティブ データ ファイルの値を数値として解釈し、ステップで指定されるデータ型が Long になります。

ビジュアルテストでのアクティブ データ ファイルのデータの更新

ビジュアルテストに関連付けられているアクティブ データ ファイルのデータは、ビジュアルテスト内で更新することができます。ビジュアルテストでアクティブ データを更新するには、式を使用します。**式デザイナー**を使用すると、アクティブ データ ファイルの列のデータを入力として使用でき、データの値を変更する式を作成できます。更新されたデータは、同じアクティブ データ ファイルの同じ列に保存できます。

アクティブ データ テストの繰り返しロジック内で使用すると、アクティブ データ ファイルの列にある各行のデータを式を介して更新し、列内の元の行に格納することができます。これには、式内で列を使用し、式でデータを更新（列内の各行のデータを更新）してから、式の内容をアクティブ データ ファイルの同じ列に戻します。また、関連付けられている他のビジュアルテストの別の列に式の内容を格納したり、更新されたデータをローカル変数に格納したりすることもできます。ローカル変数は、テスト ロジックや結果コメントで使用でき、グローバル変数または他のビジュアルテストやスクリプトに渡すことができます。

式を使用してビジュアルテストで更新されたアクティブ データは、ビジュアルテストの再生が正常に完了した時点で自動保存されます。ただし、ビジュアルテストを変更して、更新されたデータの即時保存、再生中の任意の時点での保存、自動保存の回避を行うこともできます。

ビジュアルテストでの式を使用したアクティブ データの更新

ビジュアルテストに関連付けられているアクティブ データ ファイルのデータは、ビジュアルテスト内で更新することができます。これを行うには、**式デザイナー**を使用して、データの指定、更新を行います。更新されたデータを変数に保存してその変数をビジュアルテストで使用したり、元のアクティブ データ ファイルに保存したりできます。

1. ビジュアルテストで、作成する式で使用する任意のローカル変数を作成します。
2. ビジュアルテストで、作成する式で使用するデータ列を含むアクティブ データ資産を関連付けます。
3. 式ステップを作成する場所の直前のステップを選択します。



ヒント: アクティブ データ ファイルのデータを使用する式は、アクティブ データを使用する繰り返しロジックの外側を含め、ビジュアルテスト内の任意の場所に配置できます。ただし、式で更新されたデータをアクティブ データ列の元の行にそれぞれ保存する場合は、繰り返しロジック内でデータを更新するステップの直後に配置する必要があります。Expression ステップを繰り返しロジックのあとに配置すると、アクティブ データ ファイルで最後にアクセスされた行のデータのみが更新されます。

4. **挿入 > 式** を選択します。**式デザイナー** ダイアログ ボックスが開きます。
5. **式デザイナー** ダイアログ ボックスを使用して、アクティブ データ ファイルの列のデータを更新する式を作成します。

列内のアクティブ データを更新し、内容を同じ列に戻すには、**式デザイナー** ダイアログ ボックスの **式エディタ** を使用して値を更新してから、式の内容を同じ列に戻します。

たとえば、**式エディタ** でデータ ファイルの「Gender」という列（列は string データ型で、値は「M」または「Fem」）を入力として使用する場合は、**式エディタ** で、データに「ale」を追加して「Male」または「Female」にします。その後、データはダイアログ ボックスの **出力** セクションに示された同じ列に戻されます。

6. 式を作成してテストしたあと、**式デザイナー** ダイアログ ボックスで **OK** をクリックします。Silk Test Workbench によって、選択したステップのあとに 式：ステップが挿入されます。



ヒント: 作成したあとで式を編集するには、式：ステップを選択し、ステップの 式プロパティを選択して、式デザイナー をクリックし、**式デザイナー** ダイアログ ボックスを使用して式を編集します。

式で更新されたアクティブ データは、再生が正常に完了した時点で元のアクティブ データ ファイルに自動保存されます。ただし、元のアクティブ データ ファイルに更新内容を即時保存するステップを作成したり、保存を取り消して更新されたデータの保存を回避するステップを作成したりすることもできます。

ビジュアル テスト内で更新されたアクティブ データの即時保存

ビジュアル テストの再生中に式内で更新されるアクティブ データは、再生が正常に完了した時点で元のアクティブ データ ファイルに自動保存されます。アクティブ データ ファイルのさまざまなセルのデータが変更されている場合、他のセルが指定されていないかぎり、Silk Test Workbench は更新されたデータを同じセルに保存します。

再生が正常に完了するまでデータが元のアクティブ データ ファイルに自動保存されるのを待つ代わりに、更新された特定のデータを直ちに保存するステップを作成することができます。このステップでは、ビジュアル テストでこのステップを実行するまでに更新された選択済みのアクティブ データのみが保存されます。ステップの実行後に更新されたアクティブ データに関しては、自動保存を取り消さないかぎり、再生が正常に完了した時点で保存されます。

1. ビジュアル テストで、作成する式で使用する任意のローカル変数を作成します。
2. ビジュアル テストで、作成する式で使用するデータ列を含むアクティブ データ資産を関連付けます。
3. アクティブ データ ファイルのデータを更新し、元のセルに更新内容を格納する式を作成します。
4. ビジュアル テスト内で、データを更新するステップのうち、直ちに保存を行う最後の 式:ステップを選択します。
5. **挿入 > アクティブ データ > 直ちに保存** を選択します。Silk Test Workbench によって、選択したステップの下に「アクティブ データ ファイル " に値を直ちに保存します」ステップが挿入されます。
6. 作成した「アクティブ データ ファイル " に値を直ちに保存します」ステップを選択します。ステップのプロパティが **プロパティ** ペインに表示されます。
7. アクティブ データ カテゴリ プロパティを設定して、更新された値を直ちに保存するデータ ファイルを含むアクティブ データ資産を指定します。



ヒント: 必要に応じて、「アクティブ データ ファイル " に値を直ちに保存します」ステップの **アクティブ データ** プロパティを使用して、保存オプションや保存するアクティブ データを変更します。

ビジュアル テスト内で更新されたアクティブ データの自動保存の回避

ビジュアル テストの再生中に式内で更新されるアクティブ データは、再生が正常に完了した時点で元のアクティブ データ ファイルに自動保存されます。ただし、更新されたアクティブ データの自動保存を取り消すステップを作成することができます。



注: このステップでは、このステップを実行するまでに更新されたアクティブ データの保存のみが回避されます。このステップの実行後に更新されたアクティブ データに関しては、更新後に自動保存を取り消さないかぎり元のアクティブ データ ファイルに保存されます。

1. ビジュアル テストで、アクティブ データ ファイルのデータを更新し、元のセルに更新内容を格納する式を作成します。
2. ビジュアル テスト内で、データを更新するステップのうち、自動的に保存を行わない最後の 式:ステップを選択します。

3. **挿入 > アクティブ データ > 保存のキャンセル** を選択します。Silk Test Workbench によって、選択したステップの下に「アクティブ データ ファイル " に値を保存しません」ステップが挿入されます。
4. 作成した「アクティブ データ ファイル " に値を保存しません」ステップを選択します。ステップのプロパティが **プロパティ ペイン** に表示されます。
5. ステップの **アクティブ データ** カテゴリ プロパティを設定して、更新された値を直ちに保存しないデータ ファイルを含むアクティブ データ資産を指定します。

ビジュアル テストでアクティブ データに使用するシートを指定する

アクティブ データ資産のソースとして Excel スプレッドシートを定義し、このアクティブ データ資産をビジュアル テストでのアクティブ データ資産テストに使用した場合、ビジュアル テストで実行中に使用するシートを直接指定できます。

1. ビジュアル テストを開きます。
2. **アクティブ データ <資産名> を使用して繰り返します** ステップを選択します。
3. ステップの **プロパティ ペイン** を開きます。
4. **条件** 領域の **シート名** フィールドにシートの名前を入力します。
ビジュアル テストの実行時に、ここで指定したシートで、アクティブ データ資産で指定したシートが上書きされます。このフィールドを空のままにすると、資産で指定したシートが使用されます。
5. ビジュアル テストを保存します。

指定したシートが Excel スプレッドシート内に存在しない場合、ビジュアル テストの実行時に Silk Test Workbench はエラーをスローします。この場合、Excel スプレッドシートを確認して存在するシートを指定してください。

アクティブ データ資産の行数と列数の取得

アクティブ データ資産の行数や列数を取得するには、資産に関連付けられたビジュアル テストに 行数の取得 または 列数の取得 ステップをそれぞれ追加します。


1. これらのステップは、次のようにしてビジュアル テストに追加できます。
 - Silk Test Workbench メニューから、**挿入 > アクティブ データ > 行数の取得** または **挿入 > アクティブ データ > 列数の取得** を選択します。
 - ビジュアル テストの **テスト ステップ** ペインで、**操作 > 挿入 > アクティブ データ > 行数の取得** または **操作 > 挿入 > アクティブ データ > 列数の取得** を選択します。
 - ビジュアル テストの **テスト ステップ** ペインで右クリックして、**挿入 > アクティブ データ > 行数の取得** または **挿入 > アクティブ データ > 列数の取得** を選択します。
2. これらのステップの **プロパティ** ウィンドウで、次の操作を実行します。
 - a) 使用する関連付けられたアクティブ データ資産を選択します。
 - b) 行数と列数のどちらを取得するかを選択します。
 - c) 結果を保存する変数を選択します。
結果は常に `st_LastReturnValue` 予約変数に保存されます。このため、ローカル変数の選択は必須ではありません。また、テスト結果の **詳細** ペインでは、テスト ステップの **結果の詳細** 列にも結果が保存されます。


行数の取得 または 列数の取得 ステップがアクティブ データを使用して繰り返します ループ内で使用される場合、行数はアクティブ データを使用して繰り返します ステップで設定された開始行と終了行に基づきます。たとえば、アクティブ データ ファイルに 10 行保存されている場合に、アクティブ データを使用して繰り返します ループで 2 行目から 6 行目だけを使用するように設定していると、行数の取得 ステップは 5 を返します。

ステップをアクティブ データを使用して繰り返します ループの外で使用した場合は、アクティブ データ ファイルの実際の行数と列数が返されます。

スクリプトでのアクティブ データの使用

アクティブ データを使用するスクリプトの作成処理は、記録機能から独立しています。Silk Test Workbench データベースの外部からのテスト データの格納と取得は、外部ファイルを使用して行われま
す。テスト データは、Excel スプレッドシート ファイル (.xls、.xlsx)、テキスト ファイル (.txt)、または
カンマ区切り値ファイル (.csv) に格納できます。テスト データを外部ファイルに格納すると、複数の
アクティブ データ資産、および任意のビジュアル テストまたはスクリプトで、テスト データを再利用でき
るようになります。

 **注:** Microsoft Excel® がインストールされていない場合、すべての .xls ファイルは読み取り専用モード
になります。Silk Test Workbench で .xlsx ファイルを使用するには、Microsoft Excel® 2007 以
降をインストールしてください。Microsoft Excel® 2007 以降がインストールされていない場合、
2007 Office System ドライバ：データ接続コンポーネント をインストールして、読み取り専用モード
で .xlsx ファイルを使用できます。

 **注:** アクティブ データ ファイルを開くときにエラーが発生した場合は、エラー処理を使用してテスト
実行を続行できます。


アクティブ データ テスト資産の作成

新しいアクティブ データ資産を作成するには、以下を実行します。

1. **表示 > アセットブラウザ** を選択します。
2. **資産の種類** で、**アクティブ データ** を右クリックして、**アクティブ データの新規作成** を選択します。**ア
クティブ データ** ウィンドウが開きます。

3. **名前** テキスト ボックスに、資産のわかりやすい名前を入力します。

アクティブ データ資産名は、Silk Test Workbench 資産の命名規則に従う必要があります。

 **ヒント:** アクティブ データ資産に、その資産に使用されるアクティブ データ ファイルと同じ名前
を付けると、テストで使用する資産を特定しやすくなります。


4. 資産が使用するデータを含んだ新しいアクティブ データ ファイルを作成するか、既存のアクティブ デ
ータ ファイルを選択できます。

- 新しいアクティブ データ ファイルをファイル システムに作成するには、**外部ファイル** を選択してから **新規作成** をクリックします。
- 既存のアクティブ データ ファイルのデータを使用するには、**外部ファイル** を選択してから **参照** を
クリックし、ファイルを選択します。アクセスできるようにするため、ファイルを閉じてください。
- 新しいアクティブ データ資産を Silk Test Workbench データベースに作成するには、**内部データ** を
選択してから **新規作成** をクリックします。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースに格納して、アクティブ デ
ータ資産を使って内部ファイルからデータを使用するには、**内部データ** を選択してから **アップロード**
をクリックし、ファイルを選択します。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースからファイル システムに
コピーするには、**内部データ** を選択してから **ダウンロード** をクリックし、ファイルを選択します。
保存したファイルは、Excel などのアプリケーションを使って編集できます。

ファイル テキスト ボックスには、アクティブ データ ファイルの名前が表示されます。

5. **省略可能 : 説明** テキスト ボックスに、資産の説明を入力します。

説明を使用して、データ ファイルとして使用されるファイルの種類、資産を作成するアクティブ データ
テストによって実行されるテストの種類、またはアクティブ データ テストでテストされるアプリケー
ションを示すことができます。**アセットブラウザ** では、説明が入力されているすべてのアクティブ デ
ータ資産に説明が表示されます。

 **注:** 空のデータ ファイルが必要な場合は、**保存** および **閉じる** をクリックしてすべての変更内容を
保存し、**アクティブ データ** 設定ウィンドウを閉じます。

6. **アクティブ データ** ウィンドウで、**オプション** タブに切り替えます。
7. アクティブ データ ファイルに対して読み取りオプションを設定します。
8. **アクティブ データ** ウィンドウで、**詳細** タブに切り替えます。
9. アクティブ データ ファイル内のデータを編集します。
- 10 **保存** または **データの保存** をクリックしてすべての変更内容を保存するか、または **保存して閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** ウィンドウを閉じます。
 アクティブ データ資産のデータ ファイルが外部の場合、つまりファイル システムに格納されており、Silk Test Workbench データベースではない場合、**データの保存** をクリックすると、データ ファイルだけが保存されます。この場合、アクティブ データ資産は保存されません。

新しいアクティブ データ資産が **アセットブラウザ** に表示されます。

スクリプトへのアクティブ データの組み込み

開始する前に、使用するデータを含むアクティブ データ資産、またはデータを書き込むアクティブ データ資産を作成します。

アクティブ データ資産から使用するスクリプトにデータをマップします。さらに、アクティブ データ資産に行または列を追加して、スクリプトからアクティブ データ資産にデータを追加できます。

1. テストする操作を記録するか、またはテストする操作についてのスクリプトを手動で作成します。
 詳細については、「スクリプトの記録」を参照してください。
 たとえば、Web アプリケーションで姓、名、および電話番号を記録するスクリプトは以下のようになります。

```
Public Sub Main()
  With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
    .DomTextField("@id='txtFirstName']").SetText("Pat")
    .DomTextField("@id='txtLastName']").SetText("Smith")
    .DomTextField("@id='txtPhoneNumber']").SetText("555-121-3434")
    .DomButton("@id='btnAdd']").Select()
  End With
```

2. LoadActiveData コマンドを使用して、アクティブ データ資産名を指定します。
 必要に応じて、アクティブ データ ファイル内の開始行と終了行、使用するランダム行数、およびデータが読み取り専用であるかどうかを指定できます。

```
Workbench.LoadActiveData("activeDataName", [start row,
  end row, randomcount, readonly])
```
3. 記録したテキストをアクティブ データ資産のテキストで置き換えるには、アクティブ データ ファイル内の対応する各行名に対して宣言を作成し、宣言をスクリプト内の適切なコードにマップします。
 たとえば、PhoneBookData アクティブ データ資産で利用可能なすべてのデータを使用するようにステップ 1 のスクリプトを変更するには、スクリプトを以下のように変更します。

```
Public Sub Main()
  AddAllToPhoneBook()          'Reads all records in order
End Sub

Public Sub AddAllToPhoneBook()
  Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
  Dim row As ActiveDataRow

  With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")

    For Each row In data

      Dim FirstName As String = row.GetString("fname")
      Dim LastName As String = row.GetString("lname")
      Dim PhoneNumber As String = row.GetString("phonenumber")
```



```

.DomTextField("@id='txtFirstName").SetText(FirstName)
.DomTextField("@id='txtLastName").SetText(LastName)
.DomTextField("@id='txtPhoneNumber").SetText(PhoneNumber)

.DomButton("@id='btnAdd").Select()
Next
End With
End Sub

```

スクリプトを再生すると、アクティブ データ資産のすべての宣言されているデータがスクリプトに組み込まれます。

4. 既存のアクティブ データ資産に列を追加するには、アクティブ データ資産を宣言して、アクティブ データ資産名と列名を指定します。
たとえば、PhoneBookData 資産に「Occupation」という新しい列を追加するには、以下のように入力します。

```

Public Sub Main()

    AddColumnToDataFile( "occupation" )
End Sub

Public Sub AddColumnToDataFile( name As String )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
    data.AddColumn( "Occupation", name )
    data.Save()
End Sub
End Module

```

スクリプトを再生すると、アクティブ データ資産によって指定されたファイルに新しい列が追加されます。

5. 既存のアクティブ データ資産に行を追加するには、アクティブ データ資産を宣言して、アクティブ データ資産名と行名を指定します。
たとえば、特定の ID 番号の新しいデータ行を追加するには、以下のように入力します。

```

Public Sub Main()

    InsertNewNumberToDataFileWithOccupation( 108, "Jay", "Jones", "5551219", _
        "QA Analyst" )
End Sub

Public Sub InsertNewNumberToDataFileWithOccupation( id As Integer, _
    firstName As String, lastName As String, phoneNumber As String, _
    occupation As String )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
    Dim row As ActiveDataRow = data.AddRow()

    row.SetLong( "id", id )
    row.SetString( "fname", firstName )
    row.SetString( "lname", lastName )
    row.SetString( "number", phoneNumber )
    row.SetString( "occupation", occupation )

    data.Save()
End Sub
End Module

```

スクリプトを再生すると、アクティブ データ資産によって指定されたファイルに新しい行が追加されます。

6. 特定のセルに変数の値を追加するには、スクリプトに次のコード サンプルを追加します。

```
Dim row As ActiveRecord = data.Item(1)
row.SetString("Columnname", Variable)
```

VB .NET スクリプトでアクティブ データに使用するシートを指定する

アクティブ データ資産のソースとして Excel スプレッドシートを定義し、このアクティブ データ資産を VB .NET スクリプトでのアクティブ データ資産テストに使用した場合、LoadActiveData メソッドの sheetName パラメータを使用して、実行中に使用するシートを指定できます。詳細については、「LoadActiveData メソッド」を参照してください。

指定したシートが Excel スプレッドシート内に存在しない場合、VB .NET スクリプトの実行時に Silk Test Workbench はエラーをスローします。この場合、Excel スプレッドシートを確認して存在するシートを指定してください。

ビジュアル テストの拡張


このセクションでは、ビジュアル テストを変更して、より高度な機能を実行する方法を示します。

ビジュアル テスト ステップで実行される操作の変更

操作の選択 ダイアログ ボックスを使用して、ビジュアル テストのステップのコントロールで実行される操作を変更できます。

リストには以下の列が含まれています。

操作をクリックして選択し、**OK** をクリックすると、テスト ステップでコントロールに対してその操作が実行されるようになります。

 **注:** コントロールに対して、選択された操作を特定の方法で実行するために、ほとんどの操作で追加パラメータが使用されます。操作のパラメータは、**パラメータ** カテゴリにプロパティとして表示され、**プロパティ** ウィンドウで設定できます。

1. 変更する操作が含まれているビジュアル テストを開きます。
2. **テスト ステップ** ペインで、変更する操作を含むステップを選択します。
3. **プロパティ** ペインを開きます。
4. ペインの **操作** 列で、**メソッド** をクリックします。
5. **操作の選択** をクリックします。**操作の選択** ダイアログ ボックスが開き、現在のテスト ステップがやり取りする UI コントロールで利用可能なすべての操作が一覧表示されます。リストの各列のテキストを確認できるようダイアログ ボックスのサイズを変更できます。
6. UI コントロールで実行したい操作を選択します。
7. **OK** をクリックします。

ビジュアル テストへのコメントの挿入

コメントによって、ビジュアル テストにステップについての補足情報を定義できます。既存のビジュアル テストにステップとしてコメントを挿入できます。

1. ビジュアル テストの **テスト ステップ** ペインで、コメントを後ろに挿入するステップを選択します。
2. **挿入 > コメント** を選択します。**テスト ステップ** ペインに コメント ステップが追加され、このステップのプロパティが **プロパティ** ペインに表示されます。
3. **プロパティ** ペインで、**コメント** フィールドにコメントのテキストを入力します。

フラグ

フラグを使用して、ビジュアル テスト内または結果内の問題について自分自身や別のチーム メンバに注意を促すことができます。問題にフラグを付けると、挿入したフラグの種類に応じて、**フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンが追加されます。

フラグには以下の情報が含まれます。

- フラグを作成または更新したユーザーの名前
- フラグの作成日または更新日
- フラグの説明（オプション）
- フラグが割り当てられたユーザーの名前（オプション）
- ビジュアル テストまたは結果が保存されているプロジェクト（オプション）

フラグ のアイコンは、**テスト ステップ** ペインの **フラグ** 列に表示されます。フラグにポインタを合わせると、ツール ヒントとしてフラグの内容が表示されます。

以下のいずれかの方法を使用して、フラグを挿入します。

- ビジュアル テストまたは結果の **テスト ステップ** ペインでステップを選択し、フラグを挿入します。

必要に応じて、フラグに説明を付けたり、別のユーザーにフラグを割り当てたりできます。割り当てられたフラグは、ビジュアル テストまたは結果の選択したステップ、およびフラグを割り当てたユーザーの **開始画面** の **フラグ** ペインに表示されます。**開始画面** では、自身に割り当てられているすべてのフラグをすばやく参照したり、フラグをダブルクリックして対象のビジュアル テストまたは結果を開いたりできます。

- 再生時に失敗した時にビジュアル テストの結果にフラグを送信する検証ロジックをビジュアル テストに作成します。

結果では、検証ロジックを含むステップにフラグが挿入されます。このステップは、結果の **失敗** タブ、**フラグ** タブ、および **詳細** タブの **テスト ステップ** ペインに表示されます。必要に応じて、ウィザードでフラグの説明を作成できます。デフォルトでは、このフラグは自動的に現在のユーザーに割り当てられます。フラグを別のユーザーに割り当てる場合は、再生後に結果から再割り当てする必要があります。



注: フラグはビジュアル テストでのみサポートされており、Visual Basic .NET スクリプトではサポートされていません。

[テスト ステップ] ペインへのフラグの挿入

ビジュアル テストまたは結果での問題を忘れないように、フラグを挿入できます。

1. フラグを挿入するビジュアル テストまたは結果を開きます。
2. ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、フラグを挿入するステップを選択します。
3. **操作 > フラグ > 挿入** をクリックします。
また、フラグを挿入するステップの **フラグ** 列のセルをダブルクリックして、**フラグ** ダイアログ ボックスにアクセスすることもできます。
フラグ ダイアログ ボックスが開きます。
4. 省略可能： **説明** テキスト ボックスにフラグの説明を入力します。
この説明は、ポインタを **フラグ** アイコンまたは **割り当てられたフラグ** アイコンに合わせると、ツール ヒントとして表示されます。
5. 省略可能： **開始画面に表示してユーザーに割り当てる** チェック ボックスをオンにし、ユーザーを選択して、別のユーザーにフラグを割り当てます。フラグは、割り当て先のユーザーの **開始画面** に表示されます。
6. **OK** をクリックします。**フラグ** ダイアログ ボックスで選択したオプションに応じて、**フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンが、選択したステップの **フラグ** 列に表示されます。結果の

フラグの場合、OK をクリックするとすぐにフラグが保存されます。ビジュアル テストのフラグの場合、フラグを保存するにはビジュアル テストを保存する必要があります。

結果へのフラグの挿入

再生時に検証ロジックに失敗すると、Silk Test Workbench によって、ビジュアル テストの結果にフラグが送信されます。

1. ビジュアル テストを作成または記録します。
2. **テスト ロジック デザイナ** ウィザードを使用して、検証ロジックを挿入します。
3. **検証に失敗した場合に、結果に 'フラグ' を作成する** チェック ボックスをオンにし、フラグの説明を指定します。
4. ビジュアル テストを再生します。再生時に検証ロジックに失敗すると、Silk Test Workbench によって、ビジュアル テストの結果にフラグが送信されます。結果では、検証ロジックを含むステップにフラグが挿入されます。このステップは、**結果** ウィンドウの **失敗** タブ、**フラグ** タブ、および **詳細** タブに表示されます。

フラグの表示


フラグの内容を表示します。

1. フラグを表示するビジュアル テストまたは結果を開きます。
ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、フラグが付けられたステップには **フラグ** 列に **フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンが表示されます。
2. **フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンにポインタを合わせます。ツール ヒントにフラグの内容が表示されます。

開始画面からのフラグの表示

開始画面 からフラグの内容を表示します。

1. **開始画面** を開きます。デフォルトでは、フラグは作成された日付と表示される資産でグループ化され、**フラグ** ペインに表示されます。

 **注:** **フラグ** ペインの表示を変更するには、**開始画面のフラグ** オプションを変更します。このオプションは、**開始画面** の **ヘルプ** ペインの下にある **カスタマイズ** ボタンをクリックして、**開始画面** からアクセスできます。

2. 表示するフラグに移動して、**割り当てられたフラグ** のアイコンにポインタを合わせます。ツール ヒントにフラグの内容が表示されます。
3. フラグが挿入されているビジュアル テストまたは結果を開くには、**割り当てられたフラグ** のアイコンをダブルクリックします。

フラグの編集

必要に応じて、フラグの内容を変更します。

1. 編集するフラグを含むビジュアル テストまたは結果を開きます。
2. ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、編集するフラグを含むステップを選択します。
3. **操作** > **フラグ** > **編集** をクリックします。**フラグ** ペインが開きます。また、**フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンをダブルクリックして **フラグ** ダイアログ ボックスを開くこともできます。
4. 省略可能：説明 テキスト ボックスにフラグの説明を入力します。
この説明は、ポインタを **フラグ** アイコンまたは **割り当てられたフラグ** アイコンに合わせると、ツール ヒントとして表示されます。

5. 省略可能 : **開始画面に表示してユーザーに割り当てる** チェック ボックスをオンにし、ユーザーを選択して、別のユーザーにフラグを割り当てます。フラグは、割り当て先のユーザーの **開始画面** に表示されます。
6. **OK** をクリックします。結果のフラグの場合、**OK** をクリックするとすぐにフラグが保存されます。ビジュアル テストのフラグの場合、フラグを保存するにはビジュアル テストを保存する必要があります。

フラグの削除

フラグが不要になったら、削除します。

1. 削除するフラグを含むビジュアル テストまたは結果を開きます。
2. ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、削除するフラグを含むステップを選択します。
3. **操作 > フラグ > 削除** をクリックします。フラグの削除を確認するメッセージが表示されます。
4. **はい** をクリックします。結果のフラグの場合、**OK** をクリックするとすぐにフラグが削除されます。ビジュアル テストのフラグの場合、フラグを永久的に削除するにはビジュアル テストを保存する必要があります。

テスト ロジックの追加

記録されたビジュアル テストの基本的な機能を拡張するテスト ロジックを含めることで、ビジュアル テストの機能と柔軟性を強化できます。テスト ロジックを挿入すると、ビジュアル テストの実行フローを動的に変更できます。たとえば、ユーザー定義の条件を評価するテスト ロジックをビジュアル テストに挿入して、その条件の結果に基づいて以下のいずれかの操作を実行できます。

- ビジュアル テストの一連のテスト ステップを実行したり、繰り返したりする
- ビジュアル テストからビジュアル テストを実行する
- 合格/失敗メッセージ、およびテスト ステップの再生ステータスに関するフラグをビジュアル テストの結果に送信する

これらの操作を実行するために、Silk Test Workbench にはさまざまなタイプのテスト ロジックが用意されています。これらのロジックタイプは以下のカテゴリに分けられます。

- **判断ロジック** : ユーザー定義の条件を評価して、その条件の戻り値に応じて一連のテスト ステップを実行するロジックです。
- **繰り返しロジック** : 指定した回数または時間、あるいはユーザー定義の条件が true である間、一連のテスト ステップを繰り返します。
- **検証ロジック** : ユーザー定義の条件を評価して、合格/失敗メッセージとフラグをビジュアル テストの結果に送信します。
- **エラー処理ロジック** : 再生中にエラーが発生した場合、ユーザー定義の操作を実行します。

Silk Test Workbench には、テスト ロジックをビジュアル テストに挿入するために、以下のツールが用意されています。

- **テストロジックデザイナーウィザード**。
- **ロジックツールボックス**。
- 判断テスト ステップ *If*、*Else If*、*Else*。

判断ロジック

判断ロジックでは、ユーザー定義の条件を評価して、その条件の戻り値に応じて一連のテスト ステップを実行します。判断ロジックで使用される条件は、2 つの値を比較し、その結果に基づいて値を返します。詳細については、「条件デザイナー」を参照してください。条件内の値には、テスト アプリケーション内のコントロールのプロパティの値、ビジュアル テストで使用される変数の値、またはリテラル値のいずれかを使用できます。詳細については、「テストロジックデザイナー : ロジックの種類を選択 (判断)」を参照してください。

判断ロジックは、以下の要素で構成されます。

```
If <条件=TRUE>
  __<テスト ステップを順番に実行>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、ステップ 5 から 9 が実行されます。

```
If propertyVar より大きい 4
  __ステップ 5 からステップ 9 まで実行
End If
```

また、Else If と Else を使用して、任意のレベル数にネストした複雑な判断ロジックを作成することもできます。

```
If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else
  __<テスト ステップを順番に実行>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、ステップ 5 から 9 が実行されます。「4」より小さい場合は、ステップ 10 から 14 が実行されます。「4」と等しい場合は、ステップ 15 が実行されます。

```
If propertyVar より大きい 4
  __ステップ 5 からステップ 9 まで実行
Else If propertyVar より小さい 4
  __ステップ 10 からステップ 14 まで実行
Else
  __ステップ 15 を実行
End If
```

判断ロジックは、**テストロジックデザイナー** ウィザード、ロジックツールボックス、または If...Else If...Else テスト ステップを使用して作成できます。

ビジュアルテストへの判断ロジックの追加

このトピックでは、**テストロジックデザイナー** を使用して、ビジュアルテストに判断ロジックを追加する方法について説明します。判断ロジックテストステップをビジュアルテストに直接追加する方法については、「[判断テストステップを使用した判断ロジックの追加](#)」を参照してください。

1. ビジュアルテストを開きます。
2. テストロジックに含める1つのテストステップまたは一連のテストステップで右クリックします。
3. **挿入 > テストロジック > 判断** を選択します。**テストロジックデザイナー** ウィザードが開きます。
4. **次へ** をクリックします。
5. **ロジックの種類**の選択 ページで、判断ロジックの種類を選択します。
 - **コントロールのプロパティ** : テストアプリケーションのコントロールのプロパティに基づいて、一連のステップを実行する判断ロジックを作成します。例：

```
If propertyValue より大きい 2 ステップ 3 からステップ 4 まで実行 End If
```
 - **コントロールの存在確認** : 指定したコントロールが存在するかないかに基づいて、一連のステップを実行する判断ロジックを作成します。例：

```
"Button" が存在する 等しい True ステップ 3 からステップ 4 まで実行 End If
```
 - **このビジュアルテストで定義した変数** : ビジュアルテストに定義されているローカル変数の内容に基づいて一連のステップを実行する判断ロジックを作成します。例：

```
If stringVar = "red" ステップ 3 からステップ 4 まで実行 End If
```

- **高度なロジック** : ウィザードの条件の定義ステップをスキップし、**条件デザイナー** を使用して、詳細な判断ロジックを作成します。例 :

```
If stringVar = "red" および numberVar = 10 ステップ 3 からステップ 4 まで実行 End If
```

6. **次へ** をクリックします。

7. **コントロールのプロパティ** を選択した場合、**プロパティに基づく条件の定義** ページが開きます。

- a) **コントロール名** フィールドの右側にある **コントロールの識別** メニューを展開します。
- b) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション** : テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー** : テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ** : 非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- c) **プロパティの選択** テーブルで、プロパティを選択します。
- d) **条件の選択** リストから、コントロールのプロパティに適用する条件ロジックを選択します。
- e) **期待値** フィールドには、プロパティに期待する値を指定します。
- f) **次へ** をクリックします。

8. **コントロールの存在確認** を選択した場合、**コントロールの存在条件の定義** ページが開きます。

- a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション** : テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー** : テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ** : 非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。
- c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。
- d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。
デフォルトのタイムアウトは 0 で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。
- e) **次へ** をクリックします。

9. **このビジュアル テストで定義した変数** を選択した場合、**変数に基づく条件の定義** ページが開きます。

- a) **変数の選択** リストから事前に定義されたローカル変数を選択します。
このリストに変数を表示するには、ビジュアル テストにローカル変数を含める必要があります。ビジュアル テストに新しいローカル変数を追加する場合は、**新規作成** をクリックします。
評価結果 フィールドには、選択した変数のデータ型が表示されます。
- b) **条件の選択** から変数に適用する比較演算子を選択します。
- c) **期待値** フィールドには、変数に期待する値を指定します。
- d) **次へ** をクリックします。

10 **高度なロジック** を選択した場合、**次へ** をクリックします。

11 **判断の構築** ページが開きます。条件ロジック、および条件ロジックの結果が真となる場合に実行する、一連のステップを指定します。



ヒント: ポインタをリンクの上に置くと、完全なリンクの説明を含むツール ヒントが表示されます。

- a) **条件デザイナー** で条件ロジックを指定するには、**If** 文の右側にあるリンクをクリックします。
- b) 実行する一連のステップの最初のステップを指定するには、**ステップの開始** の右側にあるリンクをクリックします。
- c) 実行する一連のステップの最後のステップを指定するには、**ステップの終了** の右側にあるリンクをクリックします。
- d) 追加の条件ロジックを挿入するには、**Else If の追加** をクリックします。
用意されたリンクをクリックして **条件デザイナー** を開き、条件が True であるときに実行する一連のステップを指定します。
- e) 追加の条件ロジックを挿入するには、**Else の追加** をクリックします。
用意されたリンクを使用して、条件が False であるときに実行する一連のステップを指定します。
- f) 追加の条件ロジックおよび一連のステップを削除するには、**削除** をクリックします。
このボタンは、追加の条件ロジックを挿入したあとにのみ表示されます。

12次へ をクリックします。**要約** ページに、定義したテスト ロジックが表示されます。このページの最下部にあるチェック ボックスをオフにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェック ボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

13完了 をクリックして、ビジュアル テストのテスト ステップにテスト ロジックを挿入します。テスト ロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テスト ロジックがビジュアル テストに追加されます。

判断テスト ステップを使用した判断ロジックの追加

If、Else If、および Else テスト ステップを挿入することですばやく判断ロジックを作成できます。これらのテスト ステップを使用すると、**テスト ロジック デザイナ** ウィザードとは別の方法で判断ロジックを作成できます。また、**テスト ロジック デザイナ** ウィザードを使用しなくても、既存の判断ロジックをすばやく変更できます。

Else If および Else テスト ステップでは、任意のレベル数にネストした複雑な判断ロジックを作成できます。

```
If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else
  __<テスト ステップを順番に実行>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、ステップ 5 から 9 が実行されます。「4」より小さい場合は、ステップ 10 から 14 が実行されます。「4」と等しい場合は、ステップ 15 が実行されます。

```
If propertyVar より大きい 4
  __ステップ 5 からステップ 9 まで実行
Else If propertyVar より小さい 4
  __ステップ 10 からステップ 14 まで実行
Else
  __ステップ 15 を実行
End If
```

ビジュアル テストに判断テスト ステップをすばやく挿入するには、以下を実行します。


1. ビジュアル テストを開き、判断ロジックに含めるテスト ステップを選択します。

一連のテスト ステップを選択するには、最初のステップを選択し、Shift キーを押しながら最後のテスト ステップを選択します。最初と最後のテスト ステップの間のテスト ステップはすべて自動的に選択されます。

2. ツールバーで **If タイプのロジック項目の挿入** をクリックします。
選択したテスト ステップが、If テスト ステップと End If テスト ステップで囲まれます。
3. 省略可能 : Else If または Else テスト ステップを If...End If ロジックに挿入するには、Else If または Else テスト ステップを挿入する場所の直前のテスト ステップを選択し、**Else If タイプのロジック項目の挿入** または **Else タイプのロジック項目の挿入** をクリックします。

判断ロジック テスト ステップの編集

テスト ロジック デザイナー ウィザードまたは **条件デザイナー** を使用してテスト ロジック ステップを作成したあと、**プロパティ** ペインから直接そのテスト ステップを編集できます。

 **注:** **条件** プロパティは、If および Else If テスト ステップにのみ含まれます。Else および End If テスト ステップには含まれません。

1. 編集するテスト ステップが含まれているビジュアル テストを開きます。
2. ビジュアル テストの **テストステップ** ウィンドウで、編集するテスト ステップを選択します。選択したテスト ステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティ カテゴリごとに表示されます。
3. 使用可能な条件プロパティまたはその他のプロパティを編集します。

繰り返しロジック

繰り返しロジックは、指定した回数、またはユーザー定義の条件が true になるまで、一連のテスト ステップを繰り返します。繰り返しロジックで使用される条件は、2 つの値を比較し、その結果に基づいて値を返します。詳細については、「条件デザイナー」を参照してください。条件内の値には、テスト アプリケーション内のコントロールのプロパティの値、ビジュアル テストで使用される変数の値、リテラル値、またはアクティブ データ資産からの値のいずれかを使用できます。

指定した回数だけ一連のテスト ステップを繰り返す繰り返しロジックは、以下の要素で構成されます。

```
このループを <回数> 回繰り返します
__<テスト ステップを順番に実行>
繰り返しの終了
```

たとえば、以下の繰り返しロジックでは、テスト ステップのステップ 10 からステップ 15 までが繰り返されます。

```
このループを 10 回繰り返します
__ステップ 10 から
__ステップ 15 まで実行
繰り返しの終了
```

ユーザー定義の条件が True の間、または True になるまで一連のテスト ステップを繰り返す繰り返しロジックは、以下の要素で構成されます。

```
<条件=TRUE> の間繰り返します
__<テスト ステップを順番に実行>
繰り返しの終了
```

または、以下のようにループの最後に条件を指定できます。

```
繰り返します
__<テスト ステップを順番に実行>
<条件=TRUE> の間繰り返して終了
```

たとえば、以下の繰り返しロジックでは、変数 *propertyVar* の値がリテラル値「4」より小さい間は、一連のテスト ステップが繰り返されます。*propertyVar* が「4」より大きいか等しくなると、ループは実行されません。

```
propertyVar より小さい 4 の間繰り返します
__ステップ 10 から
__ステップ 14 まで実行
繰り返しの終了
```

または、ループの最後に条件を指定できます。以下の場合、`propertyVar` が「4」より大きいか等しいと、ループが 1 回実行されます。

```
繰り返します
__ステップ 10 から
__ステップ 14 まで実行
propertyVar より小さい 4 の間繰り返して終了
```

繰り返しロジックは、**テスト ロジック デザイナ** ウィザードまたはロジック ツールボックスを使用して作成できます。

ビジュアル テストへの繰り返しロジックの追加

このトピックでは、**テスト ロジック デザイナ** を使用して、ビジュアル テストに繰り返しロジックを追加する方法について説明します。

1. ビジュアル テストを開きます。
2. 繰り返したい 1 つのテスト ステップまたは一連のテスト ステップで右クリックします。
3. **挿入 > テスト ロジック > 繰り返し** を選択します。**テスト ロジック デザイナ** ウィザードが開きます。
4. **次へ** をクリックします。
5. **ロジックの種類を選択** ページで、繰り返しロジックの種類を選択します。

以下のオプションに基づいて、繰り返しロジックのタイプを選択します。

- **コントロールのプロパティ** : テスト アプリケーションのコントロールのプロパティに基づいて、一連のステップを繰り返す繰り返しロジックを作成します。例：
propertyValue 等しい 3 の間繰り返します __ステップ 3 から __ステップ 4 まで実行 繰り返しの終了
- **コントロールの存在確認** : 指定したコントロールが存在するかないかに基づいて、一連のステップを繰り返す繰り返しロジックを作成します。例：
"Button" が存在する 等しい True の間繰り返します __ステップ 3 から __ステップ 4 まで実行 繰り返しの終了
- **このビジュアル テストで定義した変数** : ビジュアル テストに定義されているローカル変数の内容に基づいて一連のステップを繰り返す繰り返しロジックを作成します。例：
numberVar より小さい 5 の間繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了
- **ステップ シーケンスを設定した回数繰り返す** : 一連のステップを設定した回数繰り返す繰り返しロジックを作成します。例：
このループを 10 回繰り返します __ステップ 3 から __ステップ 7 まで実行 繰り返しの終了
- **リストまたはコンボ ボックスの内容** : リスト ボックスまたはコンボ ボックス内の各項目に対して、一連のステップを繰り返す繰り返しロジックを作成します。例：
ListA の各項目ごとに繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了
- **アクティブ データ ファイルを使用してステップ シーケンスを繰り返す** : ビジュアル テストで使用するアクティブ データ ファイルのデータに基づいて、一連のステップを繰り返す繰り返しロジックを作成します。例：
アクティブ データ testData を使用して繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了
- **高度なロジック** : ウィザードの条件の定義 ステップをスキップし、**条件デザイナ** を使用して、詳細な繰り返しロジックを作成します。例：
propertyValue = red または stringVar = red の間繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了

6. **次へ** をクリックします。
7. **コントロールのプロパティ** を選択した場合、**プロパティに基づく条件の定義** ページが開きます。

- a) **コントロール名** フィールドの右側にある **コントロールの識別** メニューを展開します。
 - b) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
 - c) **プロパティの選択** テーブルで、プロパティを選択します。
 - d) **条件の選択** リストから、コントロールのプロパティに適用する条件ロジックを選択します。
 - e) **期待値** フィールドには、プロパティに期待する値を指定します。
 - f) **次へ** をクリックします。
8. **コントロールの存在確認** を選択した場合、**コントロールの存在条件の定義** ページが開きます。
- a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
 - b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。
 - c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。
 - d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。
デフォルトのタイムアウトは 0 で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。
 - e) **次へ** をクリックします。
9. このビジュアル テストで定義した変数 を選択した場合、**変数に基づく条件の定義** ページが開きます。
- a) **変数の選択** リストから事前に定義されたローカル変数を選択します。
このリストに変数を表示するには、ビジュアル テストにローカル変数を含める必要があります。ビジュアル テストに新しいローカル変数を追加する場合は、**新規作成** をクリックします。
評価結果 フィールドには、選択した変数のデータ型が表示されます。
 - b) **条件の選択** から変数に適用する比較演算子を選択します。
 - c) **期待値** フィールドには、変数に期待する値を指定します。
 - d) **次へ** をクリックします。
10. **ステップ シーケンスを設定した回数繰り返す** を選択した場合、**繰り返し回数の定義** ページが開きます。
- a) 1 から 32767 の数値を入力して、一連のステップを繰り返す回数を指定します。
 - b) **次へ** をクリックします。
11. **リストまたはコンボ ボックスの内容** を選択した場合、**コントロールに基づく繰り返しの定義** ページが開きます。
- a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。

- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

いずれのタイプのコントロールでも、リスト ボックスまたはコンボ ボックスの各項目の値を使用する繰り返しロジックを作成できます。たとえば、リスト ボックスまたはコンボ ボックスの値に対して反復処理を行い、各項目の値をテスト アプリケーションのテキスト ボックスに挿入する繰り返しループを作成できます。このようにして、リスト ボックスまたはコンボ ボックスの内容全体をすばやくテストできます。

b) **次へ** をクリックします。

12 アクティブ データ ファイルを使用してステップ シーケンスを繰り返す を選択した場合、使用するアクティブ データ資産の定義 ページが開きます。

使用するアクティブ データ資産の定義 ページを使用して、繰り返しロジックで使用するアクティブ データ資産とアクティブ データ資産内のデータの取得方法を定義します。

a) リストから **アクティブ データ資産** を選択します。

アクティブ データ資産をこのリストに表示するためには、まずビジュアル テストにアクティブ データ資産を含める必要があります。

b) 使用するデータ ファイルのシートを、**シート名** リストから選択します。

デフォルトでは、アクティブ データ資産で指定されているシートが使用されます。

c) 使用するアクティブ データ ファイルの最初のデータ行の番号を、**開始行** フィールドに入力します。

d) アクティブ データ ファイルの最後のデータ行から開始するには、**データを含む最後の行から開始** チェック ボックスをオンにします。

ファイルの最後の行が Silk Test Workbench によって自動的に検出され、アクティブ データ ファイルの行に対し逆方向で反復処理が行われます。たとえば、アクティブ データ ファイルに 10 行が含まれているときにこのオプションを選択した場合は、行 10、行 9、行 8 という順番で反復処理が Silk Test Workbench によって行われます。

e) 使用するアクティブ データ ファイルの最後のデータ行の番号を、**終了行** フィールドに入力します。

f) アクティブ データ ファイルの最後のデータ行で終了するには、**データを含む最後の行で終了** チェック ボックスをオンにします。

ファイルの最後の行が Silk Test Workbench によって自動的に検出されます。



ヒント: **データを含む最後の行から開始** チェック ボックスまたは **データを含む最後の行で終了** がオンになっている場合は、繰り返し ステップの Start Row プロパティまたは End Row プロパティの値が -1 に設定されます。

g) 行が処理される順番を選択します。

- **順序通りにすべての行を取得**：アクティブ データ ファイルで、選択した行の最初に指定した行から最後に指定した行まで順番に反復処理を行うには、このオプションを選択します。
- **ランダムな順序ですべての行を取得**：アクティブ データ ファイルのすべての行でランダムに反復処理を行うには、このオプションを選択します。
- **ランダムに X 行を取得**：指定した行数に対してランダムに反復処理を行うには、このオプションを選択してアクティブ データ ファイルの行数を入力します。

h) **次へ** をクリックします。

13 高度なロジック を選択した場合、**次へ** をクリックします。

14 繰り返しの構築 ページが開きます。繰り返す一連のステップを指定します。



ヒント: ポインタをリンクの上に置くと、完全なリンクの説明を含むツール ヒントが表示されます。

a) 繰り返し期間では、ループの開始または終了時に確認される繰り返し条件を選択します。

b) 繰り返し期間に対して、**条件デザイナー** で条件ロジックを指定するには、**繰り返し期間** 文の右側にあるリンクをクリックします。

c) 繰り返す一連のステップの最初のステップを指定するには、**ステップの開始** の右側にあるリンクをクリックします。

d) 繰り返す一連のステップの最後のステップを指定するには、**ステップの終了**の右側にあるリンクをクリックします。

15次へ をクリックします。**要約** ページに、定義したテスト ロジックが表示されます。このページの最下部にあるチェック ボックスをオフにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェック ボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

16完了 をクリックして、ビジュアル テストのテスト ステップにテスト ロジックを挿入します。テスト ロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テスト ロジックがビジュアル テストに追加されます。

リスト ボックスまたはコンボ ボックスの内容に基づいた繰り返しロジックの作成

繰り返しロジックで使用するリスト ボックス コントロールまたはコンボ ボックス コントロールを特定します。いずれのタイプのコントロールでも、リスト ボックスまたはコンボ ボックスの各項目の値を使用する繰り返しロジックを作成できます。たとえば、リスト ボックスまたはコンボ ボックスの値に対して反復処理を行い、各項目の値をテスト アプリケーションのテキスト ボックスに挿入する繰り返しループを作成できます。このようにして、リスト ボックスまたはコンボ ボックスの内容全体をすばやくテストできます。

1. ビジュアル テストを記録します。
2. リスト ボックスまたはコンボ ボックス コントロールに基づいて繰り返しロジックを作成するには、**テスト ロジック** ウィザードを使用します。
3. リテラル値を予約変数 `st_RepeatListValue` に置き換えて、リスト ボックスまたはコンボ ボックスの内容を使用する繰り返しテスト ステップを変更します。
予約変数 `st_RepeatListValue` は、繰り返しループを反復するたびに、リスト ボックスまたはコンボ ボックスの各項目の値を格納するために使用します。`st_RepeatListValue` は、項目のインデックスとテキストの両方が含まれている複合値を返します。
4. テスト ステップを変更するには、繰り返すテスト ステップを選択します。
5. **プロパティ** ウィンドウで、変更するテスト ステップの自動化プロパティを選択します。
6. **選択** をクリックして、**変数** を選択します。**変数の選択** ダイアログ ボックスが開きます。
7. **変数の種類** リストから **予約** を選択し、**st_RepeatListValue** を選択します。
8. **OK** をクリックします。Silk Test Workbench によって、テスト ステップのテキストおよびプロパティ 値が、選択した変数名に更新されます。
9. ビジュアル テストを実行します。

繰り返しロジック テスト ステップの編集

テスト ロジック デザイナ ウィザードまたは **条件デザイナ** を使用してテスト ロジック ステップを作成したあと、**プロパティ** ペインから直接そのテスト ステップを編集できます。

1. 編集するテスト ステップが含まれているビジュアル テストを開きます。
2. ビジュアル テストの **テスト ステップ** ウィンドウで、編集する繰り返しロジック テスト ステップを選択します。選択したテスト ステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティ カテゴリごとに表示されます。
3. 使用可能な条件プロパティまたはその他のプロパティを編集します。
繰り返し テスト ステップには、条件プロパティとその他のプロパティの両方が含まれています。繰り返しの終了には、その他のプロパティのみが含まれています。

検証ロジック

検証ロジックは、ユーザー定義の条件を評価して、合格/失敗メッセージをビジュアル テストまたはスクリーンショットの結果に送信します。必要に応じて、失敗のテスト ステップにフラグを付けて、ビジュアル テストの結果に表示することもできます。検証ロジックで使用される条件は、2つの値を比較し、その結果に基

づいて値を返します。詳細については、「条件デザイナ」を参照してください。条件内の値には、テストアプリケーション内のコントロールのプロパティの値、ビジュアルテストで使用される変数の値、または一連のテストステップの実行にかかる時間のいずれかを使用できます。

検証ロジックは、以下の要素で構成されます。

```
If <条件=TRUE>
  __<合格メッセージを結果に送信>
Else
  __<失敗メッセージとフラグを結果に送信>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、合格メッセージが結果に送信されます。値が「4」より小さい場合は、失敗メッセージとフラグが送信されます。

```
If propertyVar より大きい 4
  __<合格メッセージを送信>
Else
  __<失敗メッセージとフラグを送信>
End If
```

検証ロジックは、**テストロジックデザイナ** ウィザードまたはロジックツールボックスを使用して作成できます。ビジュアルテストの記録中に検証ロジックを作成することもできます。

ビジュアルテストへの検証ロジックの追加

このトピックでは、**テストロジックデザイナ** を使用して、ビジュアルテストに検証ロジックを追加する方法について説明します。どの種類の検証ロジックも、ユーザー定義の合格/失敗メッセージを送信し、フラグをビジュアルテストの結果に設定します（任意）。

1. ビジュアルテストを開きます。
2. 検証したいオブジェクトを含むテストステップで右クリックします。
3. **挿入 > テストロジック > 検証** を選択します。**テストロジックデザイナ** ウィザードが開きます。
4. **次へ** をクリックします。
5. **ロジックの種類**の選択 ページで、検証ロジックの種類を選択します。

以下のオプションに基づいて、検証ロジックのタイプを選択します。

- **コントロールのプロパティ**：コントロールのプロパティの値に基づいて検証ロジックを作成します。例：
「"Button"."Text" 等しい "Cancel"」を検証します
- **コントロールの存在確認**：指定したコントロールが存在するかどうかを検査する検証ロジックを作成します。例：
「"Button" が存在する 等しい True」を検証します
- **このビジュアルテストで定義した変数**：ビジュアルテストに定義されているローカル変数の内容に基づいて検証ロジックを作成します。例：
「st_LastValidation 等しい False」を検証します
- **Rumba 画面のコンテンツ**：Rumba 画面のコンテンツを比較するための検証ロジックを作成します。
- **検証資産**：UI のコンテンツを資産と比較するための検証ロジックを作成します。現時点では、Silk Test Workbench は、検証資産としてイメージ検証のみをサポートしています。詳細については、「資産の種類」を参照してください。例：
'Untitled_1' という名前のイメージ検証を実行します
- **ステップシーケンスの実行にかかる時間**：ステップシーケンスの実行にかかる時間に基づいて検証ロジックを作成します。例：
「タイマー番号 '1' が 0 から 2 秒の間にある」を検証します __ステップ 3 から __ステップ 4 を実行します

6. **次へ** をクリックします。

7. **コントロールのプロパティ** を選択した場合、**プロパティに基づく条件の定義** ページが開きます。

a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

b) さらに、**コントロール名** フィールドにコントロールの完全ロケータを入力して、**更新** をクリックすることもできます。テスト対象アプリケーションでコントロールが見つかったら、**プロパティの選択** リストがコントロールのプロパティで更新されます。

c) **プロパティの選択** テーブルで、プロパティを選択します。

d) **条件の選択** リストから、コントロールのプロパティに適用する条件ロジックを選択します。

e) **期待値** フィールドには、プロパティに期待する値を指定します。

f) **次へ** をクリックします。

8. **コントロールの存在確認** を選択した場合、**コントロールの存在条件の定義** ページが開きます。

a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。

c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。

d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。

デフォルトのタイムアウトは0で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。

e) **次へ** をクリックします。

9. このビジュアル テストで定義した変数を選択した場合、**変数に基づく条件の定義** ページが開きます。

a) **変数の選択** リストから事前に定義されたローカル変数を選択します。

このリストに変数を表示するには、ビジュアル テストにローカル変数を含める必要があります。ビジュアル テストに新しいローカル変数を追加する場合は、**新規作成** をクリックします。

評価結果 フィールドには、選択した変数のデータ型が表示されます。

b) **条件の選択** から変数に適用する比較演算子を選択します。

c) **期待値** フィールドには、変数に期待する値を指定します。

d) **次へ** をクリックします。

10 Rumba 画面のコンテンツを選択した場合、Rumba 画面上のコントロールを識別します。

a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。

- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

b) **次へ** をクリックします。

11 検証資産 を選択した場合、**検証資産の構築** ページが開きます。

- a) **新しい検証の作成** をクリックすると、**イメージ検証** UI が開きます。この UI で、新しい検証資産を作成できます。
- b) **既存の検証の挿入** をクリックすると **検証の参照** ダイアログ ボックスが開きます。このボックスを使用して、テストで実行する検証資産を選択できます。
- c) 省略可能：**検証の失敗時にスクリーンショットをキャプチャする** チェック ボックスをオンにすると、検証の失敗時にスクリーンショットが結果ファイルに追加されます。
- d) **次へ** をクリックします。

12 ステップ シーケンスの実行にかかる時間 を選択した場合、**タイミングに基づく条件の定義** ページが開きます。一連のテスト ステップと、それを再生する時間の範囲を指定できます。

- a) 検証する一連のステップの最初のステップを指定するには、**ステップの開始** の右側にあるリンクをクリックします。
- b) 検証する一連のステップの最後のステップを指定するには、**ステップの終了** の右側にあるリンクをクリックします。
- c) **許容時間範囲** の 2 つのフィールドの値を定義して、一連のステップの再生にかかる時間の範囲を設定します。

たとえば、時間の範囲を 10 から 30 秒に設定した場合、指定した一連のステップの実際の再生時間が 10 秒未満または 30 秒を超える場合は、ビジュアル テストの結果に失敗のメッセージが送信されます。一方、実際の再生時間が 10 から 30 秒の場合は、ビジュアル テストの結果に合格のメッセージが送信されます。

d) **タイマー番号の使用** リストからタイマーを選択します。

複数の条件の時間を計るときに、各タイマーに一意の番号を割り当ててタイマーを区別できます。タイマーを使用して検証ロジックを作成すると、**テスト ステップ** ウィンドウに 2 つのテスト ステップが表示されます。最初のテスト ステップには **タイマー番号 X の開始** というラベルが付けられ (X は特定のタイマー番号)、検証する一連のステップの最初のテスト ステップの直前に表示されます。2 番目のテスト ステップには **タイマー番号 X の停止** というラベルが付けられ、検証する一連のステップの最後のステップの直後に表示されます。



ヒント：検証するステップの順序を変更する場合は、**テスト ステップ** ウィンドウで、いずれかのタイマーのテスト ステップを切り取り、目的の位置に貼り付けます。

e) **次へ** をクリックします。

13 検証の構築 ページが開きます。ビジュアル テストの結果に表示する合格/失敗のメッセージを作成します。また、検証が失敗した場合に、結果および現在のユーザーの **開始画面** にフラグを送信することもできます。

- a) 省略可能：**If** 文の右側にある条件リンクをクリックして、**条件デザイナー** を開き、条件を編集します。
- b) **結果の詳細に以下の '合格テキスト' を表示する** フィールドに、再生時に検証が合格したときに結果に表示するテキストを入力します。
このテキストは、**詳細** タブの **テスト ステップ** ペインで、検証ステップの **結果の詳細** 列に表示されます。
- c) **そうでなければ、結果の詳細に以下の '失敗テキスト' を表示する** フィールドに、再生時に検証が失敗したときに結果に表示するテキストを入力します。
このテキストは、**失敗** タブ、**フラグ** タブ、および **詳細** タブの **テスト ステップ** ペインで、検証ステップの **結果の詳細** 列に表示されます。
- d) 省略可能：**検証に失敗した場合に、結果に 'フラグ' を作成する** チェック ボックスをオンにすると、検証が失敗したときにフラグが作成されます。

このフラグは、**失敗** タブ、**フラグ** タブ、および **詳細** タブの **テスト ステップ** ペインで、検証ステップの **結果の詳細** 列に表示されます。デフォルトでは、このフラグは現在のユーザーに自動的に割り

当てられ、現在のユーザーの **開始画面** の **フラグ** ペインに表示されます。このフラグを別の人に割り当てる場合は、再生後に結果から再割り当てする必要があります。

- e) 省略可能：**以下のフラグ説明を表示する** フィールドに検証が失敗したときのフラグの説明を入力します。
- f) 省略可能：**検証の失敗時にスクリーンショットをキャプチャする** チェックボックスをオンにすると、検証の失敗時にスクリーンショットが結果ファイルに追加されます。

14次へ をクリックします。**要約** ページに、定義したテスト ロジックが表示されます。このページの最下部にあるチェックボックスをオフにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェックボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「**全般オプションの変更**」を参照してください。

15完了 をクリックして、ビジュアルテストのテストステップにテスト ロジックを挿入します。テスト ロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テスト ロジックがビジュアルテストに追加されます。

検証ロジック テスト ステップの編集

1. 編集するテストステップが含まれているビジュアルテストを開きます。
2. ビジュアルテストの **テスト ステップ** ウィンドウで、編集する検証ロジック テストステップを選択します。選択したテストステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティ カテゴリごとに表示されます。
3. 使用可能な条件プロパティ、検証結果の説明プロパティ、フラグの設定プロパティ、またはその他のプロパティを編集します。

エラー処理ロジック

エラー処理ロジックを設定すると、再生エラーをデバッグする必要がなくなります。エラー処理を設定するには、ビジュアルテストでエラーの検出ステップとその処理方法を設定します。エラー処理を設定することによって、ビジュアルテストを最後まで再生でき、再生中に発生したエラーの情報を得ることができます。

エラー処理ロジックは、以下の要素で構成されます。

```
If <エラー=TRUE>  
  __<実行する操作>
```

たとえば、エラー処理で次のステップから再開するように設定していないかぎり、エラーが発生すると、失敗メッセージとフラグが送信されます。

エラー時に '再開して次に進む'

エラー処理ロジックを作成するには、**テスト ロジック デザイナ** ウィザードまたはロジック ツールボックスを使用するか、**挿入 > エラー処理** を選択します。

ビジュアルテストへのエラー処理ロジックの追加

このトピックでは、**テスト ロジック デザイナ** を使用して、ビジュアルテストにエラー処理ロジックを追加する方法について説明します。

1. ビジュアルテストを開きます。
2. エラー処理ロジックを追加する場所のテストステップを右クリックします。
3. **挿入 > テスト ロジック > エラー処理** を選択します。**テスト ロジック デザイナ** が開きます。
4. **次へ** をクリックします。**ロジックの種類を選択** ページが開きます。
5. **次へ** をクリックします。**エラー処理の構築** ページが開きます。
6. エラー処理ロジックを指定して、エラーが発生した場合に使用するには、リストから次の項目のいずれかを選択します。

デフォルトの再生エラー ダイアログボックスを表示します **再生エラー** ダイアログボックスを表示し、そこでそのエラーに対して実行する操作を決定します。**再生エラー** ダイアログボックスを使用してデバッグモードに入り、ビジュアルテストでエラーを診断します。

現在のビジュアルテストの再生を終了します エラーが発生したときに、ビジュアルテストの再生を終了します。ビジュアルテストのその他のステップは再生されません。再生が終了すると、**再生完了** ダイアログボックスが表示され、そこで再生の完了時に実行する操作を決定します。

エラーが発生したステップの再生を再試行します **資産** プロパティに指定されているビジュアルテストを再生してから、エラーが発生したステップの実行を試行します。**資産** プロパティにビジュアルテストが指定されていない場合、単にエラーが発生したステップの再実行を試行します。**再試行の回数** プロパティには、再生を試行する回数を指定できます。デフォルト値は、5です。指定した再試行の回数に到達した時点でステップ実行に失敗すると、再生エラーが生成されます。

次のステップで再生を再開します **資産** プロパティに指定されているビジュアルテストを再生してから、エラーが発生したステップの直後のステップの実行を試みます。エラーの原因となったステップは、再実行されません。

ラベルへ移動します **資産** プロパティに指定されているビジュアルテストを再生し、**ラベル** プロパティに指定されている事前定義のラベルステップに移動して、再生を続行します。**移動** を使用すると、ビジュアルテストで、再生中に発生したエラーと同様のエラーが発生する可能性のあるステップをスキップして、テストの別のセクションに移動できます。

他のビジュアルテストを最初に再生します 指定した別のビジュアルテストが再生されたあと、次に実行される操作をユーザーが指定します。別のテスト実行のあとに実行される操作を決定するには、テストの終了、エラーが発生したステップを再試行、ラベルへ移動、またはエラーが発生した元のテストの再開を選択します。

7. **次へ** をクリックします。**要約** ページに、定義したテストロジックが表示されます。このページの最下部にあるチェックボックスをオフにすると、**テストロジックデザイナー** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェックボックスのオン/オフを切り替えると、**テストロジックデザイナーの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

8. **完了** をクリックして、ビジュアルテストのテストステップにテストロジックを挿入します。テストロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テストロジックがビジュアルテストに追加されます。

エラー処理ロジックテストステップの編集

1. 編集するテストステップが含まれているビジュアルテストを開きます。
2. ビジュアルテストの **テストステップ** ペインで、編集するエラー処理ロジックテストステップを選択します。選択したテストステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティカテゴリごとに表示されます。
3. 必要に応じて、プロパティを編集します。

テストロジックデザイナー

テストロジックデザイナー ウィザードでは、テストロジックを作成してビジュアルテストに挿入できます。このウィザードにアクセスするには、ビジュアルテストを開いているときに、**挿入 > テストロジック** メニューを選択します。また、ロジックツールボックスからもアクセスできます。その場合、さまざまなタイプのテストロジックについて把握し、目的のテストロジックを作成するために必要なウィザードの該当ページにすばやくアクセスできます。

以下のタイプのテストロジックを挿入するには、**テストロジックデザイナー** ウィザードを使用します。

- 判断ロジック

- 繰り返しロジック
- 検証ロジック
- エラー処理ロジック

テスト ロジック デザイナ ウィザードの最初のページは **ようこそ** ページで、ここにはウィザードのロジックステップのリスト、および設計する特定のタイプのテスト ロジックが表示されます。

このウィザードおよびそれ以外のすべての Silk Test Workbench ウィザードを今後使用するとき、**ようこそ** ページをスキップするには、このページの最下部にあるチェック ボックスをオンにします。このチェック ボックスをオンにすると、**全般** オプションの **ウィザードの [ようこそ] 画面の表示** オプションが更新されます。

ウィザードのサイズは、任意のページの右下隅をドラッグして変更できます。または、**ロジック ステップ** ペインのタイトルの横の矢印ボタンをクリックするか、**全般** オプションの **テスト ロジック デザイナのステップの表示** ペイン オプションを設定することでも、ウィザードの **ロジック ステップ** ペインを表示/非表示にできます。

テスト ロジック デザイナ ウィザードの最後のページは、**要約** ページです。このページの最下部にあるチェック ボックスをオンにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェック ボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

ステップの選択

ステップの選択 ダイアログ ボックスを使用して、ビジュアル テストの単一のステップまたは一連のステップを選択できます。

1. ステップを選択します。
2. 省略可能：一連のステップを選択するには、Shift キーを押しながら最後のステップを選択します。
3. **OK** をクリックします。

ロジック ツールボックス

ロジック ツールボックスは、テスト ロジックをビジュアル テストにすばやく挿入できるナビゲーション支援機能です。ロジック ツールボックスでは、さまざまなタイプのテスト ロジックについて把握し、目的のテスト ロジックを選択してビジュアル テストに挿入できます。ロジック ツールボックスでテスト ロジックのタイプを選択すると、Silk Test Workbench によって **テスト ロジック デザイナ** ウィザードの該当ページが開き、テスト ロジックを作成して挿入できます。**このビジュアル テストの再生中に出現/消滅するオブジェクトを待機** を使用する場合は、ウィザードを使用せずに、**プロパティ** ペインでオブジェクトプロパティを指定します。

ロジック ツールボックスでは、一般的なタイプのテスト ロジックが以下のカテゴリに整理されます。

- **画面内容ロジック**: テスト アプリケーションの画面コントロールに基づいて一連のステップを実行する判断 (If) ロジックまたは繰り返しロジック。
- **ユーザー定義のフォローアップ ロジック**: ユーザー定義の合格/失敗メッセージとオプションのフラグをビジュアル テストの結果に送信する検証ロジック。また、指定したユーザーの開始画面にフラグを送信して、特定のテスト ステップまたはビジュアル テストの結果を通知することもできます。
- **エラー処理と同期**: ステップを実行する前に出現/消滅するオブジェクトを待機する同期ロジック。再生中にエラーが発生した場合、ユーザー定義の操作を実行するエラー処理ロジック。

ロジック ツールボックスの各カテゴリでは、各ロジック タイプの説明を参照し、目的のロジック タイプをダブルクリックして、ビジュアル テストにすばやく挿入できます。詳細については、「ロジック ツールボックスを使用したテスト ロジックの挿入」を参照してください。

ロジック ツールボックスを使用したテスト ロジックの挿入

ロジック ツールボックスを使用して、ビジュアル テストにテスト ロジックを挿入できます。

1. ビジュアル テストを開きます。

デフォルトでは、**ロジック ツールボックス** タブは **テスト ステップ** ペインの右側に表示されます。デフォルトでこのタブが表示されない場合は、**表示 > ロジック ツールボックス** をクリックします。

2. ロジック ツールボックス タブをクリックして、ロジック ツールボックスを開きます。



ヒント: ロジック ツールボックスのサイズを変更するには、ポインタを端に合わせて、目的の幅になるまでツールボックスをドラッグします。

3. ロジック カテゴリをクリックして、ビジュアル テストに挿入できる各カテゴリのさまざまなロジック タイプを表示します。

ロジック タイプの説明を表示するには、目的のロジック タイプをクリックします。説明は、ロジック ツールボックスの下部に表示されます。

4. ロジック タイプをダブルクリックすると、**テスト ロジック デザイナ** ウィザードの該当ページが自動的に開きます。

または、ロジック タイプを **テスト ステップ** ペインの適切な位置にドラッグ&ドロップすることもできます。

5. **テスト ロジック デザイナ** ウィザードを使用して、ビジュアル テストにテスト ロジックを作成および挿入します。

記録中のテスト ロジックの挿入

ビジュアル テストの記録中にテスト ロジックを挿入できます。

1. 記録を開始します。

2. テスト ロジックを追加するポイントに到達したとき、以下の手順のいずれかを実行します。

- Alt+F9 を押します。

このショートカット キーの組み合わせを使用すると、記録が一時的に停止され、現在強調表示されているオブジェクトが検証のために選択されます。**テスト ロジック デザイナ** ウィザードが開き、選択されたオブジェクト セットが検証対象のコントロールとなります。

- 記録中** ダイアログ ボックスで、**検証の追加** をクリックするか、または Ctrl+Alt を押します。

このオプションを実行すると、記録が一時的に停止され、**テスト ロジック デザイナ** ウィザードが表示されます。ウィザードが開いたあと、検証するオブジェクトを選択します。

3. ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。記録が完了すると、新しいテスト ロジックが **ビジュアル ナビゲータ** の **テスト ステップ** ペインに表示されます。

既存のビジュアル テストへのテスト ロジックの挿入

作成したビジュアル テストにテスト ロジックを挿入することができます。

1. ビジュアル テストを開きます。

2. テスト ロジックに含める 1 つのテスト ステップまたは一連のテスト ステップを選択します。

3. **挿入 > テスト ロジック** を選択し、適切なロジックの種類を選択します。

Ctrl+Alt を押して、ビジュアル テストに検証ロジックを追加することもできます。

テスト ロジック デザイナ ウィザードが開きます。

4. ウィザードに従って処理を行い、**完了** をクリックします。

テスト ステップの説明が長すぎて **テスト ステップ** ペインに表示できない場合は、ポインタをテスト ステップに合わせて、ツール ヒントとして説明全体が表示されます。

テスト ロジック テスト ステップがビジュアル テストに表示されます。

条件デザイナー

条件デザイナー ダイアログ ボックスでは、判断、繰り返し、および検証テスト ロジックの結果を導く条件をすばやく定義できます。**条件デザイナー** には、**テスト ロジック デザイナ** ウィザードまたはロジック

ツールボックスからアクセスします。特定のタイプのテスト ロジックを作成したあと、**プロパティ** ペインから **条件デザイナー** にアクセスして、条件を編集することもできます。

条件デザイナー を使用して定義される条件は、以下の要素で構成されます。

```
<値 1> <比較演算子> <値 2> = <戻り値>
```

例：

```
5 より大きい 4 = TRUE
```

条件の各要素は、以下のとおりです。

- 値 1 と値 2：条件の値は、互いに比較される入力パラメータとして使用されます。条件の値には、コントロールのプロパティ、変数、アクティブ データ資産、またはリテラル値を指定できます。
- 比較演算子：比較演算子は、条件の比較ロジックと、条件内の 2 つの値の相互関係を定義します。比較演算子を以下に示します。
 - 等しい (==)
 - 等しくない (!=)
 - より大きい (>)
 - より小さい (<)
 - 以上 (>=)
 - 以下 (<=)
 - 含む：値 1 のテキストに 値 2 のテキストが含まれる場合は True。それ以外の場合は False。
 - 含まない：値 1 のテキストに 値 2 のテキストが含まれない場合は True。それ以外の場合は False。
- 戻り値：比較演算子を使用して 2 つの値を比較したとき、評価された条件の戻り値は TRUE または FALSE のいずれかとなります。

各テスト ロジック タイプで、条件の戻り値を使用して実行される操作が決定されます。判断ロジックと繰り返しロジックの場合は、一連のテスト ステップが実行されます。検証ロジックの場合は、ビジュアル テストの結果に合格または失敗のメッセージが送信されるか、オプションで、指定されたテスト ステップにフラグが立てられます。

判断テスト ロジックの例

```
If counterVar 等しい 12  
ステップ 2 からステップ 4 まで実行  
End If
```

繰り返しテスト ロジックの例

```
counterVar 等しい 12 の間繰り返します  
ステップ 2 から  
ステップ 4 まで実行  
繰り返しの終了
```

検証テスト ロジックの例

```
If counterVar 等しい 12  
合格メッセージを送信  
else  
失敗メッセージを送信  
End If
```

また、論理演算子 AND または OR を使用して複合条件を作成できます。たとえば、以下のような複合の判断ロジック条件を作成できます。

```
If counterVar 等しい 12  
AND  
textVar 等しい "Admin"
```

ステップ 2 からステップ 4 まで実行
End If

条件デザイナーを使用した条件の定義

条件デザイナー ダイアログ ボックスを使用して、詳細なテスト ロジックの条件を定義または編集できます。

1. 条件デザイナー には **テスト ロジック デザイナー** ウィザードからアクセスします。


または、If テスト ステップを挿入し、**プロパティ** ウィンドウで **条件** プロパティを選択して、**条件デザイナー** をクリックします。

条件デザイナー が開きます。

2. 条件の定義 セクションの **値 1** テキスト ボックスで、最初の条件の値を指定します。この値には、コントロールのプロパティ、任意のタイプの変数、アクティブ データ資産、リテラル値、またはコントロールの存在確認を指定できます。

テキスト ボックスに直接値を入力するか、... をクリックして **条件の選択** ダイアログ ボックスにアクセスします。

 **ヒント:** 値を選択すると、その値の型が Silk Test Workbench によって **値 1** テキスト ボックスに表示されます。

 **注:** データ型は、Double から Long、テキスト、ブール値など、他の型に変換できます。ただし、このような変換によってデータがどのような影響を受けるかを確認することが重要です。データ型の変換の詳細については、「式デザイナーの関数の使用」を参照してください。

3. 演算子 リストから比較演算子を選択します。

比較演算子では、条件の比較ロジックと条件内の 2 つの値の相互関係を定義します。以下の演算子を利用できます。


- 等しい (==)
- 等しくない (!=)
- より大きい (>)
- より小さい (<)
- 以上 (>=)
- 以下 (<=)
- 含む : **値 1** のテキストに **値 2** のテキストが含まれる場合は True。それ以外の場合は False。
- 含まない : **値 1** のテキストに **値 2** のテキストが含まれない場合は True。それ以外の場合は False。

4. ステップ 2 で最初の値を指定した手順と同じ手順に従って、**値 2** テキスト ボックスで 2 番目の条件値を指定します。

5. 評価結果 リストから、条件の評価に対するデータ型を選択します。

使用可能なデータ型は、以下のとおりです。

テキスト 変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

 **注:** テキスト データ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

テキスト (大文字と小文字を区別しない) 評価中の文字列比較において、文字列の個々の文字の大文字小文字が一致しないことを除き、テキストは同じです。

数値 (Long) 変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。

数値 (Double) 変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。

ブール値 (True/False) 変数の値は True または False です。

数値 (Long Long) 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙 この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

6. 定義した条件を **条件** テーブルに追加するには、**追加** をクリックします。

単一の条件ステートメントを作成する場合、**ロジック** リスト (AND、OR) は無視してください。**ロジック** ボックスは、複合条件を作成する場合にのみ使用します。

7. 省略可能 : **上へ移動** または **下へ移動** をクリックすると、**条件** テーブルで選択した条件を 1 行上に、または 1 行下に移動できます。

論理演算子 (AND、OR) を含む条件を **条件** テーブルの先頭行に移動した場合、論理演算子は条件と一緒に移動されません。条件ロジックの構造を維持するために、演算子は 2 行めに残ります。

8. オプション: 複合条件を作成する場合は、前のステップに従って追加条件を作成し、**ロジック** リストから結合ロジックのタイプを選択します。**追加** をクリックして、作成済みの条件のあとに条件を追加します。

9. 省略可能 : 選択されている条件を **条件** テーブルから削除するには、**削除** を使用します。

10 省略可能 : **プレビュー** をクリックすると、定義済みの条件が読みやすいテキスト形式で表示されます。

条件デザイナを使用した条件の編集

1. ビジュアル テストのテスト ステップの **プロパティ** ペインで **条件** プロパティを選択し、**条件デザイナ** をクリックします。**条件デザイナ** が開きます。

2. 既存の条件の値を編集するには、編集する条件を **条件** テーブルで選択します。

ロジック、**演算子**、および **評価結果** 列にリスト矢印が表示されます。いずれかの列の値を編集するには、編集する項目をリストから選択します。**値 1** および **値 2** の各列で、選択した条件の列のセルをダブルクリックして、**条件の選択** ダイアログ ボックスを開きます。このダイアログ ボックスで、いずれかの値を編集できます。

3. 複合条件の順序を編集するには、順序を編集する条件を選択してから **上へ移動** または **下へ移動** をクリックします。

論理演算子 (AND、OR) を含む条件も **条件** テーブルの先頭行に移動できます。ただし、論理演算子は条件と一緒に移動されません。条件ロジックの構造を維持するために、演算子は 2 行めに残ります。

4. **OK** をクリックします。

条件の値の設定

ビジュアル テストに条件ロジックを追加するとき、**条件デザイナ** で条件に対する値を選択したり、リテラル値を作成するには、**条件の選択** ダイアログ ボックスを使用します。

選択 (リテラル) : このオプションを選択すると、以下のリテラル オプションが表示されます。

- **リテラル データ** : テキスト文字列、数値、またはブール (true または false) 値のいずれかを入力し、**OK** をクリックします。

1. **条件デザイナ**の**条件の定義**で、変更する値テキストボックスの右側にある...をクリックします。
条件の選択ダイアログボックスが開きます。
2. 値の種類を選択します。
 - コントロールのプロパティを選択する場合は、**プロパティ**をクリックします。
 - 変数を選択する場合は、**変数**をクリックします。
 - アクティブ データ資産を選択する場合は、**アクティブ データ**をクリックします。
 - 新しいリテラル値（文字列、数値、ブール値）を作成する場合は、**リテラル**をクリックします。
 - 指定したコントロールが存在するかどうかを検証する場合は、**存在確認**をクリックします。
3. **プロパティ**を選択した場合は、以下の操作を実行します。
 - a) **コントロールの識別**メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト アプリケーションから可視のリスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接リスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：リスト ボックスまたはコンボ ボックス コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
 - b) コントロールの識別後、**プロパティ** リストから目的のプロパティを選択します。
4. **変数**を選択した場合は、以下の操作を実行します。
 - a) **変数の種類** リストから変数の種類を選択するか、**新規作成** をクリックして新しい変数を作成します。利用可能な変数の種類は次のとおりです。
 - **ローカル**：ローカル変数を使用する場合はこれを選択します。ローカル変数は、単一のビジュアル テスト内で定義され使用される変数で、通常、他の種類の変数の値（グローバル変数や入力変数など）を格納します。
 - **入力パラメータ**：ビジュアル テストに渡されるパラメータを使用する場合はこれを選択します。
 - **出力パラメータ**：ビジュアル テストから返されるパラメータを使用する場合はこれを選択します。
 - **予約**：ビジュアル テストの再生や結果に関する情報を保存する、あらかじめ定義された変数を使用する場合はこれを選択します。
 - b) **変数** リストから目的の変数を選択します。
5. **アクティブ データ**を選択した場合は、以下の操作を実行します。
 - a) **アクティブ データ資産** テキスト ボックスの右側にある **選択** メニューを展開します。
 - b) 新しいアクティブ データ資産を作成するには、**アクティブ データの新規作成** をクリックします。詳細については、「アクティブ データ テスト用の新しいデータ ファイルの作成」を参照してください。
 - c) 既存のアクティブ データ資産に関連付けるには、**既存のアクティブ データとの関連付け** をクリックします。詳細については、「アクティブ データ ファイルのデータのビジュアル テスト データへのマップ」を参照してください。
 - d) アクティブ データ資産を指定した後、**列** リストから使用する列を選択します。
6. **リテラル**を選択した場合は、**リテラル データ** テキスト ボックスに文字列、数値、またはブール値を入力します。
7. **存在確認**を選択した場合は、以下の操作を実行します。
 - a) **コントロールの識別**メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。

- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。
- c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。
- d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。
- デフォルトのタイムアウトは 0 で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。

8. **OK** をクリックします。

式デザイナー

式デザイナー を使用すると、ビジュアル テストの機能を拡張する式をすばやく作成できます。式は、ローカル変数、予約された変数、アクティブ データ資産、演算子、および関数（ローカル変数またはアクティブ データ資産に文字列または数値を返す）の組み合わせです。

たとえば、10 個の異なる注文番号を必要とするアプリケーションをテストする場合、変数の値を増分し、一意の注文番号を返す式を作成します。

```
OrderNumVar=OrderNumVar+1
```

次に、この式を繰り返しループで使用して、ループを複数回実行します。ループが繰り返されるたびに、注文番号は 1 ずつ増分し、ローカル変数 OrderNumVar に格納されている一意の注文番号が返されます。

また、式にアクティブ データ フィールドを含めることもできます。その場合、式にアクティブ データ列を指定し、その列のすべての値を使用して一連のステップを繰り返す繰り返しロジックを設定します。たとえば、アクティブ データ列の値のうち、指定した値（以下の ADDRESS 列に示されている値など）だけを返す式を作成できます。

ADDRESS
Anytown, USA 12345
Anycity, USA 23456
Anyplace, USA 34567
Anywhere, USA 45678

ADDRESS 列の郵便番号のみを返すには、「ADDRESS」というアクティブ データ列、および文字列の右側から指定した数の文字を返す文字列関数 Right を含む式を作成します。

```
localVar = Right (ActiveDataAssetName("ADDRESS"),5)
```

この式では、文字列の右側から 5 文字が返され、戻り値が localVar に格納されます（この例では、「12345」）。アクティブ データ ファイルの行に対して反復処理を行い、上記の列の各郵便番号を返すには、列の各値を使用して式ステップを繰り返す繰り返しロジックを設定します。

式デザイナー では、文字列関数 Right だけでなく、その他の関数と演算も多数サポートされています。

式デザイナーを使用した式の定義

式デザイナー ダイアログ ボックスを使用して、ビジュアル テストの機能を拡張する式を作成または編集できます。式は、ローカル変数、予約された変数、アクティブ データ資産、演算子、および文字列、数値、またはオブジェクトを返す関数を組み合わせて作成します。詳細については、「式デザイナー」を参照してください。


1. ビジュアル テストを開きます。

2. **テストステップ** ペインで、式を挿入する場所の直前のテストステップを選択します。

3. メニューバーから、**挿入 > 式** を選択します。**式デザイナー** が開きます。

4. **出力** リストで、式の結果を格納するローカル変数またはアクティブ データ資産を選択します。

ビジュアル テストに関連付けられたローカル変数とアクティブ データ資産のみがリストされます。新しく作成した変数とアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。

 **注:** 式を **プロパティ** ペインのプロパティの値として作成する場合は、**出力** リストは表示されません。

5. 省略可能：既存のアクティブ データ資産を選択したり、新しい変数またはアクティブ データ資産を作成したりするには、**出力** リストの隣にある **選択** をクリックします。

以下のいずれか 1 つを選択できます。

- **ローカル変数の新規作成** を選択すると、**ローカル変数の追加** ダイアログ ボックスを使用して、ビジュアル テスト用のローカル変数を作成できます。作成したローカル変数はすべて、**出力** リストに表示され、式の結果を格納するために使用できます。
- **アクティブ データの新規作成** を選択すると、新しいアクティブ データ資産を作成して、ビジュアル テストに関連付けることができます。
- **既存のアクティブ データとの関連付け** を選択すると、**アクティブ データの参照** ダイアログ ボックスを使用して、既存のアクティブ データ資産をビジュアル テストに関連付けることができます。

出力 リストには、関連付けられたアクティブ データ資産の一部であるアクティブ データ ファイルの列名も表示され、式の結果を格納するために使用できます。

6. **エディタ** ウィンドウを使用して、式を作成および編集します。

式には、リテラル値、変数、アクティブ データ列、演算子、および関数を含めることができます。

- a) 式に変数をすばやく追加するには、**変数の選択 (ローカルまたは予約)** リストから目的の変数を選択し、**追加** ボタンをクリックします。
- b) 省略可能：**新規作成** をクリックすると、**ローカル変数の追加** ダイアログ ボックスが開きます。このダイアログ ボックスでは、新しい変数を作成して、それをビジュアル テストに自動的に関連付けることができます。
- c) アクティブ データ フィールドを追加するには、**アクティブ データ フィールドの選択** リストから目的のアクティブ データ フィールドを選択し、**として追加** ボタンをクリックしてデータ型を選択します。

エディタ ウィンドウでのアクティブ データ フィールドの構文は、以下のとおりです。

```
[プロジェクト名].[アクティブ データ名].データ型("列名")
```

サポートされているデータ型は、以下のとおりです。

テキスト 選択された **アクティブ データ** 列のデータは、式内でテキスト データとして使用されます (String データ型)。

数値 (Long) 選択された **アクティブ データ** 列のデータは、式内で数値データとして使用されません (Long データ型)。

数値 (Double) 選択された **アクティブ データ** 列のデータは、式内で数値データとして使用されません (Double データ型)。

ブール値 選択された **アクティブ データ** 列のデータは、式内で True または False として評価されます。(Boolean データ型)。

数値 (Long Long) 選択された **アクティブ データ** 列のデータは、式内で 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) として使用されます。

列挙 選択された **アクティブ データ** 列のデータは、式内で一連の値として使用され、1 から n まで順番に番号が付けられます。このデータ型は、パラメーターとして列挙が

必要な一部のユーザー インターフェイスで使用されます。たとえば、SAP SendVKey は送信するキーとして整数の列挙値を想定しています。限られた個数の一意の値のみを式に保持する場合は、列挙型を宣言します。ビジュアル テストで使用するためのカスタム列挙を作成することはできません。


列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

- d) 新しいアクティブ データ資産を作成したり、既存のアクティブ データ資産を選択したりするには、**選択** をクリックして目的のオプションを選択します。新しいアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。
7. **テスト** をクリックして、式の有効性を評価します。テストの結果が **テスト結果** ウィンドウに表示されます。
 8. 省略可能 : **エディタ** ウィンドウの内容全体を消去するには、**クリア** をクリックします。
 9. **OK** をクリックします。

式デザイナーを使用した式の編集

式デザイナー ダイアログ ボックスを使用して、ビジュアル テストの機能を拡張する式を作成または編集できます。式は、ローカル変数、予約された変数、アクティブ データ資産、演算子、および文字列、数値、またはオブジェクトを返す関数を組み合わせて作成します。詳細については、「式デザイナー」を参照してください。

1. 変更する式が含まれているビジュアル テストを開きます。
2. **テスト ステップ** ペインで式ステップを選択します。
3. **プロパティ** ペインで **式** プロパティを選択します。
4. **式デザイナー** をクリックします。**式デザイナー** が開きます。
5. **出力** リストで、式の結果を格納するローカル変数またはアクティブ データ資産を選択します。
ビジュアル テストに関連付けられたローカル変数とアクティブ データ資産のみがリストされます。新しく作成した変数とアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。

 **注:** 式を **プロパティ** ペインのプロパティの値として作成する場合は、**出力** リストは表示されません。

6. 省略可能 : 既存のアクティブ データ資産を選択したり、新しい変数またはアクティブ データ資産を作成したりするには、**出力** リストの隣にある **選択** をクリックします。
以下のいずれか 1 つを選択できます。

- **ローカル変数の新規作成** を選択すると、**ローカル変数の追加** ダイアログ ボックスを使用して、ビジュアル テスト用のローカル変数を作成できます。作成したローカル変数はすべて、**出力** リストに表示され、式の結果を格納するために使用できます。
- **アクティブ データの新規作成** を選択すると、新しいアクティブ データ資産を作成して、ビジュアル テストに関連付けることができます。
- **既存のアクティブ データとの関連付け** を選択すると、**アクティブ データの参照** ダイアログ ボックスを使用して、既存のアクティブ データ資産をビジュアル テストに関連付けることができます。

出力 リストには、関連付けられたアクティブ データ資産の一部であるアクティブ データ ファイルの列名も表示され、式の結果を格納するために使用できます。

7. **エディタ** ウィンドウを使用して、式を作成および編集します。
式には、リテラル値、変数、アクティブ データ列、演算子、および関数を含めることができます。
 - a) 式に変数をすばやく追加するには、**変数の選択 (ローカルまたは予約)** リストから目的の変数を選択し、**追加** ボタンをクリックします。

- b) 省略可能 : **新規作成** をクリックすると、**ローカル変数の追加** ダイアログ ボックスが開きます。このダイアログ ボックスでは、新しい変数を作成して、それをビジュアル テストに自動的に関連付けることができます。
- c) アクティブ データ フィールドを追加するには、**アクティブ データ フィールドの選択** リストから目的のアクティブ データ フィールドを選択し、**として追加** ボタンをクリックしてデータ型を選択します。

エディタ ウィンドウでのアクティブ データ フィールドの構文は、以下のとおりです。

[プロジェクト名].[アクティブ データ名].データ型("列名")

サポートされているデータ型は、以下のとおりです。

テキスト 選択された **アクティブ データ** 列のデータは、式内でテキスト データとして使用されます (String データ型)。

数値 (Long) 選択された **アクティブ データ** 列のデータは、式内で数値データとして使用されず (Long データ型)。

数値 (Double) 選択された **アクティブ データ** 列のデータは、式内で数値データとして使用されず (Double データ型)。

ブール値 選択された **アクティブ データ** 列のデータは、式内で True または False として評価されます。(Boolean データ型)。

数値 (Long Long) 選択された **アクティブ データ** 列のデータは、式内で 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) として使用されます。

列挙 選択された **アクティブ データ** 列のデータは、式内で一連の値として使用され、1 から n まで順番に番号が付けられます。このデータ型は、パラメーターとして列挙が必要な一部のユーザー インターフェイスで使用されます。たとえば、SAP SendVKey は送信するキーとして整数の列挙値を想定しています。限られた個数の一意の値のみを式に保持する場合は、列挙型を宣言します。ビジュアル テストで使用するためのカスタム列挙を作成することはできません。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

- d) 新しいアクティブ データ資産を作成したり、既存のアクティブ データ資産を選択したりするには、**選択** をクリックして目的のオプションを選択します。新しいアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。

8. **テスト** をクリックして、式の有効性を評価します。テストの結果が **テスト結果** ウィンドウに表示されます。

9. 省略可能 : **エディタ** ウィンドウの内容全体を消去するには、**クリア** をクリックします。

100K をクリックします。

式デザイナの演算子の使用

式デザイナ では、以下の演算子がサポートされています。

算術演算子

演算子	演算子名	構文	例
+	プラス演算子	+a	+1

演算子	演算子名	構文	例
-	マイナス演算子 (単項)	-a	-1
+	加算演算子	a + b	1 + 2 = 3
-	減算演算子	a - b	2 - 1 = 1
*	乗算演算子	a * b	2 * 2 = 4
/	除算演算子	a / b	4 / 2 = 2
%	剰余演算子	a % b	5%2=1 (余り)

比較演算子

演算子	演算子名	構文	例
!	ブール論理否定演算子 (単項)	!a	!a (a ではない)
&&	論理 AND	a && b	1 < 5 && 5 < 10 = true
	論理 OR	a b	1 < 5 5 > 10 = true
=	代入演算子	a = b	a = 3
==	Equal To (等値) 演算子	a == b	3 == 3
!=	Not Equal To (非等値) 演算子	a != b	4 != 3
>	Greater Than (より大きい) 演算子	a > b	4 > 3
>=	Greater Than Or Equal To (以上) 演算子	a >= b	4 >= 3
<	Less Than (より小さい) 演算子	a < b	3 < 4
<=	Less Than Or Equal To (以下) 演算子	a <= b	3 <= 4

ビット処理演算子

演算子	演算子名	構文	例
~	ビットごとの NOT (バイナリ数値のビットを反転することで、式で論理否定を実行)	a = ~ b	~1 = -2、ここで 00000001 = 1 11111110 = -2
&	ビットごとの AND (2つの式で論理積を実行)	a = b & c	1 & 1 = True 0 & 1 = False 1 & 0 = False 0 & 0 = False
	ビットごとの OR (2つの式で論理和を実行)	a = b c	1 1 = True 0 1 = True 1 0 = True

演算子	演算子名	構文	例
^	ビットごとのXOR (2つの式で排他的論理和を実行)	$a = b \wedge c$	$0 \mid 0 = \text{False}$
			$1 \wedge 1 = \text{False}$
			$0 \wedge 1 = \text{True}$
			$1 \wedge 0 = \text{True}$
			$0 \wedge 0 = \text{False}$

式デザイナーの関数の使用

式デザイナーでは、以下の関数がサポートされています。

数学関数


関数/説明	構文	例
Abs: 数値の絶対値を返します。	Abs(式)	Abs(-2) と Abs(2) は両方とも 2 を返します。
Round: 最も近い整数に丸めた数値を返します。	Round(式)	Round(1.3333) は 1 を返します。
Max: 2 つの数値を比較し、大きい方の数値を返します。	Max(数値 1, 数値 2)	Max(-123,12) は 12 を返します。
Min: 2 つの数値を比較し、小さい方の数値を返します。	Min(数値 1, 数値 2)	Min(-123,12) は -123 を返します。

文字列関数

関数/説明	構文	例
Contains: 文字列 2 が文字列 1 に含まれているかどうかを返します。	Contains(文字列 1, 文字列 2)	Contains("Hello World", "Hello") は true を返します。 Contains("Hello World", "Goodbye") は false を返します。
Find: 文字列内の特定の文字列の最初の出現位置を返します。	Find(文字列 1, 文字列 2)	Find("abcd","c") は 3 を返します。
FormatDate: 日付、日付形式、およびロケールに基づく日付文字列を返します。	FormatDate(日付, 日付形式, ロケール)	FormatDate("May 13, 2009", "MM/dd/yy", "en-US") は「05/13/09」を返します。
FormatTime: 時刻、時刻形式、およびロケールに基づく時刻文字列を返します。	FormatTime(時刻, 時刻形式, ロケール)	FormatTime("23:02:33", "hh:mm tt", "en-US") は「11:02 PM」を返します。
Left: 文字列の左側から指定した数の文字を返します。	Left(文字列, 長さ)	Left("abcd",2) は「ab」を返します。
Length: 文字列の文字数を返します。	Length(文字列)	Length("abcd") は 4 を返します。
MakeLower: テキスト文字列を小文字に変換します。	Makelower(文字列)	Makelower("ABCD") は「abcd」を返します。

関数/説明	構文	例
ToUpper : テキスト文字列を大文字に変換します。	ToUpper(文字列)	ToUpper("abcd") は「ABCD」を返します。
Mid : 文字列の中央から指定した数の文字を返します。	Mid(文字列, 開始位置, 長さ)	Mid("abcd",2,2) は「bc」を返します。
Replace : 指定した回数だけ、指定した部分文字列が別の部分文字列に置換されたテキスト文字列を返します。	Replace(文字列, 置換対象の部分文字列, 置換後の部分文字列)	Replace("abc def","bc","xy") は「axy def」を返します。
Right : 文字列の右側から指定した数の文字を返します。	Right(文字列, 長さ)	Right("abcd",2) は「cd」を返します。
Trim : 指定した文字列を、先頭と末尾のスペースを削除して返します。	Trim(文字列)	Trim(" abcd ") は「abcd」を返します。
TrimLeft : 指定した文字列を、先頭のスペースを削除して返します。	TrimLeft(文字列)	TrimLeft(" abcd ") は「abcd」を返します。
TrimRight : 指定した文字列を、末尾のスペースを削除して返します。	TrimRight(文字列)	TrimRight(" abcd ") は「abcd」を返します。

変換関数

関数/戻り値の型	構文	式の範囲
ToBool : Boolean	ToBool(式)	有効な文字列または数値の式
ToDouble : Double	ToDouble(式)	1.7E +/- 308
ToLong : Long	ToLong(式)	-2,147,483,648 から 2,147,483,647 (端数は丸められる)
ToString : Text	ToString(式)	ToString の戻り値は式の内容によって異なります。
ToLongLong : Integer	ToLongLong	-9,223,372,036,854,775,808 から 9,223,372,036,854,775,807
ToEnumeration : Integer	ToEnumeration	-9,223,372,036,854,775,808 から 9,223,372,036,854,775,807  注: Open Agent では 64 ビットのデータ型がサポートされないため、この式に推奨される範囲は -2,147,483,648 から 2,147,483,647 までです。

変換規則

データ型を変換する場合は、以下の規則に従ってください。

Double 型から Long 型への変換

小数点は切り捨てられます。Long 型の範囲外の数値はラップされます。たとえば、Double 型の数値 49.6 は 49 に丸められます (つまり、.6 は切り捨てられます)。Double 型の数値 -2147483649 は Long 型の上限 (+214748367) にラップされます。

String 型から Double 型または Long 型への変換

変換は、認識不可の最初の文字（英字など）で停止します。Long 型の場合、小数点は切り捨てられ、Long 型の範囲外の数値はラップされます。たとえば、「123.7 oranges」は、Double 型の「123.7」および Long 型の「123」に変換されます。

Boolean 型への変換

ゼロ以外の数値、テキスト「true」、または数値に変換されるすべてのテキストは、True に変換されます。ゼロ、テキスト「false」、または数値に変換できないすべてのテキストは、False に変換されます。

Boolean 型から Double 型または Long 型への変換

True = 1 および False = 0。

FormatDate 関数

説明

ユーザーが指定した日付、日付形式、およびロケールに基づいて文字列式を返す式デザイナの関数。

構文

```
local variable = FormatDate (Date, Format, Locale)
```

FormatDate 関数の構文のパラメータは以下のとおりです。

Date 指定されたロケールに有効な日付文字列または日付文字列が含まれる変数。現在のシステムの日付を指定するには、「Current」を使用します。日付文字列を指定するには、希望する日付文字列を引用符で囲んで入力します（例："January 5th, 2009"）。日付文字列が含まれる変数の場合は、引用符なしで変数名を使用します。

Format 指定したロケールに対して有効な以下の要素の組み合わせで構成される日付形式。

- 日**
- d : 日を数値で表し、数値が 1 桁の場合は先頭にゼロを付けません（「1」から「9」、
「10」から「31」）。
 - dd : 日を数値で表し、数値が 1 桁の場合は先頭にゼロを付けます（「01」から「09」、
「10」から「31」）。
 - ddd : 曜日を表す 3 文字の略語（英語の Mon、Tue、Wed など）。
 - dddd : 曜日を略さずに表します（英語の Monday、Tuesday、Wednesday など）。
- 月**
- M : 月を数値で表し、数値が 1 桁の場合は先頭にゼロを付けません（「1」から「9」、
「10」から「12」）。
 - MM : 月を数値で表し、数値が 1 桁の場合は先頭にゼロを付けます（「01」から
「09」、
「10」から「12」）。
 - MMM : 月を表す 3 文字の略語（英語の Jan、Feb、Oct など）。
 - MMMM : 月を略さずに表します。
- 年**
- y : 年の最後の 2 桁の数値で表し、10 年未満である場合は先頭にゼロを付けません
（「0」から「9」、
「10」から「99」）。
 - yy : 年の最後の 2 桁を数値で表し、10 未満である場合は先頭にゼロを付けます（「00」
から「09」、
「10」から「99」）。
 - yyy : 年の最後の 2 桁を数値で表し、10 未満である場合は先頭にゼロを付けます
（「00」から「09」、
「10」から「99」）。
 - yyyy : 年を略さずに 4 桁の数値で表します。

元号 gg : 元号。



注: Silk Test Workbench では、指定された日付とロケールに元号が存在するかどうかを検証されます。存在する場合は、その元号が Silk Test Workbench によってデフォルトのカレンダーに使用されます。存在しない場合は、同じロケールの別のカレンダーに元号が存在するかどうか Silk Test Workbench によって判断されます。別のカレンダーが存在する場合は、すべての文字列の形式が Silk Test Workbench によってそのカレンダーに合わせて設定されます。存在しない場合、文字列の形式は Silk Test Workbench によってデフォルトのカレンダーに合わせて設定され、元号形式 (gg) は無視されます。

Locale 一般的に、ロケールでは、ユーザーが自分のユーザー インターフェイスに表示するためのユーザーの言語や国、日付と時刻に使用する形式などの任意の特別なバリエーションを設定を定義します。

Locale パラメータにより、FormatDate 関数の出力で日付文字列が解釈され、表示される方法に対して、指定したロケール定義が適用されます。このパラメータの値は、Windows オペレーティングシステムのロケール名に基づきます。たとえば、フランス語 (カナダ) のロケール名は「fr-CA」です。サポートされているロケールの一覧については、Microsoft の『National Language Support Reference』の「Locale Identifier Constants and Strings」というトピックを参照してください。



注: 使用可能なロケールは、オペレーティング システムのバージョンと[地域のオプション]の設定に基づきます。

解説

すべてのパラメータが必須です。Format または Locale パラメータのデフォルト値を使用するには、式に「Default」と入力します。以下の例では、FormatDate 関数により、現在のシステムの日付が、オペレーティング システムの日付形式とロケールを使用して返されます。

```
FormatDate("Current", "Default", "Default")
```

以下の各例は、Format パラメータの要素を使用して FormatDate 関数の出力をカスタマイズする方法を示しています。

```
FormatDate("Jan 9, 2009", "dddd, MMM-dd-yyyy",  
"Default") returns "Friday, Jan-09-2009"
```

```
FormatDate("Jan 9, 2009", "d/M/yy", "Default")  
returns "9/1/09"
```

```
FormatDate("Jan 9, 2009", "yy", "Default")  
returns "09"
```

以下の例は英語のオペレーティング システムで実行するもので、Locale パラメータを使用して、フランス語のロケール日付をスペイン語のロケール日付に変換する方法を示しています。この処理は 2 段階で行われ、最初に、数値ではないフランス語の日付 (2009 janvier 9) を、フランス語の数値の日付 (2009-01-09) 形式に変換します。次に、数値の日付をスペイン語の日付 (2009-enero-09) に変換します。

```
FormatDate("2009 janvier 9", "yyyy-MM-dd",  
"fr-FR") returns "2009-01-09"
```

```
FormatDate(FrNumericDate, "yyyy-MMMM-dd", "es-ES")  
returns "2009-enero-09"
```

FormatTime 関数

説明

ユーザーが指定した時刻、時刻形式、およびロケールに基づいて文字列式を返す式デザイナーの関数。

構文

```
local variable = FormatTime(Time, Format, Locale)
```

FormatTime 関数の構文のパラメータは以下のとおりです。

Time 指定されたロケールに有効な時刻文字列または時刻文字列が含まれる変数。現在のシステムの時刻を指定するには、「Current」を使用します。時刻文字列を指定するには、希望する時刻文字列を引用符で囲んで入力します（例："01:12:23"）。時刻文字列が含まれる変数の場合は、引用符なしで変数名を使用します。

Format 指定したロケールに対して有効な以下の要素の組み合わせで構成される時刻形式。

時間

- h：時間を 12 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けません。
- H：時間を 24 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けません。
- hh：時間を 12 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けます。
- HH：時間を 24 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けます。

分

- m：分を表し、分が 1 桁の場合は先頭にゼロを付けません。
- mm：分を表し、分が 1 桁の場合は先頭にゼロを付けます。

秒

- s：秒を表し、秒が 1 桁の場合は先頭にゼロを付けません。
- ss：秒を表し、秒が 1 桁の場合は先頭にゼロを付けます。

時刻マーカー

- t：1 文字の時刻マーカー文字列（A または P など）。
- tt：午前と午後を複数文字の文字列で表します（AM、PM など）。

Locale 一般的に、Locale には、ユーザーが自分のユーザー インターフェイスに表示するためのユーザーの言語や国、日付と時刻に使用する形式などの、任意の特別なバリエーションを設定を定義します。

Locale パラメータにより、FormatTime 関数の出力で時刻文字列が解釈され、表示される方法に対して、指定したロケール定義が適用されます。このパラメータの値は、Windows オペレーティングシステムのロケール名に基づきます。たとえば、フランス語（カナダ）のロケール名は「fr-CA」です。サポートされているロケールの一覧については、Microsoft の『National Language Support Reference』の「Locale Identifier Constants and Strings」というトピックを参照してください。



注: 使用可能なロケールは、オペレーティング システムのバージョンと[地域のオプション]の設定に基づきます。

解説

すべてのパラメータが必須です。Format または Locale パラメータのデフォルト値を使用するには、式に「Default」と入力します。以下の例では、FormatTime 関数により、現在のシステムの時刻が、オペレーティングシステムの時刻形式とロケールを使用して返されます。

```
FormatTime ("Current","Default", "Default")
```

以下の各例は、Format パラメータの要素を使用して FormatTime 関数の出力をカスタマイズする方法を示しています。

```
FormatTime("9:05:08 PM", "hh:mm:ss tt", "en-US") returns "09:05:08 PM"
```

```
FormatTime("9:05:08 PM", "HH:mm:ss", "en-US") returns "21:05:08"
```

```
FormatTime("9:05:08 PM", "tt", "Default") returns "PM"
```

以下の例は、Locale パラメータを使用して、英語（米国）と英語（英国）の各ロケールの出力に影響を与える方法を示しています。

```
FormatTime("14:30", "Default", "en-US") returns "2:30:00 PM"
```

```
FormatTime("14:30", "Default", "en-GB") returns "14:30:00"
```

自動ステップの操作

このセクションでは、ビジュアルテストで自動ステップを使用する方法を説明するトピックを示します。

記録されない自動テストステップの作成

自動テストステップとは、アプリケーションをテストするために、テストアプリケーションのコントロールに対して操作を実行するステップのことです。たとえば、テストアプリケーションで項目の選択、テキストの入力、コントロールのクリックなどを行うステップが自動ステップです。

自動ステップは、通常は記録します。ビジュアルテストの記録中にテストアプリケーションのコントロールを操作すると、Silk Test Workbench によって、通常は個々の操作（クリックなど）が単独のステップとして記録されます。

ただし、記録を行わずに自動テストステップを作成して、ビジュアルテストにそのステップを挿入することもできます。記録を行わずに自動ステップを作成すると、テスト計画にわずかな変更が生じた場合に、テスト全体を記録し直すことなくその変更を吸収できます。自動テストステップを作成するには、**オブジェクトの識別** ダイアログ ボックスを使用してテストアプリケーションを操作するか、**画面プレビュー** でキャプチャされた画面イメージを操作します。**オブジェクトの識別** ダイアログ ボックスを使用すると、Silk Test Workbench によってテストアプリケーションのさまざまなコントロールがどのように認識されるかを確認できるため、アプリケーションのテストを最適化する方法を理解しやすくなります。

テストアプリケーションを使用したコントロールの自動ステップの作成

テストアプリケーションを操作して、テスト操作を実行する個々のステップを作成できます。個々のステップを作成するということは、ビジュアルテストを記録し直す必要がなくなるということです。必要に応じて、テストアプリケーションに対して追加の操作を作成するだけで済みます。

画面プレビュー 内にキャプチャされた画面を使用して、コントロールに対して追加操作を作成することもできます。

1. コントロールを識別するビジュアルテストを開きます。
2. コントロールが含まれるテストアプリケーションを開きます。
3. 挿入ポイントの直前のステップを選択します。
選択したステップで、コントロールに対して操作を実行します。
4. **挿入 > コントロール > テスト対象アプリケーションから選択** を選択します。**記録** ダイアログ ボックスが開きます。
5. テストアプリケーションで使用するコントロールを選択します。**操作の選択** ダイアログ ボックスが開きます。
6. コントロールに対する操作を指定します。選択したステップの下に自動ステップが表示されます。ステップのプロパティが**プロパティ** ペインに表示されます。選択した操作は、**操作** カテゴリに**メソッド** プロパティとして表示されます。ステップのプロパティを編集すると、操作の実行方法をさらに制御できます。

画面プレビューを使用したコントロールの自動ステップの作成

画面プレビューを使用してコントロールの自動ステップを作成する前に、オプション **コントロールのキャプチャ** がはいに設定されていることを確認してください。

テストアプリケーションを操作して、テスト操作を実行する個々のステップを作成できます。個々のステップを作成するということは、ビジュアルテストを記録し直す必要がなくなるということです。必要に応じて、**画面プレビュー** でキャプチャされているテストアプリケーションの画面に対して追加操作を作成します。

テストアプリケーションを使用して、コントロールに追加操作を作成することもできます。

1. コントロールの識別元となる、キャプチャされた画面が含まれるビジュアルテストを開きます。
2. **画面プレビュー** にテストアプリケーションの画面またはコントロールを表示するステップを選択します。これを行うには、[テストステップ] ペインでステップをクリックするか、ストーリーボードでサムネイルをクリックします。
画面プレビュー で個々のコントロールを見やすくするには、**画面プレビュー** の **操作 > ズーム** を選択し、ズームの割合を選択します。
3. **画面プレビュー** で目的のコントロールを表示し、**操作 > コントロールの挿入 > 画面プレビューから選択** を選択します。
4. **画面プレビュー** でコントロールを識別します。**操作の選択** ダイアログボックスが開きます。
5. コントロールに対する操作を指定します。選択したステップの下に自動ステップが表示されます。ステップのプロパティが**プロパティ** ペインに表示されます。選択した操作は、**操作** カテゴリに**メソッド** プロパティとして表示されます。ステップのプロパティを編集すると、操作の実行方法をさらに制御できます。
6. **OK** をクリックします。

オブジェクトの識別ダイアログを使用したコントロールの自動ステップの作成

テストアプリケーションを操作して、テスト操作を実行する個々のステップを作成できます。個々のステップを作成するということは、ビジュアルテストを記録し直す必要がなくなるということです。必要に応じて、テストアプリケーションに対して追加の操作を作成するだけで済みます。

オブジェクトの識別 ダイアログボックスを使用して、非表示コントロールに対する追加の操作を作成することもできます。

1. コントロールを識別するビジュアルテストを開きます。
2. コントロールが含まれるテストアプリケーションを開きます。
3. 挿入ポイントの直前のステップを選択します。
選択したステップで、コントロールに対して操作を実行します。
4. **挿入 > コントロール > オブジェクトの識別ダイアログ** を選択します。
5. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログボックスが開きます。コントロールを識別するブラウザーを選択し、**識別の開始** をクリックします。**オブジェクトの識別** ダイアログボックスが開きます。
6. **識別の開始** をクリックします。
7. テストしているアプリケーションにカーソルを移動します。カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが強調表示されます。関連するロケータ文字列またはオブジェクトマップ項目は、**選択済みロケータ** テキストボックスに表示されます。
8. 使用するコントロールが強調表示されているときに、コントロールをクリックします。
9. 省略可能：**ロケータの詳細の表示** をクリックすると、ロケータ ツリーや関係する属性のすべてが**ロケータ属性** テーブルに表示されます。
10. 省略可能：記録したロケータ属性は、**ロケータ属性** テーブルの別のロケータ属性で置き換えることができます。

たとえば、記録したロケータは以下のように表示されます。

```
/BrowserApplication//BrowserWindow//input[@id='loginButton']
```

ロケーター属性 テーブルに textContents Login がリストされている場合、以下のようにしてロケーターを手動で変更できます。

```
/BrowserApplication//BrowserWindow//input[@textContents='Login']
```

新しいロケーターは、**選択済みロケーター** テキスト ボックスに表示されます。

11テスト をクリックして、ロケーターがコントロールを正しく認識することを確認します。

12ロケーターを使用する新しいステップをビジュアル テストに追加するには、**貼り付け** をクリックします。**操作の選択** ダイアログ ボックスが開きます。コントロールに対して実行する操作を選択し、**OK** をクリックします。

ビジュアル テスト内でのコントロール プロパティ データの使用

Silk Test Workbench では、テスト アプリケーションの任意のコントロールのプロパティ値を、ビジュアル テストのデータとして使用できます。この場合、プロパティ変数を、ビジュアル テスト内で定義されているローカル変数に設定します。たとえば、コントロールの ID を変数に設定してからテスト ロジックでこの変数を使用すると、アプリケーションの他のコントロールで同じ ID が使用されていないことを確認できます。

変数に設定できるプロパティは、そのコントロールで使用可能なプロパティによって異なります。すべてのコントロールで同じプロパティが使用可能なわけではありません。

コントロール プロパティ 値を変数に設定するには、ビジュアル テストにその変数を作成する必要があります。

1. コントロール プロパティを変数に設定するビジュアル テストを開きます。
2. コントロール プロパティの値を保持するローカル変数を作成します。
3. コントロールのプロパティを変数に設定するステップを挿入する場所の直前のステップを選択します。
目的のプロパティ データを持つコントロールが含まれるアプリケーションが、ステップを挿入するテスト内のポイントで使用可能になっていることを確認してください。
4. **挿入 > コントロールのプロパティ** を選択します。選択したステップのあとにステップが作成されます。ステップのプロパティが **プロパティ** ペインに表示されます。
5. **プロパティ** ペインの **ロケーター** テキスト ボックスで、変数を割り当てるコントロールを識別します。
 - **テスト対象アプリケーション**： テスト アプリケーションから可視のリスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**： テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接リスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**： リスト ボックスまたはコンボ ボックス コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケーターを編集することもできます。
 - **選択**： このボタンをクリックして、リテラル、変数、式の結果、またはアクティブ データ値を割り当てます。
6. **プロパティ** リストから、コントロールのプロパティとしてリテラル、変数、式、またはアクティブ データのいずれを使用するかを指定します。
この値はローカル変数に格納されます。
7. **ローカル変数名** リストから、選択したコントロールおよびプロパティで使用する変数を選択します。
8. 省略可能：**その他** カテゴリのプロパティを編集し、ステップに関する一般的な情報を入力します。

ビジュアル テストにおける画面プレビューとステップの同期

Silk Test Workbench では、記録中と再生中にアプリケーションのスナップショットをキャプチャします。このスナップショットには、キャプチャされた画面と、さまざまなプロパティを持つ各種コントロールの列挙が含まれます。キャプチャされた画面は **画面プレビュー** に表示されます。

Silk Test Workbench では、スナップショットは以下のような状況のときに取得されます。

- 記録中のすべての自動テスト ステップの前。



注: SAP アプリケーションの場合、スナップショットはすべての自動テスト ステップの前ではなく、画面が変わったときに発生します。

- ビジュアル テストで Using ステップを実行した結果のスナップショット。
- 再生エラーが発生した場合。



ヒント: アプリケーション画面の内容を更新するのに時間のかかるテスト アプリケーションの場合は、アプリケーション画面のスナップショットを取得する前に待機するよう Silk Test Workbench を設定できます。詳細については、**アタッチとスクリーンショット間の遅延** オプションを参照してください。

自動テスト ステップ

すべての自動テスト ステップには独自のスナップショットがあり、**画面プレビュー** では (グループ画面) または (ステップ画面) というラベルが付けられます。(グループ画面) には、現在のダイアログ ボックスまたは Web ページのメイン コンテキストが含まれています。一方、(ステップ画面) は、現在のダイアログ ボックスまたは Web ページでの特定の操作に関連付けられています。

Silk Test Workbench では、アプリケーションのスナップショットは、そのスナップショットを参照する実際のテスト ステップの前に取られます。画面コンテキストが変わる自動ステップ (新しいダイアログ ボックスまたは Web ページを開くなど) では、Using ステップは、自動ステップの前のアプリケーション状態を表す画面キャプチャに関連付けられます。**画面プレビュー** では、Using ステップには (グループ画面) というラベルが付けられます。たとえば、Web ページの **画面プレビュー** での (グループ画面) は、**テスト ステップ** ペインの Using Browser Application 'BrowserApplication' ステップと対応する場合があります。Using ステップに続くテスト ステップは、現在のダイアログ ボックスまたは Web ページのコンテキストで発生する単一の操作に対応し、**画面プレビュー** では (ステップ画面) というラベルが付けられます。たとえば、テキストフィールドに郵便番号を入力した場合に、対応するテスト ステップは「'92109' を入力します」であるとします。**画面プレビュー** では、入力した値の前の郵便番号のテキストフィールドと郵便番号の値の周囲に黒い四角形が表示されます。したがって、92109 の前の値が 18966 だった場合、**画面プレビュー** には 18966 が表示されます。

Using ステップを無効にした場合、Using ステップに続くステップは、無効にした Using ステップの前の Using ステップに関連付けられます。たとえば、Using ステップ 26 を無効にした場合、それに続くテスト ステップはステップ 26 の前の Using ステップを使用します。



注: 画面プレビューからコントロールを挿入すると、**画面プレビュー** に (グループ画面) スナップショットが表示されます。コントロールは記録されなかったため、スナップショットは取得されませんでした。この結果、(グループ画面) が使用されます。

その他のテスト ステップのタイプ

ほとんどの場合、自動ステップに関連付けられていないテスト ステップでは、**画面プレビュー** にスナップショットは表示されません。たとえば、ビジュアル テストにスクリプトを挿入すると、**画面プレビュー** には **テスト ステップ** ペインに表示されるテキストが表示されます。

設定の構成

デフォルトでは、Silk Test Workbench によって記録中にグループと個別両方の画面キャプチャが記録されます。これらの設定は、**オプション** ダイアログ ボックスを使用して、グループ画面キャプチャのみを記録するように、または画面キャプチャを記録しないように変更できます。**ツール > オプション** を選択し、**記録** を展開します。**出力** をクリックし、**ビジュアル テスト** をクリックして、これらのオプションを構成します。

オプション ダイアログ ボックスでは、ビジュアル テストのグループと個別両方の画面キャプチャの再生画面キャプチャ設定を構成するか、すべてのビジュアル テストに適用されるグローバル オプションを設定できます。**ツール > オプション** を選択し、**再生** を展開します。**結果** をクリックし、**ビジュアル テスト** をクリックして、これらのオプションを構成します。

ビジュアルテストの個別の設定を構成するには、ビジュアルテストを記録し、次に、グローバル再生値を上書きする再生設定の設定ステップを挿入します。設定には、画面キャプチャと画面キャプチャ (テストステップ) があります。

たとえば、パフォーマンスを向上させるために、特定のステップについては **画面キャプチャ (テストステップ)** の再生設定を **いいえ** に設定し、個別の画面キャプチャが必要な残りのステップについては **画面キャプチャ (テストステップ)** を **はい** に戻す場合があります。

画面テストステップのキャプチャ画面の更新

ビジュアルテストまたは結果の **画面プレビュー** から、以前にキャプチャされた画面テストステップの画面を即座に更新することができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。Silk Test Workbench では、キャプチャされた画面を更新する方法には以下の2つがあります。

テストアプリケーションからの画面の再キャプチャ	この方法は、ビジュアルテストの 画面プレビュー から実行します。テストアプリケーションを開き、新しい画面が含まれるウィンドウを表示して、それを再キャプチャします。
再生中にキャプチャされた画面の使用	この方法は、結果の 画面プレビュー から実行します。テストアプリケーションにアクセスする必要はありません。

再生中に記録された画面を使用したキャプチャ画面の更新

ビジュアルテストまたは結果の **画面プレビュー** から、以前にキャプチャされた画面テストステップの画面を即座に更新することができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

1. ビジュアルテストの結果を開きます。
2. **テストステップ** ペインで、更新が必要な画面が含まれるステップを選択するか、画面が含まれるストーリーボードのサムネイルをクリックします。
3. 現在のステップのみを更新する場合は、**画面プレビュー** または **テストステップ** ペインから **操作 > 画面の更新** を選択します。Silk Test Workbench により、ビジュアルテストが開き、更新された画面が **画面プレビュー** に表示され、ビジュアルテストの画面項目を更新するか尋ねる確認メッセージボックスが表示されます。この機能を利用するには、結果とビジュアルテストの名前を同じにする必要があります。ビジュアルテストの名前を変更した場合は、その結果の名前も変更してください。変更しなかった場合は、ビジュアルテストの画面項目を変更しようとしても、Silk Test Workbench でビジュアルテストが見つかりません。



注: 画面テストステップを更新する前に、記録中にキャプチャされた画面と再生中にキャプチャされた画面の違いを表示することができます。**操作 > 相違点の表示 > 横に並べる** を選択してください。

4. **はい** をクリックして、再生中にキャプチャされた画面でビジュアルテストを更新します。ビジュアルテストが開きます。
5. ビジュアルテストのすべての画面を更新する場合は、**画面プレビュー** または **テストステップ** ペインから **操作 > すべての画面を更新** を選択します。この機能を利用するには、結果とビジュアルテストの名前を同じにする必要があります。ビジュアルテストの名前を変更した場合は、その結果の名前も変更してください。変更しなかった場合は、ビジュアルテストの画面項目を変更しようとしても、Silk Test Workbench でビジュアルテストが見つかりません。
6. 確認メッセージボックスの **結果へ戻る** チェックボックスをオンにすると、**はい** をクリックしたあとで **結果** ウィンドウが表示されます。

テストアプリケーションからキャプチャされた画面の更新

ビジュアルテストまたは結果の **画面プレビュー** から、以前にキャプチャされた画面テストステップの画面を即座に更新することができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

1. テストアプリケーションを起動します。
2. テストアプリケーションを操作して、ビジュアルテストで情報の更新が必要な画面が含まれるウィンドウを表示します。
3. ビジュアルテストを開き、更新が必要な画面が含まれる画面ステップを選択するか、画面が含まれるストーリーボードのサムネイルをクリックします。
4. **操作 > 画面の更新** を選択します。テストアプリケーションで更新するテストステップに該当する画面が見つかり、Silk Test Workbench によってその画面が強調表示されます。テスト画面の周囲が四角の境界線で囲まれて点滅します。テストアプリケーションで画面が見つからない場合は、テストアプリケーションでウィンドウが見つからないことを示すメッセージが表示されます。ウィンドウの名前が変更されているか、Silk Test Workbench でウィンドウを認識できないようにウィンドウのプロパティが変更されている場合があります。**はい** をクリックしてテストアプリケーションで更新されたウィンドウを再識別するか、別のウィンドウを識別してください。ウィンドウ情報を再キャプチャしないでビジュアルテストに戻るには、**いいえ** をクリックします。
5. テストアプリケーションの画面からキャプチャされた更新情報で、テストステップを更新するかどうかを確認するメッセージが表示されます。
 - テストステップを更新するには、**はい** をクリックします。
 - テストステップを更新しないでメッセージボックスを閉じるには、**いいえ** をクリックします。

ビジュアルテストでの項目識別子の使用

ビジュアルテストでの項目識別子は、単純な数値ではなく、かぎ括弧で囲んだインデックスで指定します。

例

たとえば、GetCellText メソッドで DomTable の 2 行 2 列のセルのテキストを返すには、次のステップをビジュアルテストに追加します。

行 '[2]'、列 '[2]' のセルテキストを返します。また、戻り値を st_LastReturnValue に割り当てます

このステップのプロパティのパラメータセクションで、row パラメータを [2] に、そして、col パラメータも [2] に設定します。



注: ビジュアルテストでは、項目識別子にはかぎ括弧が必要なため、これらのパラメータとして [2] ではなく 2 を設定すると、正しく動作しません。

ビジュアルテストでのテストデータの再利用

ビジュアルテストで使用できる再利用の種類を示します。

ビジュアルテストにおける変数

アプリケーションのテストでは、通常はデータを使用する必要があります。アプリケーションの一部をテストするためにデータを使用したり、そのアプリケーションやその他のアプリケーション全体でデータを使用したりできます。データをフィールドに入力して、そのフィールドがどのようにユーザー入力を受け入れるかをテストできます。また、データを使用して、テストアプリケーションに対してテストがどのように再生されるかを確認することもできます。

記録中、コントロールへのデータ入力はリテラルデータとして記録されます。たとえば、WPF TextBox コントロールに対して入力された値がある場合、その結果のテストステップは次のようになります: 「"Ford" を入力します」。

この値は有効ですが、これではビジュアルテストはこのテストステップで Ford 以外の値のデータを使用することができません。ただし、値 Ford を、このデータを表す変数で置き換えると、変数で使用する任意のデータが TextBox コントロールに入力されます。

変数を使用すると、ビジュアルテストで使用するデータが定数である必要がなくなるため、テストの柔軟性が高まります。

ビジュアルテストに変数を定義する場合は、その変数がどのような種類のデータを保持できるかを定義する必要があります。このような変数の特性を、データ型と呼びます。ビジュアルテストでは、以下のデータ型の変数を使用できます。

- テキスト (String)
- 数値 (Long)
- 数値 (Double)
- ブール値 (True/False)
- 数値 (Long Long)
- 列挙

たとえば、Silk Test Workbench でコントロールへのデータ入力が記録される時、記録された値が二重引用符 ("") で囲まれている場合、通常はテキストデータ型であり、テキストデータ型の変数で置き換えることができます。

ビジュアルテストでは、以下のタイプの変数を使用します。

- ローカル：通常 1 つのビジュアルテスト内で定義され、使用される変数です。ローカル変数を使用して、グローバル変数やビジュアルテスト変数などの他のタイプの変数の値を格納することができます。**挿入 > 変数 > ローカルの追加** を選択してローカル変数を追加するか、または <<開始>> ステップの **プロパティ** ペインでローカル変数を定義します。
- グローバル：グローバル変数は、ビジュアルテストのコンテキスト内で定義されますが、そのグローバル変数が含まれているビジュアルテストを実行する、任意のビジュアルテストで使用できます。**ビジュアルナビゲータ** でビジュアルテストを操作している場合は、**挿入 > 変数 > グローバルの設定** を選択して、グローバル変数の値を定義します。他のビジュアルテストでは、**挿入 > 変数 > グローバルの取得** を選択して、グローバル変数の値をローカル変数に割り当てます。
- パラメータ：1 つのビジュアルテストから別のビジュアルテストに渡される変数です。
- 入力/出力パラメータ：入力パラメータを使用して、ビジュアルテストからスクリプトにデータを渡します。出力パラメータを使用して、スクリプトからビジュアルテストにデータを渡します。
- 予約：事前に定義されている変数です。この変数を使用して、ビジュアルテストの再生に関する情報やその他の有用な再生結果情報を取得することができます。予約変数の名前は、必ず接頭部 ST で始まります。予約変数は読み取り専用ですが、式の中で他のタイプの変数と共に使用できます。



ヒント: 変数を使用することによって、スクリプトでデータを再利用することもできます。詳細については、「スクリプトにおける変数」を参照してください。


ビジュアルテストの再生における、変数の使用状況の監視


Silk Test Workbench では、ビジュアルテストの再生時に変数の使用を追跡できます。変数の使用状況と再生中に変数に格納される値を追跡することによって、ビジュアルテストのデバッグに関する有益な情報を入手できます。

ビジュアルテストを再生する前に、ローカル変数ウィンドウを表示します。このウィンドウには、ビジュアルテストの再生中に変数が作成、使用、削除されたときに変数の使用状況が表示されます。

ローカル変数ウィンドウでビジュアルテストに 1 つまたは複数のブレークポイントを設定し、テストステップを段階的に再生して、テスト実行の重要なポイントで変数の使用状況に関するスナップショットを取得できます。

1. 再生中に監視する変数が含まれているビジュアルテストを開きます。
2. **デバッグ > ローカル変数** を選択します。ローカル変数ウィンドウが開きます。このウィンドウは、移動、ドッキング、非表示、またはサイズ変更が可能です。
3. 選択した場所で再生を停止するようにブレークポイントを設定し、変数を追跡しやすくします。
4. ビジュアルテストを再生します。デバッグモードの場合は選択した場所で再生が停止し、既存の変数ごとにその現在値がローカル変数ウィンドウに表示されます。

 **ヒント:** また、ローカル変数 ウィンドウには、変数を使用しているビジュアル テストの名前も表示されます。これは、複数のビジュアル テストを再生している場合、またはビジュアル テストのステップ内で他のビジュアル テストを再生している場合に役立ちます。

 **ヒント:** ローカル変数 ウィンドウからクリップボードに変数をコピーするには、変数を右クリックします。**コピー** を選択すると、現在選択している変数（ノード）とそのすべての子変数がコピーされます。また、**値のコピー** を選択すると、現在選択している変数の **値** だけがコピーされます。変数の **値** が空の場合は、このメニュー項目は無効になります。

ローカル変数

ローカル変数は、1 つのビジュアル テスト内で使用するために作成された変数です。これらはビジュアル テストの <<開始>> ステップで定義し、**プロパティ** ウィンドウで編集し、再生中にビジュアル テストで使用できます。ローカル変数を作成すると、変数を作成したビジュアル テストで使用できます。使用できる場所とタイミングは、以下のとおりです。


- テストロジック
- 結果コメント
- 式
- 他のビジュアル テストで設定されたグローバル変数を格納するとき

ビジュアル テストへのローカル変数の追加

ローカル変数は、1 つのビジュアル テスト内で使用するために作成された変数です。これらはビジュアル テストの <<開始>> ステップで定義し、**プロパティ** ウィンドウで編集し、再生中にビジュアル テストで使用できます。ローカル変数を作成すると、変数を作成したビジュアル テストで使用できます。

1. ローカル変数を追加するビジュアル テストを開きます。
2. **挿入 > 変数 > ローカルの追加** を選択します。**ローカル変数の追加** ダイアログ ボックスが開きます。
3. 変数の名前を **変数名** フィールドに入力します。

ローカル変数名に有効な文字は、大文字/小文字の英数字とアンダースコア（「ABC」、「abc」、「_」）です。また、ローカル変数名は一意にする必要があります。

 **注:** 変数名は、「st_」で始めることはできません。

追加または編集される変数の名前が重複する場合、Silk Test Workbench によってそれが通知されます。この場合、追加または編集する変数に固有の名前を付ける必要があります。


4. **型** リストから変数の型を選択します。

ローカル変数を追加または編集する場合、表示されるデータ型は、<<開始>> ステップのプロパティ ウィンドウを使用して追加しようとしている変数のデータに基づきます。式の結果を格納するとき、リストから型を選択します。

Silk Test Workbench は、次の変数の型をサポートします。

テキスト

変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

 **注:** テキスト データ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

数値 (Long)

変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。（小数値または 10 進数値で使用される）ピリオドは使用できません。

数値 (Double)

変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。

ブール値 (True/False) 変数の値は True または False です。

数値 (Long Long) 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙 この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

5. 省略可能: 変数の初期値を **初期値** フィールドに入力します。この値は、ビジュアルテスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値(Long Long)である場合、初期値には、整数でない文字 (テキスト文字など) を含めることはできません。ブール型変数の場合、リストから **False** または **True** を選択します。変数の場合、初期値は必須ではないため、このフィールドは空白のままにすることができます。この値は、ビジュアルテストの他の箇所で決定できます。
6. **OK** をクリックします。変数名と初期値 (存在する場合) が、<<開始>> ステップの **プロパティ** ペインの選択した変数型の下に表示されます。
7. 他のローカル変数を作成するには、必要に応じて上記のステップを繰り返します。各変数型の値領域は、現在定義されている各変数型の数を示します。たとえば、数値 (Long) 項目の値領域が **3 個の項目** の場合、現在このビジュアルテストには Long データ型のローカル変数が 3 つ定義されています。

ビジュアルテスト用ローカル変数の編集

ローカル変数は、<<開始>> ステップの **プロパティ** ペインで編集できます。

1. 編集するローカル変数が含まれているビジュアルテストを開きます。
2. ビジュアルテストの <<開始>> ステップを選択します。<<開始>> ステップのプロパティが **プロパティ** ペインに表示されます。
3. **プロパティ** ペインで、**変数** カテゴリを探るか **変数** をダブルクリックしてカテゴリを展開し、ビジュアルテストに定義されたローカル変数を表示します。

変数 カテゴリで定義される変数には、以下の型があります。

ビジュアルテスト 他のビジュアルテストで定義され、このビジュアルテストに渡される変数。

テキスト 変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。



注: テキストデータ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

数値 (Long) 変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。

数値 (Double) 変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。


ブール値 (True/False) 変数の値は True または False です。

数値 (Long Long) 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙 この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

4. 編集する変数の値領域をクリックし、値領域内の **変数の編集** ボタンをクリックします。**変数の編集** ダイアログ ボックスが開きます。
5. 変数の名前を **変数名** フィールドに入力します。
ローカル変数名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。また、ローカル変数名は一意にする必要があります。

 **注:** 変数名は、「st_」で始めることはできません。

追加または編集される変数の名前が重複する場合、Silk Test Workbench によってそれが通知されます。この場合、追加または編集する変数に固有の名前を付ける必要があります。

6. 省略可能: 変数の初期値を **初期値** フィールドに入力します。この値は、ビジュアルテスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値(Long Long)である場合、初期値には、整数でない文字 (テキスト文字など) を含めることはできません。ブール型変数の場合、リストから **False** または **True** を選択します。変数の場合、初期値は必須ではないため、このフィールドは空白のままにすることができます。この値は、ビジュアルテストの他の箇所ですべて決定できます。
7. **OK** をクリックします。変数名と初期値 (存在する場合) が、<<開始>> ステップの **プロパティ** ペインの選択した変数型の下に表示されます。
8. 他のローカル変数を編集するには、必要に応じて上記のステップを繰り返します。
スクリプトに渡す変数を編集する場合は、変数値を渡されるテスト内でスクリプトを実行する必要があります。

ビジュアルテストからのローカル変数の削除

ローカル変数がビジュアルテストで必要なくなった場合は、ローカル変数を削除します。

1. ローカル変数を削除するビジュアルテストを開きます。
2. 次のいずれか 1 つのステップを行います:
 - すべてのローカル変数をテストから削除するには、<<開始>> ステップを選択して Del を押します。
 - 特定のローカル変数をテストから削除するには、<<開始>> ステップを選択し、**プロパティ** ペインで変数を選択して **削除** をクリックします。

グローバル変数

Silk Test Workbench では、ビジュアルテスト内のデータを表す変数を作成して、同じデータベース内の他のビジュアルテストからそのデータを使用できます。この場合は、グローバル変数にデータを設定します。

グローバル変数は、複数のテストで使用する必要のあるデータ、または定数として使用するデータに対して有効です。データをグローバル変数に定義すると、そのデータを必要とする個々のビジュアルテストで、そのグローバル変数を参照できます。グローバル変数で表されるデータを変更する場合は、データを 1 箇所ですべて変更します。各ビジュアルテストを次回再生したときに、自動的に新しいデータが使用されます。

グローバル変数を使用するには、1 つのビジュアルテストで変数を作成し、他のビジュアルテストでその値をローカル変数に格納します。これにより、1 つのテスト内にデータを作成し、テストソリューション

全体でそのテストを再利用できます。グローバル変数のデータがローカル変数に読み込まれると、そのデータはローカル変数と同じように使用できます。

ローカル変数とは異なり、グローバル変数は、その変数が作成されたビジュアル テストを再生するときだけでなく作成され、削除されるわけではありません。グローバル変数は、この変数が含まれるビジュアル テストを再生する任意のビジュアル テストから、アクセスおよび変更できます。

ビジュアル テストにグローバル変数を作成するには、グローバルの設定ステップを作成し、そのステップに変数を定義するプロパティを設定します。各グローバルの設定ステップは、1 つのグローバル変数定義を表します。

ビジュアル テストにおけるグローバル変数の管理

ビジュアル テストでグローバル変数値を使用するには、スクリプトに定義されたローカル変数にグローバル変数を格納し、ローカル変数として使用する必要があります。また、この変数が含まれるビジュアル テストを、値を使用するビジュアル テストに挿入する必要があります。これには、グローバル変数が含まれるビジュアル テストのすべてのステップが含まれますが、これにより不要なテスト結果が生成される場合があります。

このため、テストソリューションのすべてのグローバル変数は、1 つのビジュアル テストに作成して管理することをお勧めします。これには以下のような利点があります。

- テストソリューションで使用する各グローバル変数定義を 1 箇所を確認できます。
- グローバル変数名の重複を回避できます。
- グローバル変数が含まれるビジュアル テストを他のビジュアル テストに挿入した場合、再生時にグローバル変数のみが呼び出し元のスクリプトに読み込まれます。グローバル変数が含まれるビジュアル テストで、他の不要なステップが再生されません。

ローカル変数を使用してグローバル変数を格納すると、格納前にそのローカル変数に設定されていた値は、グローバル変数の値に変更されます。

ビジュアル テストに定義されたグローバル変数と同じ名前のローカル変数を定義できます。グローバル変数の値を変更した場合でも、同じ変数名を持つローカル変数の値は変更されません。ただし、ビジュアル テストで誤ってデータを使用したり混乱したりすることのないように、グローバル変数名はローカル変数名や他の変数名とは異なる名前にすることをお勧めします。

2 つのビジュアル テストで同じ名前のグローバル変数を作成することもできますが、グローバル変数に予想外の値が格納されたまま使用されるおそれがあるため、このような使用法も避けます。

ビジュアル テスト用グローバル変数の作成

グローバル変数は 1 つのビジュアル テスト内で使用するために作成しますが、他のビジュアル テストからも使用できます。グローバル変数は、グローバルの設定ステップに **割り当て** プロパティを使用して定義します。グローバル変数を作成すると、このグローバル変数が定義されているビジュアル テストを実行するビジュアル テストで、変数を使用できます。

1. グローバル変数を作成するビジュアル テストを開きます。
2. グローバル変数を作成する場所の直前のステップを選択します。
3. **挿入 > 変数 > グローバルの設定** を選択します。選択したステップのあとにグローバル変数の設定ステップが作成されます。グローバル変数の設定ステップのプロパティが、**プロパティ** ペインに表示されます。
4. グローバル変数の設定ステップの **プロパティ** ペインで、プロパティをカテゴリごとにグループ化します。
5. **割り当て** カテゴリ プロパティを更新して、グローバル変数の名前およびグローバル変数に割り当てる値を定義します。**割り当て** カテゴリ プロパティを更新し、ビジュアル テストを保存すると、グローバル変数の設定ステップは「グローバル変数 'グローバル変数名' に '値' を設定します」となります。
6. 他のグローバル変数を作成するには、必要に応じてステップ 2 から 5 を繰り返します。グローバル変数を設定すると、他のビジュアル テストのローカル変数に読み込むことができます。このためには、まず

グローバル変数が定義されたビジュアルテストを、ローカル変数が定義されたビジュアルテストに挿入する必要があります。

ビジュアルテストで使用するグローバル変数の取得

グローバルの設定ステップでグローバル変数が作成されると、それらのグローバル変数は、グローバルの設定ステップを含むビジュアルテストを実行する任意のビジュアルテストで使用できます。グローバル変数は、まずこのグローバル変数を使用するビジュアルテストに作成されたローカル変数に読み込んでから、ローカル変数として使用する必要があります。

1. グローバル変数を使用するビジュアルテストを開きます。
2. グローバル変数値を格納するローカル変数を作成します。
ローカル変数は任意のテストステップに作成できます。
3. グローバル変数を作成したビジュアルテストを実行するステップを作成します。同じビジュアルテストに定義されているローカル変数にグローバル変数を取得する場合は、このステップをスキップしてください。
これにより、グローバル変数を使用するビジュアルテストから、グローバル変数にアクセスできます。
4. グローバル変数を取得する場所の直前のステップを選択します。
5. **挿入 > グローバル > グローバルの取得** を選択します。選択したステップのあとにグローバル変数の内容の取得ステップが作成されます。グローバル変数の内容の取得ステップのプロパティが **プロパティ** ペインに表示されます。
6. グローバルの取得ステップの **プロパティ** ペインで、プロパティをカテゴリごとにグループ化します。
7. **割り当て** カテゴリプロパティを更新して、グローバル変数の名前およびグローバル変数に割り当てる値を定義します。プロパティを更新してビジュアルテストを保存すると、グローバル変数の内容の取得ステップが、「グローバル変数 '[グローバル変数名]' の内容を取得して、ローカル変数 '[ローカル変数名]' に格納します」のように表示されます。
8. 他のグローバル変数を作成するには、ステップ 2 から 7 を繰り返します。グローバル変数をローカル変数に取得すると、ローカル変数にグローバル変数の値が格納されます。この値をビジュアルテストの式で使用したり、同様にローカル変数として使用したりします。

予約変数

Silk Test Workbench では、ビジュアルテストの再生に関するデータを公開し、このデータをビジュアルテストの再生時に使用できます。このデータには、再生結果全体や特定の再生結果の合格/失敗の情報、コントロールに入力された現在値、最後に発生したエラーのテキスト、最後に実行されたステップの情報などが含まれます。公開された再生データには、さまざまな用途があります。たとえば、実行されたステップの成否に基づいて条件を作成できます。この条件を再生時の結果への対処を決定するために使用できません。

ビジュアルテストの再生データは、予約変数に格納されます。予約変数は、Silk Test Workbench によって定義されている変数です。この変数は読み取り専用です。つまり、その値をビジュアルテストで返して、使用することができますが、ユーザーが値を設定することはできません。予約変数は、必ず接頭部「st_」で始まります。

予約変数は、値として変数を使用可能な任意のプロパティの値として返すことができます。

予約変数は、同じテスト内で実行されるビジュアルテスト間で双方向に継承されます。たとえば、ビジュアルテスト B をビジュアルテスト A に挿入すると、ビジュアルテスト B はビジュアルテスト A の予約変数値を継承します。ビジュアルテスト B を再生したあと、ビジュアルテスト A はビジュアルテスト B の予約変数値を継承します。

ビジュアルテスト内で使用する予約変数を選択するには、**変数の選択** ダイアログボックスを使用します。

予約変数の一覧

以下の表に、ビジュアルテストの再生に関連するデータを返す予約変数を示します。

変数名	デフォルト値	タイプ	説明
st_LastError	空の文字列	テキスト	再生時に最後に生成されたエラーのテキストを返します。再生時 (st_LastError 値が返される場所まで) にエラーが生成されなかった場合、この値は Nothing です。
st_LastReturnValue	False	各種	ビジュアル テストで最後に実行されたステップの値を返します。戻り値は、ステップ タイプによって異なります。 ほとんどの自動ステップは、操作が成功すると True を返し、失敗すると False を返します。文字列、列番号、日付などの値を返す自動ステップまたはその他のステップでは、st_LastReturnValue 変数にその情報を返します。
st_LastValidation	False	ブール値	最後に実行された検証ロジックの結果を返します。 True = 最後に実行された検証ロジックは成功しました。 False = 最後に実行された検証ロジックは失敗しました。
st_Project	実行中のビジュアル テストが格納されているプロジェクトの名前	テキスト	実行中のビジュアル テストが格納されているプロジェクトの名前を返します。
st_RepeatListValue	空の文字列	テキスト	最後にアタッチされたコントロールの列挙が実行されたとき、そのコントロールの現在の入力値を返します。たとえば、繰り返しロジックを使用して ListBox コントロールの項目を繰り返し検索する場合、この変数は ListBox 内の現在の項目を返します。 この値は、コントロールに含まれる項目の名前を返します。繰り返しロジックを使用してリスト内の項目を繰り返し検索し、選択する値として st_RepeatListValue を使用します。
st_VisualTest	実行中のビジュアル テストの名前	テキスト	実行中のビジュアル テストの名前を返します。

予約変数の戻り値

予約変数およびさまざまなステップ タイプに対して返される値を以下に示します。

ステップ タイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
<<開始>>	True	変更なし	変更なし	変更なし
<<終了>>	True	該当なし	該当なし	該当なし
自動ステップ	Open Agent メソッドの戻り値。	メソッドが設定したエラー メッセージが格納されます。	変更なし	変更なし

ステップタイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
ビジュアルテストの再生	True = ビジュアルテストが実行されました。 False = ビジュアルテストが実行されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
検証	最後に実行された検証ロジックの結果を返します。 True = 最後の検証に成功しました。 False = 最後の検証に失敗しました。	すでに存在しない変数の参照など、評価された式が失敗した場合にエラーメッセージが格納される場合があります。	変更なし	変更なし
グローバル変数の内容の取得	True = 変数の内容を取得しました。 False = 内容が取得されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
グローバル変数の設定	True = 変数の内容が設定されました。 False = 変数の内容が設定されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
再生設定の取得	True = 再生設定がローカル変数に入力されました。 False = 再生設定がローカル変数に入力されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
再生設定の設定	True = 再生設定が設定されました。 False = 再生設定が設定されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
プログラムの実行	ShellExecute API関数の戻り値 (32より大きければ成功)。	st_LastReturnValue > 32 : 空です。	変更なし	変更なし

ステップタイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
	これは VBA Shell() 関数によく似ています。	st_LastReturnValue <= 32 : Windows から返されるエラーメッセージが格納されます。たとえば、2 の場合、メッセージは「指定されたファイルが見つかりません。」となります。		
メッセージボックス	<p>押されたボタンを表す数値 (Long)。</p> <p>1 = OK</p> <p>2 = キャンセル</p> <p>3 = 中止</p> <p>4 = 再試行</p> <p>5 = 無視</p> <p>6 = はい</p> <p>7 = いいえ</p>	すでに存在しない変数の参照など、評価された式が失敗した場合にエラーメッセージが格納される場合があります。	変更なし	変更なし
開始タイマー	<p>True = タイマーは開始されます。</p> <p>False = タイマーはすでに開始されています。</p>	<p>st_LastReturnValue = True : 空です。</p> <p>st_LastReturnValue = False : エラーメッセージが格納されます。</p>	変更なし	変更なし
再開タイマー	<p>True = タイマーは停止タイマーのあと、再開されます。</p> <p>False = タイマーはすでに実行中です。</p>	<p>st_LastReturnValue = True : 空です。</p> <p>st_LastReturnValue = False : エラーメッセージが格納されます。</p>	変更なし	変更なし
タイマーのリセット	True	空	変更なし	変更なし
停止タイマー	<p>指定されたタイマーの経過時間 (ミリ秒単位、Long)。</p> <p>タイマーが起動していない場合は False (ブール)。</p>	<p>st_LastReturnValue = True : 空です。</p> <p>st_LastReturnValue = False : エラーメッセージが格納されます。</p>	変更なし	変更なし
遅延	True	空	変更なし	変更なし
ラベル [ラベル名] へ移動	<p>True = ステップが実行されました。</p> <p>False = ステップが実行されませんでした。</p>	<p>st_LastReturnValue = True : 空です。</p> <p>st_LastReturnValue = False : エラーメッセージが格納されます。</p>	変更なし	変更なし

ステップタイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
エラー処理	True = ステップが実行されました。 False = ステップが実行されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
結果コメント	True = ステップが実行されました。 False = ステップが実行されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
Using [ロケータ名] (画面ステップ)	True = ステップが実行されました。 False = ステップが実行されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
項目プレースホルダの破損	False	空	変更なし	変更なし
ロジック (If/ElseIf/Else/EndIf)	変更なし	すでに存在しない変数の参照など、評価された式が失敗した場合にエラーメッセージが格納される場合があります。	変更なし	変更なし
ループ (Repeat/EndRepeat)	変更なし	すでに存在しない変数の参照など、評価された式が失敗した場合にエラーメッセージが格納される場合があります。	変更なし	ループがリストコントロールの内容を反復しない場合は変更なし。 ループがリスト型コントロールの内容を反復する場合は、項目のインデックスと値の両方が複合値として格納されます (ビジュアルテストでは書式設定文字列として表されます)。
停止	True	空	変更なし	変更なし
ラベル	変更なし	変更なし	変更なし	変更なし
コントロールのプロパティ	True = コントロールが見つかった変数が正しく設定されました。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	常にプロパティの値に設定されます (ローカル変数が指定されない場合でも)。

ステップタイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
	False = コントロールが見つからないか、ターゲット変数が見つからないなどのエラーが発生しました。			
eCATT 引数の内容の取得	引数の内容を取得した場合は True、内容を取得できなかった場合は False。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
eCATT 引数の設定	引数の内容を設定した場合は True、そうでない場合は False。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし

ビジュアルテスト間でのデータの共有

Silk Test Workbench では、ビジュアルテストから別のビジュアルテストにデータを渡すことができます。1つのテストでデータを格納するパラメータを作成します。次に、このテストを別のビジュアルテストで実行し、パラメータを渡してそのデータを使用します。

パラメータは、このパラメータが含まれるテストを実行する、任意のビジュアルテストに渡されます。パラメータが渡されると、データ値を割り当てたり、式、テストロジック、またはパラメータが使用可能な任意の場所でパラメータを使用したりできます。また、パラメータ値をローカル変数に割り当て、このローカル変数を使用することもできます。

パラメータを作成して、任意のビジュアルテストでデータを使用できるようにします。パラメータを使用するビジュアルテストで、以下の作業を実行します。

- パラメータを作成するビジュアルテストを挿入します。
- パラメータに値を割り当てます。

ビジュアルテストへのパラメータの追加

ビジュアルテストは、入力パラメータを使用してスクリプトまたは他のビジュアルテストからデータを受け取ったり、出力パラメータを使用して他のビジュアルテストにデータを渡したりできます。パラメータは、ビジュアルテストの <<開始>> ステップで定義し、**プロパティ** ウィンドウで編集し、再生中にビジュアルテストで使用できます。パラメータを作成すると、パラメータを作成したビジュアルテストで使用できます。

1. パラメータを作成するビジュアルテストを開きます。
2. **タスク** ペインで <<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**変数** カテゴリに移動します。
4. 作成するパラメータの種類を選択します。
 - 新しい入力パラメータを作成する場合は、**入力パラメータ** フィールドをクリックします。
 - 新しい出力パラメータを作成する場合は、**出力パラメータ** フィールドをクリックします。
5. **パラメータの追加** をクリックします。**パラメータの追加** ダイアログボックスが開きます。
6. パラメータの名前を **名前** フィールドに入力します。

パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア（「ABC」、「abc」、「_」）です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。

7. 型 リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。


ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。
列挙	<p>このパラメータは一連の値をグループ化し、1 から n まで順番に番号を付けます。このパラメータ型は、列挙データ型を必要とするプロパティまたはパラメータに使用します。</p> <p>列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。</p>

8. 初期値 フィールドに、渡されるパラメータの値を入力します。

空のままにした場合、ビジュアル テストで設定されたパラメータの初期値が使用されます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。


9. OK をクリックします。パラメータの名前と初期値 (存在する場合) が、<<開始>> ステップの **プロパティ** ペインの選択したパラメータ型の下に表示されます。

10 他のパラメータを作成するには、必要に応じて上記のステップを繰り返します。パラメータ型のフィールドは、その型の現在定義されているパラメータの数を示します。たとえば、数値 (Long) 項目のフィールドが **3 個の項目** の場合、現在このビジュアル テストには Long データ型のパラメータが 3 つ定義されています。

 **注:** パラメータが作成されたときには、パラメータに値は割り当てられていません。パラメータに値が割り当てられるのは、別のビジュアル テストでテストが実行されたときです。パラメータを作成したビジュアル テストを再生するときに、その値を割り当てる必要があります。このシナリオでは、再生を開始したときに **パラメータ** ダイアログ ボックスが開き、パラメータに値を割り当てることができます。

ビジュアル テストでのパラメータへの値の割り当て

ビジュアル テスト間でデータの受け渡しを行うには、あるビジュアル テストが他のビジュアル テストに内からテスト ステップとして呼び出される必要があります。以降、呼び出すビジュアル テストを **ホスト** テスト、呼び出されるビジュアル テストを **子テスト** と呼びます。実行時にホスト テストから子テストにパラメータを渡すには、ホスト テストを開き、子テストのパラメータに値を割り当てます。

 **注:** ビジュアル テストでパラメータに値を割り当てる前に、パラメータをビジュアル テストに追加する必要があります。詳細については、「[ビジュアル テストへのパラメータの追加](#)」を参照してください。



ヒント: ビジュアルテスト間で渡されるパラメータの値を確認するには、結果コメントステップをビジュアルテストに追加します。コメントの**式**プロパティに対して**変数**を選択し、リストから、渡された値が格納されるパラメータを選択します。

1. ホストテストを開きます。
2. **テストステップ** ペインで、パラメータを含む子テストを再生するステップを選択します。
ステップテキストは、「ビジュアル テスト [資産名] を再生します」のようになります (ここで、[資産名] は子テストの名前です)。
ステップのプロパティが **プロパティ** ペインに表示されます。このプロパティには、ビジュアルテストパラメータとして設定されているパラメータが含まれます。パラメータは、**入力パラメータ** と **出力パラメータ** というカテゴリに分類されます。各パラメータは、データ型ごとにリストされます。各パラメータのプロパティ名がそのパラメータ名です。
パラメータが **プロパティ** ペインに表示されない場合は、子テストを再読み込みします。これを行うには、**名前** プロパティを選択し、**パラメータの更新** をクリックしてパラメータのリストを更新します。
3. パラメータを選択します。
4. **パラメータの編集** をクリックします。**パラメータの編集** ダイアログ ボックスが開きます。
5. パラメータの値を **初期値** フィールドに入力します。
6. 他のパラメータに値を割り当てるには、上記のステップを繰り返します。

ビジュアルテストのパラメータの編集

1. パラメータを作成するビジュアルテストを開きます。
2. **タスク** ペインで **<<開始>>** ステップをクリックします。
3. **プロパティ** ペインで、**変数** カテゴリに移動します。
4. 編集するパラメータの種類をに移動します。
 - 入力パラメータを編集する場合は、**入力パラメータ** に移動します。
 - 出力パラメータを編集する場合は、**出力パラメータ** に移動します。
5. 編集するパラメータを選択します。
6. **パラメータの編集** をクリックします。**パラメータの編集** ダイアログ ボックスが開きます。
7. パラメータの名前を **名前** フィールドに入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
8. **型** リストからパラメータのデータ型を選択します。
Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

列挙

このパラメータは一連の値をグループ化し、1 から n まで順番に番号を付けます。このパラメータ型は、列挙データ型を必要とするプロパティまたはパラメータに使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

9. 初期値 フィールドに、渡されるパラメータの値を入力します。

空のままにした場合、ビジュアル テストで設定されたパラメータの初期値が使用されます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。

100K をクリックします。

ビジュアル テストからのパラメータの削除

1. 削除するパラメータが含まれているビジュアル テストを開きます。
2. **タスク** ペインで <<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**変数** カテゴリに移動します。
4. 削除するパラメータのテキスト ボックスをクリックし、**削除** をクリックします。

Silk Central からのパラメータを使用する

To enable Silk Central で テストに対して設定されたパラメータを Silk Test Workbench で使用できるようにするには、Silk Test Workbench で Silk Central のパラメータと同じ名前を持つ変数を定義し、そしてその変数を使用します。

ビジュアル テスト間でのデータの受け渡し

Silk Test Workbench では、ビジュアル テスト間でデータを受け渡すことができます。これにより、既存の Silk Test Workbench ビジュアル テストで作成されたデータを再利用できるだけでなく、テスト ソリューション内でデータの整合性を維持したまま、最も効果的なテスト資産を使用できるようになります。ビジュアル テスト間でのデータの受け渡しには、ビジュアル テストのパラメータを使用します。

ビジュアル テスト間でデータの受け渡しを行うには、あるビジュアル テストが他のビジュアル テストに内からテスト ステップとして呼び出される必要があります。以降、呼び出すビジュアル テストを **ホスト テスト**、呼び出されるビジュアル テストを **子テスト** と呼びます。ビジュアル テスト間でデータを受け渡しを行うには、ビジュアル テストが同じプロジェクトに含まれているか、子テストが参照プロジェクトに含まれている必要があります。

ビジュアル テストのパラメータを使用すると、以下の作業を実行できます。

- ビジュアル テストでデータを作成し、それを他のビジュアル テストで使用する。
- ビジュアル テストでデータを作成し、それを子テストに渡して変更し、再びホスト テストに戻す。

その手順は以下のとおりです。

1. 入力パラメータと出力パラメータを子テストに追加します。
2. 子テストをテスト ステップとしてホスト テストに挿入します。
3. ホスト テスト内で入力パラメータの値を編集することによって、ホスト テストから子テストにデータを渡します。



注: ホスト テストから送信する値は子テスト内に保存されませんが、子テストでの各種処理内で使用できます。

4. ホスト テストで、子テストの出力パラメータから渡されたデータを格納するローカル変数を作成します。

5. ホストテストでローカル変数を使用します。



注: 入力パラメータと出力パラメータを作成し、ビジュアルテストでこれらのパラメータを使用する方法の詳細な例については、「*SilkTest Workbench* ビジュアルテストチュートリアルによる」の「ビジュアルテストからのスクリプトの再生」を参照してください。

ビジュアルテスト間で受け渡すパラメータの定義

他のビジュアルテストからテストステップとしてビジュアルテストを呼び出すときに、これらのビジュアルテスト間でパラメータを受け渡すことができます。以降、呼び出すビジュアルテストをホストテスト、呼び出されるビジュアルテストを子テストと呼びます。

- ホストテストから子テストにパラメータを渡すには、子テストの入力パラメータとしてパラメータを指定する必要があります。
- 子テストからホストテストにパラメータを渡すには、子テストの出力パラメータとしてパラメータを指定する必要があります。

1. ホストテストから子テストにパラメータを渡すには、パラメータを入力パラメータとして子テストに追加します。

- a) 子テストを開きます。
- b) <<開始>> ステップを選択します。
- c) **プロパティ** ペインで、**入力パラメータ** を選択します。
- d) **パラメータの追加** をクリックします。**パラメータの追加** ダイアログボックスが開きます。
- e) **パラメータ名** フィールドに、新しいパラメータの名前を入力します。
- f) **型** リストから、ビジュアルテストに渡すデータ型を選択します。
- g) **初期値** フィールドの値は、デフォルト値のままにします。
- h) **OK** をクリックします。**プロパティ** ペインの入力パラメータのリストに、入力パラメータが表示されます。



注: ビジュアルテストに対して定義された入力パラメータは、ビジュアルテストステップには表示されません。入力パラメータは、<<開始>> ステップの**プロパティ** ペインで表示できます。



注: ホストテストから送信する値は、子テスト内に保存されませんが、子テストでの各種処理内で値を使用できます。

2. 子テストからホストテストにパラメータを渡すには、パラメータを出力パラメータとして子テストに追加します。

- a) 子テストを開きます。
- b) <<開始>> ステップを選択します。
- c) **プロパティ** ペインで、**出力パラメータ** を選択します。
- d) **パラメータの追加** をクリックします。**パラメータの追加** ダイアログボックスが開きます。
- e) **パラメータ名** フィールドに、新しいパラメータの名前を入力します。
- f) **型** リストから、ビジュアルテストから渡すデータ型を選択します。
- g) **初期値** フィールドの値は、デフォルト値のままにします。
- h) **OK** をクリックします。**プロパティ** ペインの出力パラメータのリストに、出力パラメータが表示されます。



注: ビジュアルテストに実際に宣言されている場合を除き、ビジュアルテストで定義された出力パラメータはビジュアルテストには表示されません。出力パラメータは、<<開始>> ステップの**プロパティ** ペインで表示できます。

- i) ホストテストを開きます。
 - j) 子テストからホストテストに渡すパラメータに対してローカル変数を追加します。
3. 子テストの最新のパラメータを表示するには、子テストの**再生** ステップの**プロパティ** ペインにある**名前** テキストボックスをクリックし、**パラメータの更新** をクリックします。

ビジュアルテストでの他のビジュアルテストから受け取ったパラメータの使用

あるビジュアルテストで、そこから呼び出す他のビジュアルテストの出力パラメータの値を格納できます。以降、呼び出すビジュアルテストをホストテスト、呼び出されるビジュアルテストを子テストと呼びます。ホストテストで子テストのパラメータを使用するには、子テストの出力パラメータをホストテストのローカル変数に割り当てます。

1. ホストテストを開きます。
2. ホストテストに渡す子テストの出力パラメータそれぞれに対してローカル変数を作成します。
詳細については、「[ビジュアルテストへのローカル変数の追加](#)」を参照してください。
3. **テストステップ** ペインで、子テストを実行するステップを選択します。ステップテキストは、「ビジュアルテスト '資産名' を再生します」のようになります (ここで、'資産名' は子テストの名前です)。
4. ステップの **プロパティ** ペインで、**出力パラメータ** カテゴリを展開します。
子テストの出力パラメータの名前は、**出力パラメータ** サブカテゴリに表示されます。
パラメータが **プロパティ** ペインに表示されない場合は、**名前** テキストボックスをクリックして **パラメータ名の更新** をクリックします。
5. 出力パラメータ名のテキストボックスをクリックし、リストから適切なローカル変数を選択します。
6. ホストテストに渡す子テストの出力パラメータそれぞれに対して前の手順を繰り返します。
7. 省略可能 : 子テストからホストテストに渡される出力パラメータの値を確認する場合は、各パラメータに対して結果コメントステップをホストテストに挿入します。
 - a) ホストテストで、結果コメントを挿入する前のステップを右クリックします。
たとえば、子テストを呼び出すステップをクリックします。
 - b) **挿入 > 結果コメント** を選択します。
 - c) 新しいステップのプロパティで、**式** テキストフィールドをクリックします。
 - d) **選択** をクリックします。
 - e) リストから **変数** を選択します。**変数の選択** ダイアログボックスが開きます。
 - f) 子テストの出力パラメータに対応する変数を選択します。
8. ローカル変数を使用するように、ホストテストの適切なテストステップを変更します。
 - a) **テストステップ** ペインでステップを選択します。
 - b) **プロパティ** ペインで、**パラメータ** セクションを展開します。
 - c) ローカル変数のプロパティの種類に対応したテキストボックスをクリックします。
ローカル変数がテキスト型の場合は、たとえば、AUT のテキストフィールドにテキストを入力するテストステップ (typeKeys) を選択し、**keys** テキストボックス (テキスト型) をクリックします。
 - d) **選択** をクリックして、**変数** を選択します。**変数の選択** ダイアログボックスが開きます。
 - e) 使用するローカル変数を選択し、**OK** をクリックします。

再生するホストテストに子テストが含まれていて、その入力パラメータと出力パラメータが正しく定義されていれば、ホストテストを再生すると、ホストテストと子テストの間でパラメータが正しく受け渡されます。



注: 入力パラメータと出力パラメータを作成してビジュアルテストで使用する方法の詳細な例については、「[SilkTest Workbench ビジュアルテストチュートリアル](#)による」の「[ビジュアルテストからのスクリプトの再生](#)」を参照してください。

テストフローの管理

ビジュアルテストの再生中、通常、ステップは Silk Test Workbench によって順番に実行されます。ただし、ステップを順番に実行しない方が便利な場合もあります。たとえば、ビジュアルテストを完了する前に、ビジュアルテスト内の複数のポイントで一連のステップを実行する必要がある場合などです。テスト内の複数の場所で一連のステップを繰り返す代わりに、テストの最後の一連のステップを配置して、各ポイントからそこに移動することができます。

特定のステップへの移動は、一定の条件が満たされた場合に繰り返される繰り返しロジックを終了する手段としても使用できます。

ビジュアルテスト内でラベルを使用して特定の領域を「ブックマーク」すると、テストのナビゲーションに使用できます。ビジュアルテストで定義された任意のラベルに移動するテストステップを作成できます。ラベルステップはビジュアルテスト内での単純なマーカーであるため、**テストステップ** ペインに表示される後続の一連のステップを説明する手段としても使用できます。

ビジュアルテスト内でのスクリプトの実行

ビジュアルテストでは、既存の .NET スクリプトを呼び出して実行できます。これにより、アプリケーションのテストに対するモジュール式アプローチが可能になり、テストの実行をより制御できるようになります。ビジュアルテスト内でスクリプトを実行する場合は、テストサイトの整合性を保ち、テスト対象のテストアプリケーションが正しい初期再生状態になるようにすることが重要です。

この手順を使用してドライバテストを作成することもできます。

1. スクリプトを挿入するビジュアルテストを開きます。
2. 挿入したスクリプトを実行するポイントの前のテストステップを選択します。
3. **挿入 > .NET スクリプト** を選択します。 **.NET スクリプトの参照** ダイアログ ボックスが開きます。
4. 挿入するスクリプトを選択し、**OK** をクリックします。Silk Test Workbench によって、選択したステップの下にステップが挿入されます。挿入されたステップによって、選択したスクリプトが呼び出され、再生されます。ステップテキストは、次のようになります：

```
.NET スクリプト '資産名' を再生します
```

ここで、'資産名' は、挿入されたスクリプトの名前です。

再生中、前のステップが実行されると、ビジュアルテスト内の次のステップが実行される前に、挿入されたスクリプトが最後まで再生されます。スクリプトがビジュアルテストに挿入されたあと、スクリプトを編集して、再生に使用される設定を構成できます。

テスト対象アプリケーションは、必要に応じて設定、フック、開始され、再生セッションの終了時までフックされたままになります。たとえば、ビジュアルテストにメモ帳用のアプリケーション構成があり、.NET スクリプトに電卓用のアプリケーション構成がある場合には、次のようになります。

- ビジュアルテストの実行の開始時に、メモ帳がフックされ、開始されます。
- 挿入された .NET スクリプトの実行の開始時に、電卓がフックされ、開始されます。メモ帳はフックされたままです。
- 挿入された .NET スクリプトの実行を停止しても、メモ帳と電卓の両方ともフックされたままになります。
- 再生セッションの終了時、つまり、ビジュアルテストの実行を停止すると、メモ帳と電卓の両方のフックが解除されます。

ビジュアルテストに挿入されたスクリプトの変更

スクリプトがビジュアルテストに挿入されたあと、スクリプトを編集して、再生に使用される設定を構成できます。

1. 変更するスクリプトを開き、変更し、保存します。
たとえば、スクリプトの入力パラメータまたは出力パラメータを変更する場合があります。
2. 変更したスクリプトを挿入したビジュアルテストを開きます。
3. **テストステップ** ペインで、スクリプトを実行するビジュアルテスト内のステップを選択します。ステップテキストは、「.NET スクリプト '資産名' を再生します」のようになります (ここで、資産名は挿入されたスクリプトの名前です)。

ステップのプロパティが **プロパティ** ペインに表示されます。ここには、スクリプトの入力パラメータと出力パラメータが表示されます。

4. ステップの **プロパティ** ペインで、**.NET スクリプト** カテゴリを展開します。

挿入されたスクリプトの名前が **名前** 行に表示されます。

- ビジュアル テストに挿入されているスクリプトを変更するには、**.NET スクリプトの参照** をクリックし、使用するスクリプトを選択します。
- パラメータ名のみを更新するには、**パラメータ名の更新** をクリックし、プロンプトが表示されたら **はい** をクリックします。
- 使用可能なパラメータを再読み込みするには、**パラメータの更新** をクリックし、プロンプトが表示されたら **はい** をクリックします。
- 再生設定を変更するには、**再生設定** リストから別の値を選択します。

ビジュアル テスト内でのメッセージ ボックスの作成

ビジュアル テストでは、再生中にテスト担当者に対してメッセージを表示できます。たとえば、メッセージ ボックスを使用して、テスト出力を手動で検証したり、コントロールに関する情報を表示したり、テスト中にその他の任意の情報を通知したりできます。

メッセージ ボックスを作成するには、ビジュアル テストにメッセージ ボックス ステップを挿入します。再生中、メッセージ ボックスが表示され、ステップの作成時に指定したメッセージの内容が示されます。

- メッセージを挿入するビジュアル テストを開きます。
- メッセージ ボックスを表示するポイントの前のステップを選択します。
- 挿入 > メッセージ ボックス** を選択します。 Silk Test Workbench によって、選択したステップの下にメッセージ ボックスステップが挿入されます。
- メッセージ ボックスステップの **プロパティ** ペインで、アイコンをクリックして、プロパティをカテゴリごとにグループ化するか、アルファベット順に並べ替えます。
- プロパティ** ペインで **全般** カテゴリのプロパティを編集し、メッセージ ボックスのタイトルとテキストを指定します。
- プロパティ** ペインで **詳細設定** カテゴリのプロパティを編集し、メッセージ ボックスの表示要素（ボタン、アイコンなど）を指定します。

ビジュアル テスト ナビゲーション用のラベル テスト ステップの作成

ラベル テスト ステップでは、テスト操作は実行されません。プログラム フロー管理用のラベル テスト ステップを使用して、「移動」ステップの移動先として使用すると共に、後続の一連のステップを記述します。

- ラベルを挿入するビジュアル テストを開きます。
- ラベル ステップが必要なポイントの前のステップを選択します。
- テスト ステップ** ペインで、**操作 > 挿入 > ラベル** をクリックします。選択したステップのあとにラベル ステップが作成されます。ステップの **プロパティ** ウィンドウには、ラベル ステップの **ラベル** プロパティが表示されます。
- このラベル ステップの **ラベル** プロパティの値領域をクリックして、ラベルの説明テキストを入力します。

説明テキストを使用して、ラベル ステップの用途を示します。ラベルを「移動」ステップの移動先として使用する場合は、プログラム フローがこのステップに分岐する理由を示すテキストを入力します。ラベルを使用して後続の一連のステップを説明する場合は、後続の一連のステップの目的を示すテキストを入力します。

ラベル プロパティの値を変更できます。ただし、そのラベル ステップを参照するステップ（ロジック ステップ、「移動」ステップなど）もすべて手動で更新して、ラベル ステップの新しい値を参照するようになる必要があります。

- 保存** をクリックして、ラベル ステップとその **ラベル** プロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアル テストに表示されます。
- ラベル ステップをナビゲーション用を使用する場合は、必要に応じて、そのラベル ステップに移動する「移動」ステップを作成します。

ビジュアル テスト ナビゲーション用の移動テスト ステップの作成

移動テスト ステップでは、テスト操作は実行されません。プログラム フロー管理用の移動テスト ステップを使用して、テストの再生中に指定したラベルのステップに移動します。

1. 移動ステップを挿入するビジュアル テストを開きます。
2. 移動ステップを挿入するポイントの前のステップを選択します。
3. **挿入** > **プログラム フロー** > **移動** を選択します。選択したステップのあとに移動ステップが作成されます。ステップの **プロパティ** ウィンドウには、ラベルステップの **ラベル** プロパティが表示されます。
4. 移動ステップの **ラベル** プロパティの値領域をクリックして、リストから移動先のラベルを選択します。



ヒント: 移動ステップの移動先としてラベル ステップを使用するには、最初にラベル ステップを作成します。

5. **保存** をクリックして、移動ステップとその **ラベル** プロパティの値を保存します。
更新されたステップおよびその説明テキストがビジュアル テストに表示されます。

ビジュアル テスト内での外部アプリケーションの起動

ビジュアル テストでは、再生中に外部アプリケーションを起動できます。これにより、テスト アプリケーションが予想どおり起動するかどうかをテストしたり、起動中のアプリケーションをテストするときのテストサイトの状態を保つことができます。

1. 外部アプリケーションを起動するビジュアル テストを開きます。
2. アプリケーションを起動するポイントの前のステップを選択します。
3. **挿入** > **外部プログラム** を選択します。Silk Test Workbench によって、選択したステップの下にプログラムの実行 ステップが挿入されます。
4. 新しく作成されたプログラムの実行 ステップを選択して、**プロパティ** ペインにそのプロパティを表示します。**コマンド** カテゴリのプロパティが **プロパティ** ペインに表示されます。
5. **パス名** テキスト ボックスで、以下のいずれかのステップを実行します。
 - 起動する外部プログラムのパスを入力します。
 - **ファイルの選択** をクリックして [ファイルの選択] ダイアログ ボックスを開き、起動する外部プログラムに移動します。
 - **選択** をクリックし、外部プログラムを起動するために使用するリテラル値、変数、式、またはアクティブ データを指定します。



注: 実行する操作が **検索** または **探索** の場合、このプロパティではパス名のみが必要です。

6. **操作** リスト ボックスから、アプリケーションの起動時に実行する操作を選択します。
以下のいずれか 1 つを選択できます。
 - **開く** (デフォルト) : 指定したパラメータを使用して、指定ファイルを開きます。ファイルには、アプリケーション、ドキュメント ファイル、またはフォルダを指定できます。
 - **編集** : エディタを起動して、編集するファイルを開きます。**パス名** プロパティで指定されたファイルが、ドキュメント エディタで開くことのできるドキュメント ファイルではない場合、この操作は実行されません。
 - **探索** : Windows エクスプローラを使用して、**パス名** プロパティで指定されたフォルダを開きます。
 - **検索** : 指定されたディレクトリを起点として、Windows エクスプローラを検索モードで起動します。ディレクトリは、**パス名** プロパティまたは **ディレクトリ** プロパティを使用して指定できます。**パス名** プロパティが使用され、かつ、ファイル名が含まれている場合は、テストを再生しても、このステップは実行されません。
 - **印刷** : **パス名** プロパティで指定されたドキュメント ファイルを印刷します。**パス名** プロパティで指定されたファイルがドキュメント ファイルではない場合、テストを再生してもこのステップは実行されません。

7. **パラメータ** テキスト ボックスで、起動のためにアプリケーションに渡すパラメータを指定します。
パス名 プロパティで指定されたファイルがアプリケーション ファイルの場合は、このプロパティを使用して、アプリケーションに渡して起動時に使用するパラメータを含む NULL 終端文字列を指定します。**パス名** プロパティで指定されたファイルがドキュメント ファイルまたはその他のタイプのファイルの場合は、値を空白のままにします。
8. **ディレクトリ** テキスト ボックスで、**操作** プロパティで指定された操作に使用できるデフォルト ディレクトリを指定します。
9. **ウィンドウ スタイル** リスト ボックスから、起動時のアプリケーションまたはプログラムの表示方法を選択します。
 値は以下のとおりです。
 - **非表示**：プログラムまたはアプリケーションが起動したとき、ウィンドウは非表示です。フォーカスは、他のウィンドウに渡されます。
 - **最大化**：アプリケーション ウィンドウを最大化します。
 - **最小化**：起動時にアプリケーション ウィンドウを最小化し、Z オーダーで次に最上位にくるウィンドウをアクティブにします。
 - **復元**：アプリケーション ウィンドウを起動して表示します。ウィンドウが最小化または最大化されていた場合は、そのウィンドウの元のサイズと位置に復元されます。この値は、最小化されたウィンドウを復元するときに使用します。
 - **表示**：アプリケーション ウィンドウを起動し、現在のサイズと位置に表示します。
 - **デフォルトの表示**：アプリケーションを起動したプログラムからの情報に基づいて、ウィンドウの表示状態を設定します。
 - **表示の最大化**：アプリケーションを起動し、ウィンドウを最大化して表示します。
 - **表示の最小化**：アプリケーションを起動し、ウィンドウを最小化して表示します。
 - **表示を最小化し、アクティブ化しない**：最小化されたウィンドウでアプリケーションを表示します。現在のアクティブ ウィンドウは、アクティブなままです。
 - **現在の状態で表示し、アクティブ化しない**：現在のウィンドウ状態でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
 - **最新の位置に表示し、アクティブ化しない**：最後に使用したときのウィンドウ サイズと位置でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
 - **通常表示**：アプリケーションを起動して、ウィンドウを表示します。最後に使用したウィンドウの位置と最小化または最大化の状態が復元されます。この値は、はじめてウィンドウを表示するときに使用します。

ビジュアル テストへのオブジェクトの待機ステップの挿入

ビジュアル テストの任意の場所にステップを挿入して、ビジュアル テストの再生中にオブジェクトが存在するようになるまで待機したり、オブジェクトが消滅するまで待機したりできます。ビジュアル テストの再生とテスト アプリケーションとの間で同期を取るために、オブジェクトの待機ステップを組み込みます。

1. オブジェクトの待機ステップを組み込むビジュアル テストを開きます。
2. 以下のいずれかのステップを実行します。
 - オブジェクトの待機ステップを実行する場所の直前のステップを選択し、**挿入 > 同期とタイミング > オブジェクトの待機** を選択します。
 - **ロジック ツールボックス** を開き、**このビジュアル テストの再生中に出現/消滅するオブジェクトを待機** を **テスト ステップ** ペインの適切な位置にドラッグします。

Silk Test Workbench によって、選択したステップの下に " が存在するまで待機します ステップが挿入されます。
3. 新しく作成された " が存在するまで待機します ステップを選択して、**プロパティ** ペインの **待機** カテゴリにそのプロパティを表示します。
4. **プロパティ** ペインの **ロケーター** テキスト ボックスで、待機するオブジェクトを識別します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
 - **選択**：このボタンをクリックして、リテラル、変数、式の結果、またはアクティブ データ値を割り当てます。
5. **待機の種類** リストから、オブジェクトの存在を待機するか消滅を待機するかを指定します。
 6. **タイムアウト** テキスト ボックスで、オブジェクトが存在または消滅するまで待機する最大時間をミリ秒で入力します。
デフォルトでは、この値は **オプション** ダイアログ ボックスの **デフォルトの待機タイムアウト (ミリ秒)** オプション値に設定されます。タイムアウト期間内に一致するオブジェクトがない場合は、再生エラーが発生します。
 7. **保存** をクリックします。

プロパティの待機

テスト対象アプリケーションのコントロールのプロパティが、特定の値になるまで待機するテスト ステップをビジュアル テストに追加できます。このようなステップは、ビジュアル テストの再生とテスト アプリケーションとの間で同期を取るために使用します。

1. ステップを追加するビジュアル テストを開きます。
2. ステップを挿入する場所の直前のステップを右クリックし、**挿入 > 同期とタイミング > プロパティの待機** を選択します。Silk Test Workbench によって、選択したステップの下に新しいステップが挿入されます。
3. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
4. **ロケータ** フィールドには、待機するプロパティを含む AUT のコントロールに対するロケータを指定します。
5. **プロパティ** フィールドで、プロパティを選択します。
6. **値** フィールドに、待機する値を指定します。
7. **タイムアウト** フィールドには、再生の続行を待機する最大時間を指定します。
8. **保存** をクリックして、ステップとそのプロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアル テストに表示されます。

ビジュアル テストの再生遅延の設定

ビジュアル テストの任意の場所に、ビジュアル テストの再生を遅延させるための遅延ステップを挿入できます。遅延の時間を制御できます。ビジュアル テストに挿入できる遅延の数に制限はありません。ビジュアル テストの再生とテスト アプリケーションとの間で同期を取るために遅延を使用します。

1. 再生遅延を設定するビジュアル テストを開きます。
2. 遅延ステップを実行するポイントの前のステップを選択します。
3. **挿入 > 同期とタイミング > 遅延** を選択します。Silk Test Workbench によって、選択したステップの下に 1 秒の遅延 ステップが挿入されます。
4. 遅延の増分と時間を変更するには、新しく作成された 1 秒の遅延 ステップを選択し、そのプロパティを **プロパティ** ペインに表示します。
 - a) **遅延タイプ** リストから、遅延間隔に対して **秒** または **ミリ秒** を選択します。
 - b) **遅延時間** テキスト ボックスで、遅延の時間を指定します。
リテラル値を入力するか、あるいは **選択** をクリックして変数、式の結果、またはアクティブ データ値を割り当てます。

5. **保存** をクリックします。

ビジュアル テストへのタイマーの挿入

ビジュアル テストの任意の場所にタイミング ステップを挿入して、再生中にタイマーを開始、停止、または再開できます。タイマーを組み込むと、特定のステップを再生するのにかかる時間を特定できます。

1. 開始タイマー、停止タイマー、または再開タイマーを組み込むビジュアル テストを開きます。
2. タイマーを挿入する場所の直前のステップを選択し、**挿入 > 同期とタイミング** を選択して、以下のコマンドのいずれかを選択します。
 - **開始タイマー**：時間再生する一連のステップの最初のステップを指定するには、このオプションを選択します。
 - **停止タイマー**：時間再生する一連のステップの最後のステップを指定するには、このオプションを選択します。
 - **再開タイマー**：一連のステップのタイミングの設定を再開するには、このオプションを選択します。たとえば、いくつかのテスト ステップの時間を記録することが必要でない場合は、停止タイマーを挿入し、次に、タイミングに含めるステップに対して再開タイマーを挿入できます。

Silk Test Workbench によって、選択したステップの下にタイマー番号 n の開始、タイマー番号 n の停止、またはタイマー番号 n の再開ステップが挿入されます。

3. 新しく作成されたステップを選択し、そのプロパティを **プロパティ** ペインの **タイマー** カテゴリに表示します。
4. **タイマーの動作** リストから、使用するタイマーの種類を選択します。
5. **タイマー番号** テキスト ボックスで、タイマーに割り当てる番号を入力するか、リストから選択します。複数の条件の時間を計るときに、各タイマーに一意の番号を割り当ててタイマーを区別できます。ビジュアル テストには、最大で 10 個のタイマーを組み込むことができます。
6. **保存** をクリックして、ステップとそのプロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアル テストに表示されます。

ビジュアル テストを再生すると、**結果** ウィンドウの **詳細** タブで、テスト ステップの **結果の詳細** 列にタイマーの結果が表示されます。

テスト ステップ

ステップ (またはテスト ステップ) は、ビジュアル テストにおける主な実行可能単位です。ステップは、スクリプトのコード行に類似しています。

テスト ステップは、**ビジュアル ナビゲータ** の **テスト ステップ** ペインでのみ実行できます。

ビジュアル テストのステップ フローとロジックは、スクリプトのコード行のフローとロジックに似ています。たとえば、ビジュアル テストでは、コントロールに対して操作を実行する前に、テスト アプリケーションのそのコントロールにアタッチする必要があります。スクリプトと同様、ビジュアル テストにも、テスト アプリケーションのコントロールを表すオートメーション オブジェクトが含まれています。このオートメーション オブジェクトによって、コントロールに対して実行可能なプロパティとメソッドが公開されます。オートメーション操作の 1 つ 1 つが、ビジュアル テストの 1 ステップとして表現されます。

複数のステップの選択

ビジュアル テストのステップをグループとして選択すると、そのステップのグループごとに情報を更新できます。

1. 連続したテスト ステップのグループを選択するには、以下のステップを実行します。
 - a) 連続したステップのグループで最初のテスト ステップを選択します。
 - b) Shift キーを押したまま、連続したグループの最後のステップを選択します。または、**編集 > すべて選択** を選択して、ビジュアル テストのすべてのステップを選択します。

2. 連続していないテスト ステップのグループを選択するには、以下のステップを実行します。

- a) ステップのグループで最初のテスト ステップを選択します。
- b) Ctrl キーを押したまま、任意にステップを選択します。ステップが連続している必要はありません。

テスト ステップのコピーと貼り付け

<<開始>> ステップと <<終了>> ステップ以外の任意のテスト ステップは、開いているビジュアル テストにコピーして貼り付けできます。テスト ステップをコピーおよび貼り付ける場合は、以下の点を考慮してください。

- Silk Test Workbench では、テスト ステップは、選択した順序ではなく、ビジュアル テストに表示されている順序でクリップボードにコピーされます。たとえば、テスト ステップ 5 を選択してからテスト ステップ 2 を選択し、その両方をクリップボードにコピーした場合は、ビジュアル テストにテスト ステップを貼り付けると、Silk Test Workbench によってテスト ステップ 2、テスト ステップ 5 の順に貼り付けられます。
- 画面テスト ステップをコピーしても、関連するテスト ステップはコピーされません。
- ラベルテスト ステップをコピーして貼り付ける場合は、ラベル テスト ステップが必ず一意になるように Silk Test Workbench によって「_1」が追加されます。
- 繰り返しテスト ステップまたは繰り返しの終了テスト ステップをコピーすると、関連する繰り返しの終了テスト ステップまたは繰り返しテスト ステップもコピーされます。
- If テスト ステップまたは End If テスト ステップをコピーすると、関連するすべての判断ロジック テスト ステップ (If、Else、Else If、End If) がコピーされます。ただし、判断ロジック テスト ステップに含まれる依存テスト ステップはコピーされません。
- 複数の Else テスト ステップまたは Else If テスト ステップは、すべての判断ロジック テスト ステップ (If、Else、Else If、End If) をコピーしないかぎり、クリップボードにコピーできません。Else If テスト ステップを貼り付ける場合、Silk Test Workbench で貼り付けできるのは、正しいコンテキスト内にあるステップのみです。

1. **テスト ステップ** ペインでビジュアル テストを開き、コピーするステップを選択します。

連続した、または連続していない複数のステップを選択して、そのステップを連続したグループとしてコピーし、貼り付けることができます。

2. **編集 > コピー** を選択します。

コピーしたステップを別のビジュアル テストに貼り付けるには、そのビジュアル テストを開き、コピーしたステップを貼り付けます。コピーではなく、選択したステップを移動するには、**編集 > 切り取り** を選択します。

3. コピーしたステップを貼り付ける場所の直前のテスト ステップを選択します。

4. **編集 > 貼り付け** を選択します。Silk Test Workbench によって、ビジュアル テストの選択したステップのあとに、コピーしたテスト ステップが貼り付けられます。

画面プレビューを使用したテスト ステップの削除

画面プレビュー 内のキャプチャされたテスト アプリケーションのイメージは、テスト ステップに関連付けられているため、キャプチャされたイメージを **画面プレビュー** で削除すると、関連付けられているテスト ステップも **テスト ステップ** ペインから削除できます。次のテスト ステップが関連します。

- 関連付けられているすべての自動テスト ステップ
- その他のタイプのすべての単一ノード テスト ステップ
- すべての検証テスト ステップ

削除するキャプチャ イメージに関連付けられた判断および繰り返しロジックのテスト ステップは、**画面プレビュー** から削除することはできません。これにより、ロジックが保持され、ビジュアル テストが正しく再生されます。判断および繰り返しテスト ステップは、**テスト ステップ** ペインから手で直接削除する必要があります。

1. **テストステップ** ペインで、キャプチャされたイメージのテストステップから、削除するステップを選択します。**画面プレビュー** に、キャプチャされたイメージが表示されます。
2. キャプチャされたイメージを **画面プレビュー** で選択します。
3. Del を押します。サムネイルとそれに関連するステップの削除を確認するメッセージが表示されます。
4. 次のいずれか 1 つのステップを行います：
 - **テストステップ** ペインで、キャプチャされたイメージとそれに関連するテストステップを削除するには、**はい** をクリックします。
 - **画面プレビュー** のキャプチャされたイメージのみを削除し、関連するテストステップを保持するには、**いいえ** をクリックします。
 - **キャンセル** をクリックして、サムネイルを削除することなくメッセージを閉じます。

[テストステップ] ウィンドウを使用したテストステップの削除

<<開始>> ステップと <<終了>> ステップ以外のテストステップは、**テストステップ** ペインから削除できます。ただし、テストステップによっては、正しく実行できるかどうかは他のテストステップに依存するため、以下のルールが適用されます。

- ロジックの種類が If のテストステップを削除すると、削除された If テストステップに関連付けられているすべての Else If、Else、または End If テストステップも削除されます。
- ロジックの種類が End If のテストステップを削除すると、削除された End If テストステップに関連付けられているすべての If、Else、または Else If テストステップも削除されます。
- ロジックの種類が Else または Else If のテストステップを削除すると、Else または Else If テストステップのみが削除されます。If テストステップと End If テストステップは削除されません。
- ロジックの種類が 繰り返し のテストステップを削除すると、繰り返しの終了テストステップも削除されます。
- 画面にアタッチするテストステップまたは自動ステップのテストステップを削除すると、判断ロジックと繰り返しロジックのテストステップを除き、関連するステップがすべて削除されます。アタッチステップに関連付けられた自動ステップは削除されます。ただし、関連付けられた操作ステップの前に、同じ画面にアタッチする別のアタッチステップが存在する場合は、削除されません。

1. **テストステップ** ペインでビジュアルテストを開き、削除するテストステップを右クリックして **削除** を選択します。
連続した、または連続していない複数のステップを選択して、そのステップを連続したグループとして削除できます。
2. テストステップのいずれかが画面タイプのステップの場合は、画面項目とそれに関連するステップの削除を確認するメッセージが表示されます。
 - 選択したステップとそれに関連するステップを削除するには、**はい** をクリックします。
 - 選択したステップのみを削除し、関連するテストステップを削除しない場合は、**いいえ** をクリックします。
 - テストステップを削除しないでメッセージを閉じるには、**キャンセル** をクリックします。

ストーリーボードを使用したテストステップの削除

サムネイルはテストステップのグループと関連付けられているため、**ストーリーボード** を使用してサムネイルを削除すると、関連するテストステップも **テストステップ** ペインから削除することができます。次のテストステップが関連します。

- 関連付けられているすべての自動テストステップ
- その他のタイプのすべての単一ノードテストステップ
- すべての検証テストステップとイベントロジックテストステップ

削除するキャプチャイメージに関連付けられた判断および繰り返しロジックのテストステップは、**画面プレビュー** から削除することはできません。これにより、ロジックが保持され、ビジュアルテストが正しく


再生されます。判断および繰り返しテストステップは、**テストステップ** ペインから手動で直接削除する必要があります。

1. **ストーリーボード** で、削除するテストステップに対応するサムネイルをクリックします。
2. Del を押します。サムネイルとそれに関連するステップの削除を確認するメッセージが表示されます。
3. 次のいずれか 1 つのステップを行います：
 - **テストステップ** ペインで、サムネイルとそれに関連するテストステップを削除するには、**はい** をクリックします。
 - **ストーリーボード** のサムネイルのみを削除し、関連するテストステップを保持するには、**いいえ** をクリックします。
 - **キャンセル** をクリックして、サムネイルを削除することなくメッセージを閉じます。

ビジュアルテストの比較

2 つの異なるビジュアルテストや、同じビジュアルテストの 2 つのバージョンを比較して、これらの間の相違点を調べることができます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **ビジュアルテスト** を選択します。選択したプロジェクトの既存のビジュアルテストが、右側のペインに表示されます。

 **ヒント:** 調査したいビジュアルテストが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2 つの異なるビジュアルテストを比較するには：
 - a) **Ctrl + クリック** を使用して、比較する 2 つのビジュアルテストを選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
4. 同じビジュアルテストの 2 つのバージョンを比較するには：
 - a) 調査したいビジュアルテストを右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。
 - 変更箇所は黄色で表示されます。
 - 追加箇所は緑色で表示されます。
 - 比較した資産の一方に存在しない項目はグレイで表示されます。

スクリプトの拡張

このセクションでは、VB .NET スクリプトを変更して、より高度な機能を実行する方法について説明します。

スクリプト内からのスクリプトの参照

よく使用される関数のスクリプトを作成し、別のスクリプトからこのスクリプトを再利用することができます。そのためには、**.NET スクリプト参照の追加** コマンドを使用します。

スクリプトの参照の追加

1. 別のスクリプトからの再利用または参照する単一のスクリプトを作成します。
2. 再利用されるスクリプトを呼び出す起動元スクリプトを新規作成します。
3. **プロパティ** タブを展開します。

4. 右クリックして、**.NET スクリプト参照の追加** を選択します。
5. 使用するスクリプトを選択して、**OK** をクリックします。

スクリプトの例

次のサンプルには、別のスクリプトから参照される、*Calculator* というスクリプトが含まれています。



注: 次のサンプルには `Main()` メソッドが含まれていないため、独自に実行することはできません。

```
Public Class Calculator
    Public Shared Function Add(left As Integer, right As Integer)
        Return left + right
    End Function

    Public Shared Function Subtract(left As Integer, right As Integer)
        Return left - right
    End Function

    Public Shared Function Multiply(left As Integer, right As Integer)
        Return left * right
    End Function

    Public Shared Function Divide(left As Integer, right As Integer)
        Return left / right
    End Function
End Class
```

別のスクリプトを作成し、参照として *Calculator* を追加して (プロパティ タブを展開し、右クリックして、**.NET スクリプト参照の追加** を選択して、**Calculator** を選択)、次のコードを使用します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        MsgBox("2 + 3 = " + Calculator.Add(2, 3).ToString())
    End Sub
End Module
```

記録中のスクリプトへの検証の追加

以下の操作を行って、スクリプトの記録中に検証を追加します。

1. 記録を開始します。
2. 検証するオブジェクトの上にマウスカーソルを移動して、**Ctrl+Alt** を押します。
モバイル Web アプリケーションを記録する場合は、オブジェクトをクリックして **検証の追加** をクリックすることもできます。
このオプションを実行すると、記録が一時的に停止され、**テスト ロジック デザイナ** ウィザードが表示されます。
3. ウィザードが開いたあと、検証するオブジェクトを選択します。複合プロパティ (**Position** など) を使用してフィールドを選択した場合は、以下のコードに類似したコードがスクリプトに追加されます：
`Workbench.Verify(New TextPosition(1, 0), .TextField("TextField").Position)`。
4. ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。

コーディングによるスクリプトへの検証の追加

スクリプトの結果が予想された結果と一致するかどうかを判断するために、検証を挿入します。

1. 検証を組み込むスクリプトを開きます。

2. 以下の検証メソッドのいずれかを挿入します。

- 期待値と実際の値を比較するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object)
```

たとえば、`Workbench.Verify("red", "red")` は検証に合格します。一方、`Workbench.Verify("red", "green")` は検証に失敗し、メッセージ「実際：[green]; 予想： [red]」が表示されます。

- 期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object, comment As String)
```

たとえば、`Workbench.Verify("red", "green", "checking colors")` は検証に失敗し、メッセージ「checking colors - 実際：[green]; 予想： [red]」が表示されます。

- 期待値と実際の値を比較して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object, comment As String, verifyFlags As VerifyFlags)
```

たとえば、`Workbench.Verify("red", "green", "checking colors", verifyFlags.ScreenShotOnFailure)` は検証に失敗し、メッセージ「checking colors - 実際：[green]; 予想： [red]」が表示され、スクリーンショットが結果ファイルに追加されます。

- 予想結果によって返された値を検証するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean)
```

たとえば、`Workbench.Verify(True)` は検証に合格します。一方、`Workbench.Verify(False)` は検証に失敗します。

- 予想結果によって返された値を検証して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean, comment As String)
```

たとえば、`Workbench.Verify(True, "Test Passed")` は検証に合格します。一方、`Workbench.Verify(False, "Test Failed")` は検証に失敗します。

- 予想結果によって返された値を検証して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean, comment As String, verifyFlags As VerifyFlags)
```

たとえば、`Workbench.Verify(True, "Test Passed", verifyFlags.ScreenShotOnFailure)` は検証に合格し、スクリーンショットは追加されません。一方、`Workbench.Verify(False, "Test Failed", ScreenShotOnFailure)` は検証に失敗し、スクリーンショットが結果ファイルに追加されます。

- IEnumerable オブジェクト（リストや配列など）について実際の値と期待値を比較するには、以下のように入力します。


```
Workbench.Verify(expectedEnumerable, actualEnumerable)
```


例：

```
Dim selectedItemsList = listBox.SelectedItems ' we assume that a list with the
items "red" and "blue" is returned
Dim expectedItemsList = New List(Of String)()
expectedItemsList.Add("red")
expectedItemsList.Add("blue")
```

```
Workbench.Verify(selectedItemsList, expectedItemsList) ' verification passes
```

```
Dim expectedItemsArray = New String() { "red", "blue" }
Workbench.Verify(selectedItemsList, expectedItemsArray) ' verification passes
```

 **注:** 2つの IEnumerable オブジェクトは、要素の数が同じであり、それらの要素が等しく、かつ順序が同じである場合に、等しいとみなされます。

 **注:** 浮動小数点数を含む数学演算では、2つの数値が、ユーザーの視点では等しいにもかかわらず、内部表現のために完全には同一ではないとみなされる場合があります。このため、浮動小数点数 (単精度、倍精度) は、差異が 0.00001 未満である場合は等しいとみなされます。この値は、状況によっては正しくない場合があります。その場合は、代わりに `Verify(result As Boolean)` を使用して 2つの値を比較します。

スクリプトでのテスト データの再利用

スクリプトで使用できる再利用の種類を示します。

スクリプトにおける変数

Silk Test Workbench ではスクリプト言語として VB.NET が使用されるため、スクリプトで VB.NET 変数を使用できます。変数は名前を付けられたエイリアスで、再生中に使用および変更が可能なデータを含んでいます。

記録時に、コントロールに対するデータ入力はリテラル データとして記録されます。たとえば、WPF TextBox コントロールに対して入力された値がある場合、その結果のコードは以下のようになります。

```
.WPFTextBox("@automationId='CarType').SetText("Ford")
```

これは機能しますが、スクリプトでは TextBox コントロールのデータとして *Ford* という値以外を使用できません。ただし、*Ford* をそのデータを表す変数で置き換えた場合、変数内でどのようなデータが使用されても、それが TextBox コントロールに入力されます。

```
.WPFTextBox("@automationId='CarType').SetText(sCarmake)
```

変数を使用すると、スクリプトで使用するデータが定数である必要がなくなるため、テストの柔軟性が高まります。

型を指定しないで変数を宣言した場合、Silk Test Workbench によって、型が指定されなかったことを示すメッセージが生成されます。Silk Test Workbench では、その場合 Object データ型がデフォルトで使用されます。許容される任意の VB.NET データ型を変数データ型として使用できます。スクリプトで VB.NET 変数を使用するときには、VB.NET 変数の宣言に関するルールが適用されます。

例：

```
Dim dSalesPrice As Double
```

```
Dim iQuantity As Integer
```

```
Dim sCarmake As String
```

Silk Test Workbench でコントロールへのデータ入力が記録される時、記録された値が二重引用符 ("") で囲まれている場合、通常は String データ型であり、String データ型の変数で置き換えることができます。

スクリプトにおける Public および Private 変数

関数および変数は、テスト スクリプトで Public および Private として宣言できます。Private として宣言された変数は、その変数が作成された資産に対してのみ可視化されます。このため、変数がスクリプトまたは関数で作成された場合、その変数はその資産の内部でのみ使用可能になり、資産によって呼び出されるその他の資産では使用できません。

Public として宣言された変数は、作成されたあと、すべての資産で使用可能になります。ある関数から別の関数またはあるスクリプトから別のスクリプトへ渡されるデータはいずれも、Public 変数として宣言する必要があります。ただし、Public 変数を使用すると、Public として宣言されている変数が、同時に再生される別のスクリプトで使用されるという状況が発生する可能性があります。これにより、データが破損したり、コンパイル エラーが生じたりする可能性があります。

変数を Public として宣言するのは、同じ変数名が別の資産で使用されていないことが確実にある場合に限定する必要があります。

スクリプト内の Public 変数と Private 変数は、ビジュアル テストでグローバル変数とローカル変数が使用されるのと同じように使用されます。

スクリプトでの変数の宣言

変数は、Dim、Private、または Public ステートメントを使用して宣言できます。構文は以下のとおりです。

```
Dim VariableName As [data type]
'or
Private VariableName As [data type]
'or
Public VariableName As [data type]
```

たとえば、String 型の変数を宣言し、その変数に値を割り当てるには、以下のように入力します。

```
Dim carmake As String
carmake = "Honda"
```

または、以下のコードでは String 型の変数が作成され、その変数に値が割り当てられます。変数が String 型になるのは、「Honda」が String である必要があることがコンパイラによって認識されるためです。このコードの結果は前の例と同じになります。一般的には、以下に示す例をお勧めします。

```
Dim carmake = "Honda"
```

スクリプトで型を指定しないで変数を宣言した場合、Silk Test Workbench によって、変数に対して型が指定されなかったことを示すメッセージが表示されます。デフォルトでは、変数は Object 型であるとコンパイラによって仮定されます。

スクリプト内の変数はすべて宣言することをお勧めします。変数を宣言しない場合、エラーの原因になる場合があります。たとえば、スペルを誤ると 1 つしかないはずの変数が 2 つあると認識され、スクリプトの誤動作の原因になる場合があります。これは非常に一般的なエラーですが、多くの場合その診断は困難です。



注: 手動で変数を強制的に宣言する必要はありません。変数宣言は、デフォルトで強制的に行われません。Option Explicit ステートメントを使用して、スクリプトの変数宣言の強制を手動で無効にすることはできません。これは、スクリプトが起動される前に Silk Test Workbench が非表示の自動インポートを実行するためで、Option Explicit ステートメントは、最初のインポートの前に設定する必要があります。

キャプチャされたデータのための変数の作成

オブジェクトの識別 ダイアログ ボックスを使用して、オブジェクト情報をキャプチャします。その後、キャプチャしたロケータ情報を表す変数を作成できます。これを行うには、**オブジェクトの識別** ダイアログ ボックスを使用してオブジェクトをキャプチャし、オブジェクト情報をスクリプトに貼り付けます。以下に示すように、オブジェクトに対して Console.Write メソッドを使用し、オブジェクトに保持されているテキストを取得します。

```
With _desktop.WPFWindow("my main window")
    .WPFTextBox("@automationId='textBoxSingle']").SetText(carmake)
    Console.Write (carmake)
End With
```

または、以下の例も有効です。

```
<my window variable>.WPFTextBox("@automationId='textBoxSingle']").SetText(carmake)
Console.Write (carmake)
```

次に、このステートメントの結果、つまりコントロールに保持されるテキストを表す変数を作成できます。データ型に *String* を使用して変数を宣言し、この変数をステートメントの結果に設定します。変数を設定すると、そのデータを評価、検証、表示、または他のテストの目的のために使用することができます。以下は、キャプチャされたテキストの値に設定された変数をメッセージボックスに表示します。

```
Dim carmake As String  
  
carmake = window.WPFTextBox("@automationId='textBoxSingle']").GetLineText(0)  
  
MsgBox (carmake, vbOKOnly)
```

アプリケーションの情報をキャプチャし、変数に設定するこの技法は、テストアプリケーションで識別可能な任意のオブジェクト（ウィンドウ、リストボックス、ボタンなど）で使用することができます。

この目的のためには、`Console.WriteLine` メソッドを使用する以外に、`Text` プロパティも使用できます。

コードは以下のようになります。

```
Dim price = browser.DomTextField("@name=txtDealerPrice").Text
```

`Text` プロパティは、コントロールからのすべてのデータを戻します。ただし、`Text` プロパティはすべての Silk Test Workbench オブジェクトのメンバではありません。`Text` プロパティは、サポートされるオブジェクトでデータが表示フィールドのサイズを超える可能性がある場合に使用します。

複数のコントロールまたは画面全体からテキストをキャプチャするには、`Console.WriteLine` を使用します。たとえば、Web サイトのすべてのテキストを取得するには、以下のように入力します。

```
Console.WriteLine(_desktop.BrowserApplication().Find("//HTML").GetProperty("innerText"))
```

Web サイトのテキストの取得の詳細については、「*textContent*、*innerText*、および *innerHTML* の違い」を参照してください。

変数のデータの検証

再生中にデータ変数の内容を検証する最適な方法は、Visual Basic の `MsgBox` 機能を使用することです。

再生によって、変数の内容がメッセージボックスに表示されます。メッセージボックスは、**OK** ボタンをクリックするまで表示されたままになり、クリックすると再生が継続されます。

以下の例は、変数の内容を表示するためのコマンドの簡単な使用例です。

```
Imports SilkTest.Ntf.Wpf  
  
Public Module Main  
    Dim _desktop As Desktop = Agent.Desktop  
  
    Public Sub Main()  
        Dim sCarmake = "Honda"  
        MsgBox(sCarmake)  
  
        With _desktop.WPFWindow("@caption='WPF Sample Application'")  
            .Restore()  
            .WPFMenuItem("@caption='Basic Controls').Select()  
        End With  
        With _desktop.WPFWindow("@caption='Basic Controls'")  
            .WPFTabControl("@automationId='tabControl').Select("Text")  
            .WPFTextBox("@automationId='textBoxSingle').SetText(sCarmake)  
        End With  
    End Sub  
  
End Module
```

上記の例では、`sCarmake` 変数値（この場合は *Honda*）が再生中にメッセージボックスに表示されます。



注: この例では、MsgBox 機能の実装方法を示すことを意図しています。このコードを正しく実装するには、アプリケーション構成を追加して、テスト アプリケーションを必要に応じて調整する必要があります。このサンプル コードをスクリプトに貼り付けても、アプリケーション構成が存在しないため、エラーが発生します。

スクリプトでの変数の使用

任意の Silk Test Workbench コマンド内で二重引用符で囲まれている値（文字列）は、変数の値によって置き換えることができます。通常、これは文字列変数です。ただし、フィールドへの値の入力時に Integer や Double などの他のデータ型に対して変数を使用できます。

変数の最も一般的な使用法は、エディット ボックスなどのコントロールへの入力として使用されるリテラル値を置き換えることです。コントロールにデータを入力する任意のメソッド、たとえば SetText() や TypeKeys() は、リテラル値を変数で置換できます。そのためには、変数を宣言し、値を割り当てて、リテラル値を変数で置き換えます。例：

```
browser.DomTextField("@id='name-search:lastName']").SetText("Smith")
```

'becomes:

```
Dim lastName = "Smith"
```

```
.DomTextField("@id='name-search:lastName']").SetText(sLastName)
```

リストやメニューから項目を選択する場合、入力選択で変数を使用することもできます。例：

```
browser.DomListBox("@id='quick-link:jump-menu').Select("Agent Lookup")
```

'becomes:

```
Dim selection = "Auto Quote"
```

```
browser.DomListBox("@id='quick-link:jump-menu').Select(sSelection)
```

メソッドまたはプロパティから返された値を、評価のために変数に入れることができます。この場合の一般的な使用法は、条件ロジックでの評価に対しブール データ型の値を戻すものです。以下のコードでは、AllowsMultiSelect プロパティによって戻されたブール型の値を使用して、リスト ボックス コントロールで複数の項目を選択できるかどうかを評価する方法を示しています。

```
Dim canMultiSelect As Boolean
```

```
' Returns true if the second list box in the window allows multiple selections
```

```
canMultiSelect = mainWindow.ListBox("[2]").AllowsMultiSelect
```

```
If canMultiSelect Then
```

```
    MsgBox ("Employee selection list box allows to select multiple employees.")
```

```
Else
```

```
    MsgBox ("Employee selection list box allows to only select one employees.")
```

```
EndIf
```


変数は、コントロールの生のアタッチ名で置き換えることができます。文字列連結文字 & を使用します。例：

```
browser.DomLink("@caption='Eye').Select()
```

'could also read:

```
Dim sLink As String
```

```
sLink = "Eye"  
browser.DomLink("@caption=" & sLink).Select()
```


 **ヒント:** Silk Test Workbench では、GUI マップ、またはコントロールやウィンドウにおける多様性に対処するために、ワイルドカード文字もサポートしています。

スクリプト内のパラメータはビジュアルテストと共有できるため、テストソリューション全体にわたって同じデータを使用できます。詳細については、「スクリプトとビジュアルテスト間でのデータの受け渡し」を参照してください。

スクリプトとビジュアルテスト間でのデータの受け渡し

ビジュアルテストとスクリプトの両方を、同じテストソリューションで使用できます。総合的なテストソリューションを考えた場合、それぞれに利点があります。したがって、テストソリューションには、ビジュアルテストとスクリプトの両方を含めることをお勧めします。スクリプトを使用すると、ビジュアルテストでは実行が困難なタスクを実行できます。また逆に、ビジュアルテストを使用すると、スクリプトでは実行が困難なタスクを実行できます。

テストソリューションへのスクリプトとビジュアルテストの統合を支援するために、Silk Test Workbench ではスクリプトとビジュアルテスト間でデータの受け渡しを行えるようになっています。これにより、既存の Silk Test Workbench スクリプトベースで作成されたデータを再利用できるだけでなく、テストソリューション内でデータの整合性を維持したまま、最も効率的なテスト資産を使用できるようになります。スクリプトとビジュアルテスト間でのデータの受け渡しには、スクリプトパラメータを使用します。


 **注:** スクリプトとビジュアルテスト間でデータの受け渡しを行うには、スクリプトがビジュアルテストに挿入され、そこから再生される必要があります。また、スクリプトとビジュアルテストが同じプロジェクトに含まれているか、挿入されたスクリプトまたはビジュアルテストが参照プロジェクトに含まれている必要があります。

スクリプトパラメータを使用すると、以下のことが可能になります。


- ビジュアルテストでデータを作成し、それをスクリプトで使用する。
- ビジュアルテストでデータを作成し、それをスクリプトに渡して変更し、再びビジュアルテストに戻す。
- スクリプトでデータを作成し、それをビジュアルテストに渡す。

その手順は以下のとおりです。

1. 入力パラメータと出力パラメータをスクリプトに追加します。
2. スクリプトをビジュアルテストに挿入します。
3. ビジュアルテスト内で入力パラメータの値を編集することによって、ビジュアルテストからスクリプトにデータを渡します。

 **注:** ビジュアルテストから送信する値はスクリプト内に保存されませんが、スクリプトでの計算で使用できます。

4. ビジュアルテストで、スクリプトの出力パラメータから渡されたデータを格納するローカル変数を作成し、それをビジュアルテスト内で使用します。

 **注:** 入力パラメータと出力パラメータを作成してビジュアルテストで使用方法の詳細な例については、「Silk Test Workbench ビジュアルテストチュートリアルによるこそ」の「ビジュアルテストからのスクリプトの再生」を参照してください。

ビジュアルテストとスクリプト間で受け渡すパラメータの定義

スクリプトとの間で受け渡されるすべてのパラメータは、スクリプトの入力パラメータおよび出力パラメータとして定義する必要があります。

1. ビジュアルテストとの間でパラメータを受け渡すために使用するスクリプトを開きます。

2. ビジュアル テストからスクリプトに渡されるパラメータを定義する入力パラメータを追加します。

- a) **プロパティ** ペインで、**入力パラメータの追加** を右クリックして選択します。**スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
- b) パラメータの名前を **名前** フィールドに入力します。
- c) **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

- d) **デフォルト値** テキスト ボックスは空のままにします。
- e) **OK** をクリックします。**プロパティ** ペインの入力パラメータのリストに、入力パラメータが表示されます。



注: スクリプトに対して定義された入力パラメータは、スクリプト自体には表示されません。入力パラメータは **プロパティ** ペインで確認できます。

3. スクリプトからビジュアル テストに渡されるパラメータを定義する出力パラメータを追加します。

- a) **プロパティ** ペインで、**出力パラメータの追加** を右クリックして選択します。**スクリプト出力パラメータの追加** ダイアログ ボックスが開きます。
- b) **名前** テキスト ボックスに、ビジュアル テストに渡すパラメータの名前を入力します。
- c) **OK** をクリックします。**プロパティ** ペインの出力パラメータのリストに、出力パラメータが表示されます。



注: スクリプトに実際に宣言されている場合を除き、出力パラメータはスクリプトには表示されません。

次のコード サンプルは、渡した変数の使用方法を示します。作成するパラメータを含めるように Main() サブルーチンを変更する方法と、ディクショナリを使用してこれらのパラメータにアクセスする方法を示します。入力パラメータとして `inputParameterName` を、出力パラメータとして `outputParameterName` を定義します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main(args As IDictionary(Of String, Object))
        Dim text As String = args.Item("inputParameterName")
        args.Item("outputParameterName") = "someValue"
    End Sub
End Module
```

4. スクリプトとの間でパラメータを受け渡すために使用するビジュアル テストを開きます。
5. そのスクリプトが少なくとも 1 つのビジュアル テストに含まれている場合、ビジュアル テストで最新のパラメータを表示するには、挿入されたスクリプト ステップの **プロパティ** ペインで、**名前** テキスト ボックスをクリックし、**パラメータの更新** をクリックします。
6. スクリプトで使用する入力パラメータの値を、ビジュアル テストの **プロパティ** ペインで編集します。ビジュアル テストを再生すると、入力パラメータはスクリプトに渡されます。



注: ビジュアルテストから送信する値はスクリプト内に保存されませんが、スクリプトでの計算で使用できます。

パラメータをビジュアルテストに渡すには、出力パラメータもビジュアルテストでローカル変数として設定する必要があります。

ビジュアルテスト内でのスクリプトの実行

ビジュアルテストでは、既存の .NET スクリプトを呼び出して実行できます。これにより、アプリケーションのテストに対するモジュール式アプローチが可能になり、テストの実行をより制御できるようになります。ビジュアルテスト内でスクリプトを実行する場合は、テストサイトの整合性を保ち、テスト対象のテストアプリケーションが正しい初期再生状態になるようにすることが重要です。

この手順を使用してドライバテストを作成することもできます。

1. スクリプトを挿入するビジュアルテストを開きます。
2. 挿入したスクリプトを実行するポイントの前のテストステップを選択します。
3. **挿入 > .NET スクリプト** を選択します。 **.NET スクリプトの参照** ダイアログ ボックスが開きます。
4. 挿入するスクリプトを選択し、**OK** をクリックします。 Silk Test Workbench によって、選択したステップの下にステップが挿入されます。挿入されたステップによって、選択したスクリプトが呼び出され、再生されます。ステップテキストは、次のようになります：

```
.NET スクリプト '資産名' を再生します
```

ここで、'資産名' は、挿入されたスクリプトの名前です。

再生中、前のステップが実行されると、ビジュアルテスト内の次のステップが実行される前に、挿入されたスクリプトが最後まで再生されます。スクリプトがビジュアルテストに挿入されたあと、スクリプトを編集して、再生に使用される設定を構成できます。

テスト対象アプリケーションは、必要に応じて設定、フック、開始され、再生セッションの終了時までフックされたままになります。たとえば、ビジュアルテストにメモ帳用のアプリケーション構成があり、.NET スクリプトに電卓用のアプリケーション構成がある場合には、次のようになります。

- ビジュアルテストの実行の開始時に、メモ帳がフックされ、開始されます。
- 挿入された .NET スクリプトの実行の開始時に、電卓がフックされ、開始されます。メモ帳はフックされたままです。
- 挿入された .NET スクリプトの実行を停止しても、メモ帳と電卓の両方ともフックされたままになります。
- 再生セッションの終了時、つまり、ビジュアルテストの実行を停止すると、メモ帳と電卓の両方のフックが解除されます。

ビジュアルテストにおける、スクリプトから受け取ったパラメータの使用

スクリプトに対して入力パラメータと出力パラメータが定義されたら、再生中にスクリプトを実行するビジュアルテストに出力パラメータを渡すことができます。ただし、渡されたパラメータをビジュアルテストで使用するには、それをビジュアルテストのローカル変数に設定する必要があります。

1. ビジュアルテストを開きます。
2. **テストステップ** ペインで、スクリプトを実行するビジュアルテスト内のステップを選択します。ステップテキストは、「.NET スクリプト '資産名' を再生します」のようになります (ここで、資産名は挿入されたスクリプトの名前です)。

ステップのプロパティが **プロパティ** ペインに表示されます。ここには、スクリプトの入力パラメータと出力パラメータが表示されます。

3. ステップの **プロパティ** ペインで、**出力パラメータ** カテゴリを展開します。

スクリプトからビジュアルテストに渡される出力パラメータとして定義された変数の名前が、[出力パラメータ] サブカテゴリに表示されます。これらのパラメータを、ビジュアルテストに作成されたローカル変数に設定する必要があります。

パラメータが **プロパティ** ペインに表示されない場合は、パラメータが含まれるスクリプトを再読み込みします。これを行うには、挿入されたスクリプトステップの **プロパティ** ペインで、**名前** テキストボックスをクリックし、**パラメータ名の更新** をクリックします。

4. **プロパティ** ペインで出力パラメータ名のテキストボックスをクリックし、リストから適切なローカル変数を選択します。これにより、スクリプトからビジュアルテストのローカル変数に渡されるパラメータの値が設定されるため、この値をビジュアルテスト内で使用できるようになります。
5. スクリプトからビジュアルテストに渡す **プロパティ** ペインの出力パラメータごとに、上記のステップを繰り返します。



注: スクリプトからビジュアルテストに渡された値を確認するには、**結果コメント** ステップをビジュアルテストに挿入します。コメントの [式] **プロパティ** に対して、**選択** をクリックして **変数** を選択します。渡された値を含むローカル変数をリストから選択します。

6. ビジュアルテストで、適切なテストステップを変更してローカル変数を組み込みます。
ローカル変数によって、スクリプトの出力パラメータからテストステップにデータが渡されます。
 - a) 使用するステップを **テストステップ** ペインで選択します。
 - b) **プロパティ** ペインで、ローカル変数のプロパティタイプに対応するプロパティのテキストボックスをクリックします。
たとえば、**パラメータ** カテゴリの **text** テキストボックスをクリックします。
 - c) **選択** ボタンをクリックして、**変数** を選択します。**変数の選択** ダイアログボックスが開きます。
 - d) 使用するローカル変数を選択し、**OK** をクリックします。ステップテキストは、次のようになります：
す：「'[localvariablename]」を入力します」

再生するビジュアルテストにスクリプトが含まれていて、その入力パラメータと出力パラメータが正しく定義されている場合は、ビジュアルテストを再生すると、ビジュアルテストとこのビジュアルテスト内のスクリプトの間でパラメータを正しく受け渡しできます。



注: 入力パラメータと出力パラメータを作成してビジュアルテストで使用方法の詳細な例については、「*SilkTest Workbench* ビジュアルテストチュートリアルによるこそ」の「ビジュアルテストからのスクリプトの再生」を参照してください。

スクリプトでのパラメータの作成と受け渡し

パラメータを作成して渡すことは、チームによるテスト環境で役立ちます。たとえば、経験の豊富なテスト担当者が共通するテスト機能を実行するスクリプトのライブラリを作成し、経験の浅い開発者がそこからスクリプトを選択して、より基本的なスクリプトを作成することができます。パラメータを作成して渡すことは、1人のテスト担当者の環境でも、複雑な機能を複数のスクリプトで再利用する場合に役立ちます。

親スクリプトと子スクリプト間で、または単一のスクリプトを使用して、パラメータを作成して渡すことができます。親子シナリオを使用することには、複数の子スクリプトで親スクリプトを使用できるという利点があります。たとえば、親スクリプトによって乱数が作成される場合に、その機能を複数の子スクリプトで使用することができます。

スクリプト間でのパラメータの作成と受け渡し

親スクリプトと子スクリプト間で、または単一のスクリプトを使用して、パラメータを作成して渡すことができます。親子シナリオを使用することには、複数の子スクリプトで親スクリプトを使用できるという利点があります。たとえば、親スクリプトによって乱数が作成される場合に、その機能を複数の子スクリプトで使用することができます。

1. 渡すパラメータを組み込む親スクリプトを作成します。
 - a) **ファイル** > **新規作成** を選択します。**資産の新規作成** ダイアログボックスが開きます。
 - b) [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキストボックスにスクリプトの名前を入力します。
たとえば、スクリプトに「NameParameters」という名前を付けることができます。

- c) **OK** をクリックして、スクリプトを保存します。Silk Test Workbench によってスクリプトが資産として保存され、スクリプト テンプレートが表示されます。
- d) 作成して子スクリプトに渡すパラメータが含まれるように、Main() サブを変更します。たとえば、以下の親スクリプトは、テスト アプリケーションで姓名の文字列を渡します。

```
Public Sub Main()
    Dim args As New Dictionary(Of String, Object)
    args("FName") = "Chris"
    args("LName") = "Smith"
```

- e) 子スクリプトの呼び出しと作成したパラメータの名前指定を行う、RunScript コマンドを組み込みます。たとえば、以下の親スクリプトは、テスト アプリケーションで姓名の文字列を渡し、*childscript* という名前の子スクリプトを *args* というパラメータを使用して呼び出します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        Dim args As New Dictionary(Of String, Object)
        args("FName") = "Chris"
        args("LName") = "Smith"

        Workbench.RunScript("childscript", args)
    End Sub
End Module
```

2. 省略可能 : スクリプトで入力されるパラメータを返すメッセージ ボックスを組み込むには、以下のコードを親スクリプトに追加します。

```
MsgBox("Hello, " + args("FName") + "!")
MsgBox ("Hello, " + args("FName") + " " + args("LName") + "!")
```

スクリプトの再生時に、テストで入力される値を示すメッセージ ボックスが表示されます。

3. 親スクリプトからパラメータを呼び出す、子スクリプトを作成します。たとえば、サンプル Web アプリケーションの登録フォームで姓名を処理する場合は、姓名を含む登録フォームのスクリプトを記録します。以下に、スクリプトの例を示します。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")
            .DomButton("@id='login-form:signup']").Select()
            .DomTextField("@id='signup:fname']").SetText("first")
            .DomTextField("@id='signup:lname']").SetText("last")
            .DomElement("@src='http://extjs.com/s.gif']").DomClick(MouseButton.Left,
                New Point(8, 16))
            .DomButton("@textContents='Today']").Select()
            .DomTextField("@id='signup:email']").SetText("test@test1.com")
            .DomTextField("@id='signup:street']").SetText("123 street rd")
            .DomTextField("@id='signup:city']").SetText("Marlton")
            .DomListBox("@id='signup:state']").Select("Massachusetts")
            .DomTextField("@id='signup:zip']").SetText("09876")
            .DomTextField("@id='signup:password']").SetText("test")
            .DomButton("@id='signup:signup']").Select()
            .DomButton("@id='signup:continue']").Select()
        End With
    End Sub
End Module
```

4. 設定して親スクリプトに渡す、IDictionary(Of String, Object) という 1 つのパラメータを使用するように Sub Main() を変更します。

IDictionary クラスは、.NET によって提供されます。クラスのドキュメントは、<http://msdn.microsoft.com/en-us/library/s4ys34ea.aspx> で確認できます。

修正された Sub Main () コードは、以下のようになります。

```
Public Sub Main(args As IDictionary(Of String, Object))
```

5. パラメータを使用するコントロールを変更します。

たとえば、Web アプリケーションの記録で姓名を変更するには、以下のコードを置換します。

```
.DomTextField("@id='signup:fname']").SetText("first")  
.DomTextField("@id='signup:lname']").SetText("last")
```

以下のように置換します。

```
.DomTextField("@id='signup:fname']").SetText(args("FName"))  
.DomTextField("@id='signup:lname']").SetText(args("LName"))
```

6. 親スクリプトから、**再生** をクリックします。



注: 子スクリプトからのパラメータを使用するスクリプトを実行すると、パラメータが指定されていないため、スクリプトは失敗します。ただし、パラメータを使用しない 2 番目の Main() サブを追加し、デフォルトのパラメータ値を他方の Main() に渡して単一のスクリプトから直接実行できます。

単一のスクリプト内でのパラメータの作成と受け渡し

親スクリプトと子スクリプト間で、または単一のスクリプトを使用して、パラメータを作成して渡すことができます。親子シナリオを使用することには、複数の子スクリプトで親スクリプトを使用できるという利点があります。たとえば、親スクリプトによって乱数が作成される場合に、その機能を複数の子スクリプトで使用することがあります。

1. スクリプトを作成します。

a) **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。

b) [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。

たとえば、スクリプトに「NameParameters」という名前を付けることができます。

c) **OK** をクリックして、スクリプトを保存します。Silk Test Workbench によってスクリプトが資産として保存され、スクリプト テンプレートが表示されます。

2. アプリケーションをテストするスクリプトを記録または入力します。

たとえば、サンプル Web アプリケーションの登録フォームで姓名を処理する場合は、姓名を含む登録フォームのスクリプトを記録します。以下に、スクリプトの例を示します。

```
Imports SilkTest.Ntf.XBrowser  
Public Module Main  
    Dim _desktop As Desktop = Agent.Desktop  
  
    Public Sub Main()  
        With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")  
            .DomButton("@id='login-form:signup']").Select()  
            .DomTextField("@id='signup:fname']").SetText("first")  
            .DomTextField("@id='signup:lname']").SetText("last")  
            .DomElement("@src='http://extjs.com/s.gif']").DomClick(MouseButton.Left,  
                New Point(8, 16))  
            .DomButton("@textContents='Today']").Select()  
            .DomTextField("@id='signup:email']").SetText("test@test1.com")  
            .DomTextField("@id='signup:street']").SetText("123 street rd")  
            .DomTextField("@id='signup:city']").SetText("Marlton")  
            .DomListBox("@id='signup:state']").Select("Massachusetts")  
        End With  
    End Sub  
End Module
```

```

.DomTextField("@id='signup:zip").SetText("09876")
.DomTextField("@id='signup:password").SetText("test")
.DomButton("@id='signup:signup").Select()
.DomButton("@id='signup:continue").Select()
End With

```

```

End Sub
End Module

```

3. 作成して子スクリプトに渡すパラメータが含まれる、2 番目の Main() サブを作成します。たとえば、以下のスクリプトは、テスト アプリケーションで姓名の文字列を渡します。

```

Public Sub Main()
    Dim args As New Dictionary(Of String, Object)
    args("FName") = "Chris"
    args("LName") = "Smith"

```

4. 最初の Main() サブ スクリプトの呼び出しと作成したパラメータの名前指定を行う、Main(args) コマンドを組み込みます。たとえば、以下のコードは、テスト アプリケーションで姓名の文字列を渡し、args というパラメータを呼び出します。

```

Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        Dim args As New Dictionary(Of String, Object)
        args("FName") = "Chris"
        args("LName") = "Smith"

        Main(args)
    End Sub
End Module

```

5. 設定して親スクリプトに渡す、IDictionary(Of String, Object) という 1 つのパラメータを使用するように、アプリケーションをテストする Sub Main() を変更します。

IDictionary クラスは、.NET によって提供されます。クラスのドキュメントは、<http://msdn.microsoft.com/en-us/library/s4ys34ea.aspx> で確認できます。

修正された Sub Main () コードは、以下のようになります。

```

Public Sub Main(args As IDictionary(Of String, Object))

```

6. パラメータを使用するコントロールを変更します。

たとえば、Web アプリケーションの記録で姓名を変更するには、以下のコードを置換します。

```

.DomTextField("@id='signup:fname").SetText("first")
.DomTextField("@id='signup:lname").SetText("last")

```

以下のように置換します。

```

.DomTextField("@id='signup:fname").SetText(args("FName"))
.DomTextField("@id='signup:lname").SetText(args("LName"))

```

スクリプトは以下のようになります。

```

Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main(args As IDictionary(Of String, Object))
        With _desktop.BrowserWindow("/BrowserApplication[1]/BrowserWindow")
            .DomButton("@id='login-form:signup").Select()
            .DomTextField("@id='signup:fname").SetText(args("FName"))

```

```

.DomTextField("@id='signup:lname").SetText(args("LName"))
.DomElement("@src='http://extjs.com/s.gif").DomClick(MouseButton.Left,
    New Point(8, 16))
.DomButton("@textContents='Today').Select()
.DomTextField("@id='signup:email").SetText("test@test1.com")
.DomTextField("@id='signup:street").SetText("123 street rd")
.DomTextField("@id='signup:city").SetText("Marlton")
.DomListBox("@id='signup:state").Select("Massachusetts")
.DomTextField("@id='signup:zip").SetText("09876")
.DomTextField("@id='signup:password").SetText("test")
.DomButton("@id='signup:signup").Select()
.DomButton("@id='signup:continue").Select()
End With

End Sub

Public Sub Main()
    Dim args As New Dictionary(Of String, Object)
    args("FName") = "Chris"
    args("LName") = "Smith"

    Main(args)
End Sub

End Module

```

7. 省略可能 : スクリプトで入力されるパラメータを返すメッセージ ボックスを組み込むには、以下のコードを追加します。

```

MsgBox("Hello, " + args("FName") + "!")
MsgBox ("Hello, " + args("FName") + " " + args("LName") + "!")

```

スクリプトの再生時に、テストで入力される値を示すメッセージ ボックスが表示されます。

スクリプトは以下ようになります。

```

Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main(args As IDictionary(Of String, Object))
        With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")
            .DomButton("@id='login-form:signup").Select()
            .DomTextField("@id='signup:fname").SetText(args("FName"))
            .DomTextField("@id='signup:lname").SetText(args("LName"))
            .DomElement("@src='http://extjs.com/s.gif").DomClick(MouseButton.Left,
                New Point(8, 16))
            .DomButton("@textContents='Today').Select()
            .DomTextField("@id='signup:email").SetText("test@test1.com")
            .DomTextField("@id='signup:street").SetText("123 street rd")
            .DomTextField("@id='signup:city").SetText("Marlton")
            .DomListBox("@id='signup:state").Select("Massachusetts")
            .DomTextField("@id='signup:zip").SetText("09876")
            .DomTextField("@id='signup:password").SetText("test")
            .DomButton("@id='signup:signup").Select()
            .DomButton("@id='signup:continue").Select()
        End With

    End Sub

    Public Sub Main()

```

```

Dim args As New Dictionary(Of String, Object)
args("FName") = "Chris"
args("LName") = "Smith"

Main(args)

MsgBox("Hello, " + args("FName") + "!")
MsgBox ("Hello, " + args("FName") + " " + args("LName") + "!")

End Sub

End Module

```

8. **再生** をクリックします。

パラメータを含むスクリプトの例

以下の例は、パラメータを作成して渡すスクリプトを示しています。

リストの例

以下の例は、メモ帳アプリケーションをテストする親スクリプトと子スクリプト間でリストパラメータを作成して渡す方法を示しています。

以下に、親スクリプトの例を示します。

```

Public Module Main
Dim _desktop As Desktop = Agent.Desktop

Public Sub Main()
Dim numList As New List(Of Integer)
For i As Integer = 0 To 10
numList.Add(i)
Next

Dim args As New Dictionary(Of String, Object)
args("Numbers") = numList

Workbench.RunScript("ChildScriptArgs", args)
End Sub
End Module

```

以下に、子スクリプトの例を示します。

```

Public Module Main
Dim _desktop As Desktop = Agent.Desktop
Public Sub Main(args As IDictionary(Of String, Object))
Dim nums As List(Of Integer) = args("Numbers")

With _desktop.Window("/Window[@caption='Untitled - Notepad']")
For Each num As Integer In nums
.TextField().TypeKeys(num.ToString() + "<Enter>")
Next
End With

End Sub
End Module

```

列挙の例

以下の例は、メモ帳アプリケーションをテストする親スクリプトと子スクリプト間で列挙パラメータを作成して渡す方法を示しています。

以下に、親スクリプトの例を示します。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)
    args("Day") = DayOfWeek.Friday

    Workbench.RunScript("ChildDayScript", args)

  End Sub
End Module
```

以下に、子スクリプトの例を示します。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    Dim day As DayOfWeek = args("Day")

    With _desktop.Window("/Window[@caption='Untitled - Notepad']")
      .TextField().TypeKeys("The day of the week is " + day.ToString() + ".")
    End With

  End Sub
End Module
```

数値の例

以下の例は、メモ帳アプリケーションをテストする単一のスクリプトで数値パラメータを作成して渡す方法を示しています。

スクリプトの内容は以下のとおりです。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    Dim N1 As Integer = CType(args("Num1"), Integer)
    Dim N2 As Integer = CType(args("Num2"), Integer)

    With _desktop.Window("@caption='Untitled - Notepad'")
      .TextField().ClearText()
      .TextField().TypeKeys("<Enter>" + "Numbers" + "<Enter>")
      .TextField().TypeKeys(N1)
      .TextField().TypeKeys("<Enter>")
      .TextField().TypeKeys(N2)
      .TextField().TypeKeys("<Enter>")
    End With
  End Sub

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)

    args.Add("Num1", 35)
    args.Add("Num2", 25)

    Main(args)
  End Sub
End Module
```

ブール値の例

以下の例は、メモ帳アプリケーションをテストする単一のスクリプトでブール値パラメータを作成して渡す方法を示しています。

スクリプトの内容は以下のとおりです。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    Dim B1 As Boolean = CType(args("Bool1"), Boolean)
    Dim B2 As Boolean = CType(args("Bool2"), Boolean)

    With _desktop.Window("@caption='Untitled - Notepad'")
      .TextField().ClearText()
      .TextField().TypeKeys("<Enter>" + "Boolean Values" + "<Enter>")
      .TextField().TypeKeys(B1)
      .TextField().TypeKeys("<Enter>")
      .TextField().TypeKeys(B2)
      .TextField().TypeKeys("<Enter>")
    End With
  End Sub

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)

    args.Add("Bool1", True)
    args.Add("Bool2", False)

    Main(args)
  End Sub
End Module
```

配列の例

以下の例は、メモ帳アプリケーションをテストする単一のスクリプトにおいて、配列パラメータを作成して渡す方法を示しています。

スクリプトの内容は以下のとおりです。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    Dim A1() As Integer = CType(args("Arr1"), Integer())

    With _desktop.Window("@caption='Untitled - Notepad'")
      .TextField().TypeKeys("Array Numbers<Enter>")

      For Each num As Integer In A1
        .TextField().TypeKeys(num)
        .TextField().TypeKeys("<Enter>")
      Next
      .TextField().TypeKeys("<Enter>")
    End With
  End Sub

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)
```

```
args("Arr1") = New Integer() {1,2,3,4,5}

Main(args)
End Sub
End Module
```

スクリプトの管理

このセクションでは、VB .NET スクリプトの管理方法について説明します。

スクリプト内のコードの検索と置換

VB .NET スクリプト内のコードの検索と置換を使用すると、スクリプトを迅速に変更できます。

1. 検索するスクリプトを開きます。
2. **編集 > 検索と置換** を選択し、以下のいずれかのオプションを選択します。
 - **検索** : **検索する文字列** テキスト ボックスに検索するコードを指定し、**次を検索** をクリックします。
 - **次を検索** : **検索する文字列** テキスト ボックスに指定した項目の次のインスタンスを検索するには、このオプションを選択します。
 - **前を検索** : 検索された最後のエントリの前にあるスクリプトで、**検索する文字列** テキスト ボックスに指定した項目を検索するには、このオプションを選択します。
 - **置換** : **検索する文字列** テキスト ボックスに検索するコードを指定します。**置換後の文字列** テキスト ボックスに、既存のコードと置換するコードを指定し、**置換** または **すべて置換** をクリックします。

ブックマークを使用したスクリプトでのコードの移動

ブックマークを使用して、スクリプトの重要な部分に移動します。

1. ブックマークを組み込むスクリプトを開きます。
2. **編集 > ブックマーク** を選択し、以下のいずれかのオプションを選択します。
 - **ブックマークの切り替え** : 選択したコード行でブックマークを組み込むか非表示にするには、このオプションを選択します。
 - **次のブックマーク** : スクリプト内の次のブックマークに移動するには、このオプションを選択します。
 - **前のブックマーク** : スクリプト内の検索された最後のブックマークの前のブックマークに移動するには、このオプションを選択します。
 - **ブックマークのクリア** : スクリプトからすべてのブックマークを削除するには、このオプションを選択します。

スクリプトの入力パラメータの追加

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。

1. **プロパティ** ペインで、**入力パラメータの追加** を右クリックして選択します。**スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
2. パラメータの名前を **名前** フィールドに入力します。

パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
3. **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. デフォルト値 テキスト ボックスは空のままにします。

空のままにした場合、ビジュアルテストで設定されたパラメータの初期値が使用されます。この値は、ビジュアルテスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。

5. OK をクリックします。プロパティ ペインの入力パラメータのリストに、入力パラメータが表示されます。

スクリプトの入力パラメータの編集

スクリプトの入力パラメータを編集して、パラメータを渡すスクリプトやビジュアルテストに行える変更を反映します。

1. プロパティ ペインで、編集する入力パラメータを右クリックし、**<inputparametername> の編集** を選択します。スクリプト入力パラメータの編集 ダイアログ ボックスが開きます。

2. パラメータの名前を **名前** フィールドに入力します。

パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。

3. **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. デフォルト値 テキスト ボックスは空のままにします。

空のままにした場合、ビジュアルテストで設定されたパラメータの初期値が使用されます。この値は、ビジュアルテスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。

5. OK をクリックします。

スクリプトからの入力パラメータの削除

必要に応じて、スクリプトから入力パラメータを削除できます。

1. **プロパティ** ペインで、削除する入力パラメータを右クリックし、**<inputparametername> の削除** を選択します。
2. **はい** をクリックして、入力パラメータを削除することを確認します。

スクリプトの出カパラメータの追加

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。

1. **プロパティ** ペインで、**出力パラメータの追加** を右クリックして選択します。 **スクリプト出力パラメータの追加** ダイアログ ボックスが開きます。
2. **名前** テキスト ボックスに、ビジュアル テストに渡すパラメータの名前を入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
3. **型** リストから、データを渡すビジュアル テスト内のローカル変数のデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. **OK** をクリックします。 **プロパティ** ペインの出力パラメータのリストに、出力パラメータが表示されます。

スクリプトの出カパラメータの編集

ビジュアル テストにデータを渡すために、スクリプトの出力パラメータを編集します。

1. **プロパティ** ペインで、編集する出力パラメータを右クリックし、**<outputparametername> の編集** を選択します。
2. **名前** テキスト ボックスに、ビジュアル テストに渡すパラメータの名前を入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
3. **型** リストから、データを渡すビジュアル テスト内のローカル変数のデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。

数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. **OK** をクリックします。

スクリプトからの出力パラメータの削除

必要に応じて、スクリプトから出力パラメータを削除できます。

1. **プロパティ** ペインで、削除する出力パラメータを右クリックし、**<outputparametername> の削除** を選択します。
2. **はい** をクリックして、出力パラメータを削除することを確認します。

スクリプトへのファイルの追加

スクリプトが参照するファイルを追加して、特定の目的ごとにコードを論理的にセグメント化できます。たとえば、スクリプトで 4000 行のコードが必要な場合、**メイン** ウィンドウにはコードの中心的な行のみを含めるために、ヘルパコード用の別のファイルを作成することがあります。ヘルパコードとは、データ転送のためにメイン アプリケーションをデータベースに接続するコードなどのことです。

1. ファイルをスクリプトに追加するには、以下のいずれかのステップを実行します。
 - **挿入 > ファイル** を選択します。
 - **プロパティ** ペインを右クリックし、**ファイルの追加** を選択します。

コード ウィンドウで新しいタブが開きます。

2. ファイルに含めるコードを追加し、**保存** をクリックします。
3. **メイン** タブをクリックし、ファイルの参照を追加します。

スクリプトからのファイルの削除

スクリプトが参照するファイルを削除できます。

1. 削除するファイルが含まれているスクリプトを開きます。
2. **プロパティ** ペインで、**ファイル** カテゴリを展開し、ファイル名を右クリックして **ファイルの削除** を選択します。

スクリプト内のファイル名の変更

スクリプトが参照するファイルの名前を変更できます。

1. 名前を変更するファイルが含まれているスクリプトを開きます。
2. **プロパティ** ペインで、**ファイル** カテゴリを展開し、ファイル名を右クリックして **ファイル名の変更** を選択します。
3. **プロパティ** ペインで新しい名前を入力します。

.NET スクリプト内からの.NET アセンブリの参照

スクリプトを記録する場合、操作するコントロールに対して適切なすべての参照が参照ライブラリに自動的に追加されます。手動でスクリプトを作成するか、またはアプリケーション構成を追加すると、参照を手動でライブラリに追加することが必要となる場合があります。.NET スクリプトの参照として、.NET Framework の .NET アセンブリを追加できます。

1. スクリプト参照ライブラリに参照を追加するには、**プロパティ** ペインを右クリックし、**.NET アセンブリ参照の追加** を選択します。**.NET アセンブリ参照の追加** ダイアログ ボックスが開きます。

2. コンポーネントを追加するには、以下のいずれかのステップを実行します。

- コンポーネント リストからコンポーネントをダブルクリックします。
- **参照** をクリックし、追加するコンポーネントに移動し、**開く** をクリックして、**OK** をクリックします。

パス 列には、DLL ファイルの場所の一覧が表示されます。

新しいコンポーネントが **プロパティ** ペインの **参照** カテゴリに表示されます。

.NET スクリプト内から他の .NET スクリプトのクラスの参照

.NET スクリプトで他の .NET スクリプトで定義したクラスを使用するには、他の .NET スクリプトへの参照を .NET スクリプトに追加します。

1. **プロパティ** ペインを右クリックし、**.NET スクリプト参照の追加** を選択します。**.NET スクリプトの参照** ダイアログ ボックスが開きます。
2. ダイアログ ボックスに表示する .NET スクリプトを選択します。
 - 自分で作成したすべての .NET スクリプトを表示する場合は、**自分で作成したファイル** をクリックします。
 - すべてのユーザーが作成したすべての .NET スクリプトを表示する場合は、**すべてのユーザーが作成したファイル** をクリックします。
3. リストから .NET スクリプトを参照として現在の .NET スクリプトに追加するには、以下のステップのいずれかを実行します。
 - リストから .NET スクリプトをダブルクリックします。
 - リストから .NET スクリプトを選択して、**OK** をクリックします。
 - リストから .NET スクリプトを選択して、**挿入して開く** をクリックして、.NET スクリプトを参照として追加して、参照したスクリプトを開きます。


参照した .NET スクリプトは **プロパティ** ペインの **参照** カテゴリに表示されます。

スクリプト参照ライブラリからの参照の削除

スクリプト参照ライブラリから参照を削除するには、**プロパティ** ペインを右クリックし、**参照の削除** を選択します。

[.NET スクリプトの参照] ダイアログ ボックス


.NET スクリプトの参照 ダイアログ ボックスを使用して、自分自身または他のユーザーが作成したスクリプトをすばやく検索して選択できます。

 **注: 資産の選択** リストに表示されるのは、*Common* プロジェクトのスクリプトのみです。

.NET スクリプトの参照 ダイアログ ボックスには、以下のコントロールがあります。

コントロール	説明
作成した資産の表示	資産の選択 リストに表示する資産を定義するためのリストボックスです。 自分で作成したファイル を選択すると、ユーザーが作成した資産のみが表示され、 すべてのユーザーが作成したファイル を選択すると、任意のユーザーが作成した資産が表示されます。
名前フィルタ	資産の名前をこのテキスト フィールドに入力すると、 資産の選択 リストにその名前と一致した資産のみが表示されます。
資産の選択	適用したフィルタによって表示が変化する資産のリストです。


資産の選択 リストから目的の VB .NET スクリプトを選択し、**OK** ボタンを選択して使用します。

 **注:** 列ヘッダーをクリックすると、クリックした列によって **資産の選択** リストがアルファベット順にソートされます。

.NET スクリプトの比較

2つの異なる .NET スクリプトや、同じ .NET スクリプトの2つのバージョンを比較して、これらの間の相違点を調べることができます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **.NET スクリプト** を選択します。選択したプロジェクトの既存の .NET スクリプトが、右側のペインに表示されます。

 **ヒント:** 調査したい .NET スクリプトが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2つの異なる .NET スクリプトを比較するには：
 - a) **Ctrl + クリック** を使用して、比較する2つの .NET スクリプトを選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
4. 同じ .NET スクリプトの2つのバージョンを比較するには：
 - a) 調査したい .NET スクリプトを右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する2つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。


- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレイで表示されます。


Windows DLL の呼び出し


このセクションでは、DLL を呼び出す方法について説明します。DLL は Open Agent のプロセス内から、または AUT (テスト対象アプリケーション) から呼び出すことができます。これにより、テスト スクリプト内の既存のネイティブ DLL を再利用できます。

Open Agent 内の DLL 呼び出しは通常、AUT 内の UI コントロールと対話しないグローバル関数を呼び出す場合に使用されます。

AUT 内の DLL 呼び出しは通常、アプリケーションの UI コントロールと対話する関数を呼び出す場合に使用されます。これにより、Silk Test Workbench は再生中に DLL 呼び出しを自動的に同期できます。

 **注:** 32 ビット アプリケーションでは 32 ビット DLL を、64 ビット アプリケーションでは 64 ビット DLL を呼び出すことができます。Open Agent は 32 ビットと 64 ビットの両方の DLL を実行できます。

 **注:** .NET Framework では、P/Invoke という DLL 呼び出しも組み込みでサポートされています。P/Invoke を Visual Basic スクリプト内で使用すると、このスクリプトを実行するプロセス内で DLL 関数を呼び出すことができます。ただし、AUT では Silk Test Workbench を使用して DLL 関数を呼び出すことができる一方で、自動同期は行われません。

 **注:** DLL を呼び出すには、C インターフェイスを使用する必要があります。同様に .dll というファイル拡張子の付いた .NET アセンブリを呼び出す場合は、DLL 呼び出し機能を使用しないで、.NET スクリプト内でアセンブリへの参照を追加します。

スクリプトからの Windows DLL の呼び出し

DLL の宣言を開始するには、DLL 属性を持つインターフェイスを使用します。宣言の構文は次のとおりです。

```
<Dll("dllname.dll")> Public Interface DllInterfaceName
    FunctionDeclaration
    [FunctionDeclaration]...
End Interface
```

dllname Visual Basic スクリプトから呼び出す関数が含まれた DLL ファイルの完全パスの名前。DLL パスの環境変数は自動的に解決されます。パスでは二重のバックスラッシュ (¥¥) を使用する必要はありません。単一のバックスラッシュ (¥) で十分です。

DllInterfaceName スクリプト内で DLL と対話するために使用される識別子。

FunctionDeclaration 呼び出そうとしている DLL 関数の関数宣言。

DLL 関数の宣言構文

DLL 関数の宣言は、一般に以下の形式を取ります。

```
Function function-name( [arg-list] ) As return-type
```

戻り値のない関数の場合、宣言の形式は以下のとおりです、

```
Sub function-name( [arg-list] )
```

return-type 戻り値のデータ型。

function-name 関数の名前。

arg-list 関数に渡される引数のリスト。

リストは以下のように指定します。

```
[pass-mode] identifier As data-type
```

pass-mode 引数が関数に渡されるのか (ByVal)、それとも引数を関数によって変更できるのか (ByRef) を指定します。ByRef キーワードを使用して、引数の値を関数に渡すこともできます。この場合、関数によって引数の値が変更され、新しい値が出力されます。

data-type 引数のデータ型。

identifier 引数の名前。

DLL 呼び出しの例

この例では、user32.dll の SendMessage DLL 関数を呼び出して、フィールドに「hello world!」というテキストを書き出します。

DLL の宣言 :

```
// VB .NET code
<Dll("user32.dll")> Public Interface IUserDll32Functions
    Function SendMessageW( _
        ByVal obj As TestObject, ByVal message As Integer, ByVal wParam As Integer, ByVal
```

```
IParam As String) As Integer  
End Interface
```

以下のコードは、AUT で宣言された DLL 関数を呼び出す方法を示します。

```
// VB .NET code  
Public Sub Main()  
    Dim user32Functions As IUserDll32Functions = DllCall.CreateInProcessDllCall(Of  
IUserDll32Functions)()  
    Dim textField = _desktop.Window().TextField()  
    user32Functions.SendMessageW(textField, WindowsMessages.WM_SETTEXT, 0, "my text")  
End Sub
```



注: DLL 関数の最初のパラメーターに C データ型の HWND が指定されている場合は、AUT 内で DLL 関数の呼び出しのみを実行できます。

次のコードは、Open Agent のプロセスで宣言された DLL 関数を呼び出す方法を示します。

```
// VB .NET code  
Public Sub Main()  
    Dim user32Functions As IUserDll32Functions = DllCall.CreateAgentDllCall(Of  
IUserDll32Functions)()  
    Dim textField = _desktop.Window().TextField()  
    user32Functions.SendMessageW(textField, WindowsMessages.WM_SETTEXT, 0, "my text")  
End Sub
```



注: コード例では、DLL 関数で使用するのに便利な Windows メッセージングに関連する定数を定義した WindowsMessages クラスを使用しています。

DLL 関数への引数の受け渡し

DLL 関数は C で記述されているため、これらの関数に渡す引数には適切な C データ型を指定する必要があります。次のデータ型がサポートされます。

Integer

次のデータ型を持つ引数または戻り値には、このデータ型を使用します。

- int
- INT
- long
- LONG
- DWORD
- BOOL
- WPARAM
- HWND

Visual Basic の Integer 型は、4 バイト値を持つすべての DLL 引数に対して有効です。

Long

C データ型 long および int64 を持つ引数または戻り値には、このデータ型を使用します。Visual Basic の Long 型は、8 バイト値を持つすべての DLL 引数に対して有効です。

Short





C データ型 short および WORD を持つ引数または戻り値には、このデータ型を使用します。Visual Basic の Short 型は、2 バイト値を持つすべての DLL 引数に対して有効です。

Boolean

C データ型 bool を持つ引数または戻り値には、このデータ型を使用します。

String

C で String となる引数または戻り値には、このデータ型を使用します。

- Double** C データ型 double を持つ引数または戻り値には、このデータ型を使用します。
- Single** C データ型 float を持つ引数または戻り値には、このデータ型を使用します。
- SilkTest.Ntf.Rectangle** C データ型 RECT を持つ引数には、このデータ型を使用します。Rectangle は戻り値として使用できません。
- SilkTest.Ntf.Point** C データ型 POINT を持つ引数には、このデータ型を使用します。POINT は戻り値として使用できません。
- SilkTest.Ntf.TestObject** C データ型 HWND を持つ引数には、このデータ型を使用します。TestObject は戻り値として使用できませんが、戻り値型として Integer を持つ HWND を戻す DLL 関数を宣言できます。
-  **注:** 渡された TestObject は SilkTest.Ntf.INativeWindow インターフェイスを実装して、DLL 関数に渡される TestObject のウィンドウハンドルを Silk Test Workbench が判別できるようにする必要があります。そうしないと、この DLL 関数を呼び出すときに、例外がスローされます。
- List** ユーザー定義の C 構造体の配列には、このデータ型を使用します。List は戻り値として使用できません。
-  **注:** List を出力パラメーターとして使用する場合は、ByRef を使用して、渡されるリストに、戻される内容を保持できるだけのサイズを確保する必要があります。
-  **注:** C 構造体は List で表すことができます。この場合、すべてのリスト要素は構造体のメンバに対応しています。最初の構造体メンバは、リスト内の最初の要素で表されます。2 番目の構造体メンバは、リスト内の 2 番目の要素で表されます (以下同様)。
-  **注:** DLL 関数に渡す引数の前には、いずれかの Visual Basic データ型を配置する必要があります。

DLL 関数で変更できる引数の受け渡し

値が DLL 関数によって変更される引数は、ByRef キーワード を使用して宣言する必要があります。

例

この例では、現在のカーソル位置を取得するために、user32.dll の GetCursorPos 関数を使用しています。

DLL の宣言 :

```
<Dll( "user32.dll" )> Public Interface IUserDll32Functions
    Function GetCursorPos( ByRef point As Point) As Integer
End Interface
```

使用法 :

```
Public Sub Main()
    Dim user32Functions As IUserDll32Functions =
    DllCall.CreateAgentDllCall( Of IUserDll32Functions )()
    Dim point As Point
    user32Functions.GetCursorPos(point)
```

```
MsgBox( "cursor position = (" & point.X & ", " & point.Y & ")" )  
End Sub
```

DLL 関数への文字列引数の受け渡し

DLL 関数に渡している文字列、または DLL 関数から戻される文字列は、デフォルトでは Unicode Strings として処理されます。DLL 関数に ANSI String 引数が必要な場合は、DllFunctionOptions 属性の CharSet プロパティを使用します。

例

```
<Dll( "user32.dll" )> Public Interface IUserDll32Functions  
    <DllFunctionOptions(CharSet:=CharacterSet.Ansi)> Function  
    SendMessageA( _  
        ByVal obj As TestObject, ByVal message As Integer , ByVal wParam  
        As Integer , ByRef lParam As String ) As Integer  
End Interface
```

DLL 呼び出しから String を ByRef 引数として戻した場合、String のサイズが 256 文字以下であれば、デフォルトの動作に従います。戻される String が 256 文字を超えている場合は、作成された String を保持できるだけの長さを持つ、Visual Basic String を渡します。

例

1024 個の空白文字を含む String を作成するには、以下のコードを使用します。

```
Dim longEmptyString = New String ( " "c , 1024 )
```

この String を ByRef 引数として DLL 関数に渡します。すると、この DLL 関数は最大 1024 文字の String を戻します。

関数の戻り値として DLL から String が戻される場合、DLL は DLL 関数 FreeDllMemory を実装し、DLL 関数から戻される C String ポインターを受け入れて、以前に割り当てられたメモリーを解放する必要があります。このような関数が存在しない場合、メモリーはリークされます。

DLL 名のエイリアス設定

DLL 関数に、Visual Basic の予約語と同じ名前が付いている場合、または DLL 関数に名前ではなく序数が付いている場合は、宣言内でこの関数の名前を変更し、エイリアス ステートメントを使用して、宣言した名前と実際の名前をマッピングする必要があります。

例

たとえば、Exit ステートメントは Visual Basic コンパイラーで予約されています。したがって、関数 exit を呼び出すには、次のようにその関数を別の名前で宣言し、エイリアス ステートメントを追加する必要があります。

```
<Dll("mydll.dll")> Public Interface IMyDllFunctions  
    <DllFunctionOptions(Alias:="exit")> Sub MyExit()  
End Interface
```

DLL 関数呼び出しの表記規則

DLL 関数を呼び出す場合は、次に示す呼び出し規則がサポートされています。

- `__stdcall`
- `__cdecl`

DLL 関数を呼び出す場合は、`__stdcall` 呼び出し規則がデフォルトで使用されます。この呼び出し規則は、すべての Windows API DLL 関数で使用されます。

DLL 関数の呼び出し規則を変更するには、`DllFunctionOptions` 属性の `CallingConvention` プロパティを使用します。

例

次のコード例では、`__cdecl` 呼び出し規則を使用して DLL 関数を宣言します。

```
<Dll("msvcrt.dll")> Public Interface IMsVisualCRuntime
  <DllFunctionOptions(CallingConvention:=CallingConvention.Cdecl)>
  Function cos(ByVal input As Double) As Double
End Interface
```

Microsoft ユーザー補助を使用したオブジェクト解決の向上

Microsoft ユーザー補助を、クラスレベルでオブジェクトを簡単に認識するために使用することができます。Internet Explorer や Microsoft アプリケーションのいくつかのオブジェクトには、ユーザー補助を有効にすることで Silk Test Workbench によってより良く認識されるようになるものがあります。たとえば、ユーザー補助を有効にしないと、Silk Test Workbench は Microsoft Word のメニューバーや表示されるタブについて基本的な情報のみを記録します。しかし、ユーザー補助を有効にすると、Silk Test Workbench はそれらのオブジェクトを完全に認識するようになります。

例

ユーザー補助を使用しないと、Silk Test Workbench は `DirectUIHwnd` コントロールを完全に認識できません。これは、このコントロールのパブリックな情報が存在しないためです。Internet Explorer は、2 つの `DirectUIHwnd` コントロールを使用しています。1 つはブラウザ ウィンドウの下部に表示されるポップアップです。このポップアップには、通常、次の情報が表示されます。

- Internet Explorer を既定のブラウザにしたいかどうかを尋ねるダイアログ ボックス。
- ダウンロード オプション（開く、保存、キャンセル）。

Silk Test Workbench でプロジェクトを開始して、`DirectUIHwnd` ポップアップに対してロケータを記録すると、ユーザー補助が無効になっている場合、単一のコントロールのみが表示されます。ユーザー補助を有効にした場合には、`DirectUIHwnd` コントロールを完全に認識した情報が得られます。

ユーザー補助の使用

Win32 では、ジェネリック コントロールとして認識されるコントロールにユーザー補助 サポートが使用されます。Win32 は、コントロールを特定すると、ユーザー補助オブジェクトをコントロールのすべてのユーザー補助の子とともに取得しようとします。

ユーザー補助によって返されるオブジェクトは、`AccessibleControl`、`Button`、`CheckBox` のいずれかのクラスになります。`Button` および `CheckBox` は、そのクラス用に定義されたメソッドとプロパティの通常セットをサポートするので個別に扱われます。ユーザー補助によって返されるすべてのジェネリック オブジェクトの場合、クラスは `AccessibleControl` です。

例

ユーザー補助が有効になる前、アプリケーションのコントロール階層が次のようになっていたとします。

- コントロール
 - コントロール
- ボタン

ユーザー補助を有効にすると、階層は次のようになります。

- コントロール
 - コントロール
 - ユーザー補助コントロール
 - ユーザー補助コントロール
 - ボタン
- ボタン

ユーザー補助の有効化

Win32 アプリケーションをテストしているときに、がオブジェクトを認識できない場合は、最初にユーザー補助を有効にする必要があります。ユーザー補助は、オブジェクトの認識機能をクラス レベルで強化するためのものです。

ユーザー補助を有効にするには、以下の手順を実行します。

1. ツール > オプション をクリックします。オプション ダイアログ ボックスが表示されます。
2. 詳細設定 をクリックします。
3. Microsoft ユーザー補助を使用する オプションを選択します。ユーザー補助が有効になります。

UI オートメーション

Microsoft UI オートメーション (UI オートメーション) は、アプリケーションの UI 要素をプログラム上からアクセス、識別、操作できるようにするためのフレームワークです。UI オートメーション プロバイダー インターフェイスを実装する Windows ベースのアプリケーションに対してテストを実行する場合、UI オートメーションを使うと、アプリケーションのコントロールに対するオブジェクト解決を改善することができます。このようなコントロールのことを、以降では UI オートメーション コントロール と呼びます。



注: Silk Test Workbench は、UI オートメーションを実装した Windows ベースのアプリケーションに対する Microsoft Windows 8 以降のマシン上でのテストをサポートします。


UI オートメーションは、次のテクノロジーを利用したアプリケーションに対するフォールバック サポートとして利用できます。


- Win32
- WPF
- WinForms
- Oracle JavaFX
- QT
- PowerBuilder
- Delphi
- Microsoft Office

たとえば、Silk Test Workbench がアプリケーションのオブジェクトを解決できなかったり、Silk Test Workbench がアプリケーションのすべてのオブジェクトを Control として解決することが原因で、アプリ

ケーションに対するテストを記録できない場合、UI オートメーション サポートを有効化してみることができま

UI オートメーション サポートを記録時に有効にするには、記録を停止して **Microsoft UI オートメーション オプション** を有効にしてから、記録を再開します。詳細については、「」を参照してください。

 **注:** UI オートメーション サポートは、標準のテクノロジー ドメイン固有のサポートをオーバーライドします。UI オートメーション サポートが必要なコントロールに対する操作を完了したら、UI オートメーション サポートをオフにして、標準コントロールに対する操作に戻すこともできます。


 **注:** Java FX アプリケーションをテストする場合は、UI オートメーション サポートをオンにする必要はありません。Silk Test Workbench では、Java FX アプリケーションに対して、デフォルトでオンになります。

UI オートメーション コントロールのサポートするメソッドが、すべてのサポートするテクノロジーの対応するコントロールに対して有効であることを保障するため、Silk Test Workbench API がサポートするのは、これらのコントロールに対して利用できるメソッドやプロパティのサブセットになります。コントロールに対して Silk Test Workbench API で利用できないその他のメソッドやプロパティの呼び出しには、動的呼び出しを使用してください。

UI オートメーション メソッドの動的呼び出し

UI オートメーション コントロールのサポートするメソッドが、すべてのサポートするテクノロジーの対応するコントロールに対して有効であることを保障するため、Silk Test Workbench API がサポートするのは、これらのコントロールに対して利用できるメソッドやプロパティのサブセットになります。コントロールに対して Silk Test Workbench API で利用できないその他のメソッドやプロパティの呼び出しには、動的呼び出しを使用してください。

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型。

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point、Rect など) が含まれます。

- 列挙型。

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合には、Visible、Hidden、Collapsed の文字列値を使用できます。

- .NET 構造体とオブジェクト。

.NET 構造体とオブジェクトパラメータはリストとして渡します。リスト内の要素は、テストアプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。

- その他のコントロール。

コントロールパラメータは TestObject として渡すことができます。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- MSDN が定義する AutomationElement クラスのすべてのパブリック メソッドとプロパティ。詳細については、<http://msdn.microsoft.com/en-us/library/system.windows.automation.automationelement.aspx> を参照してください。
- MSUIA が公開するすべてのメソッドとプロパティ。利用可能なメソッドとプロパティは「パターン」で分類されます。パターンとは、MSUIA 固有の用語です。すべてのコントロールは、いくつかのパターンを実装します。一般的なパターンについての概要およびすべての利用可能なパターンについては、<http://msdn.microsoft.com/en-us/library/ms752362.aspx> を参照してください。カスタムコントロールの開発者は、MSUIA パターン セットを実装することによって、カスタムコントロールのテストサポートを提供できます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。
- すべてのその他の型の場合は文字列。

この文字列表現を取得するには、テスト対象アプリケーションの返された .NET オブジェクトに対して ToString メソッドを呼び出します。

例

この例では、動的呼び出しを使って UIADocument コントロールのスクロール メソッドを呼び出す方法を紹介합니다。Silk Test Workbench の API ではこのようなスクロール メソッドを提供していません。これは、これらのメソッドが、UI オートメーションプロバイダー インターフェイスを実装するすべてのテクノロジの UIADocument コントロールで利用可能ではないためです。

コントロールに対して利用できるメソッドとプロパティを表示するには、以下に示すようなコードを使用できます。

```
' VB .NET code
Dim textBox As UIADocument = mainWindow.Find("//UIADocument")
Dim propertyList = textBox.GetPropertyList()
Dim methodList = textBox.GetDynamicMethodList()
```

この例では、GetPropertyList メソッドが返す *propertyList* には、ScrollPattern.VerticalScrollPercent プロパティが含まれます。

GetDynamicMethodList メソッドが返す *methodList* には、ScrollPattern.ScrollVertical メソッドが含まれます。

動的呼び出しを使うと、次のようなコードで ScrollPattern.ScrollVertical メソッドを呼び出すことができます。

```
textBox.Invoke("ScrollPattern.ScrollVertical", ScrollAmount.SmallIncrement)
```

または、次のようなコードで ScrollPattern.VerticalScrollPercent プロパティを呼び出すことができます。

```
textBox.GetProperty("ScrollPattern.VerticalScrollPercent")
```

UI オートメーションを使ってコントロールを識別するためのロケータ属性

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールに対して、次のロケータ属性がサポートされています。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールを識別するには、スクリプトで *automationId*、*caption*、*className*、*name*、またはすべての動的ロケータ属性を使用できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下のようになります: //

```
UIButton[@automationId="okButton"]
```

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナ上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	// UIButton[@automationId="okButton"]
<i>caption</i>	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//UIButton[@caption="Ok"]
<i>className</i>	UI オートメーション コントロールのクラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 UI オートメーション コントロールから派生したカスタム コントロールを識別するのに役立ちます。	// UIButton[@className='MyCustomButton']

属性の種類	説明	例
name	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	//UIButton[@name="okButton"]

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性を、この表に示した順番に使用して UI オートメーション コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* を使って固有に識別できれば、Silk Test Workbench によるロケータの作成に使用されます。

ユーザーの AUT の UI オートメーション コントロールに対して使用できるその他のカスタム属性を確認するには、**プロパティの検証** ダイアログ ボックスを使用できます。使用するときは、記録する UI オートメーション コントロール上にマウス カーソルを移動し、**Ctrl + Alt** をクリックします。これで、コントロールに対して使用できるプロパティを確認できます。たとえば、アプリケーションによっては、属性 value が有効な場合もあります。

UI オートメーション コントロールでのスクロール

Silk Test Workbench では、UI オートメーション コントロールに応じて、2 種類のスクロール方法とプロパティを提供します。

- 1 つめの種類のコントロールには、それ自体でスクロール可能なコントロールが含まれ、スクロールバーは子として明示的に表示されません。たとえば、コンボ ボックス、ペイン、リスト ボックス、ツリー コントロール、データ グリッド、オート コンプリット ボックスなどがあります。
- 2 つめの種類のコントロールには、それ自体ではスクロール不可能なコントロールが含まれ、スクロール用にスクロールバーが子として表示されます。たとえば、テキスト フィールドがあります。

Silk Test Workbench にこのような違いがあるのは、UI オートメーション コントロールがこの 2 通りの方法でスクロールを実装するためです。

スクロールをサポートするコントロール

この場合、スクロール方法とプロパティは、スクロールバーを含むコントロールで使用できます。したがって、Silk Test Workbench ではスクロールバー オブジェクトは表示されません。

例

以下のコマンドでは、リスト ボックスが一番下までスクロールされます。

```
listBox.SetVerticalScrollPercent(100)
```

以下のコマンドでは、リスト ボックスが 1 ユニットずつ下方へスクロールされます。

```
listBox.ScrollVertical(ScrollAmount.SmallIncrement)
```

スクロールをサポートしないコントロール

この場合、スクロールバーが表示されます。コントロール自体で可能なスクロール方法とプロパティはありません。水平スクロールバーと垂直スクロールバーの各オブジェクトを使用すると、対応する API 関数でパラメータとして増分または減分、または最終位置を指定することでコントロール内をスクロールできます。増分または減分として *ScrollAmount* 列挙の値を使用できます。詳細については、MSUIA のドキュメントを参照してください。最終位置は、オブジェクトの位置に関連し、アプリケーション設計者によって定義されます。

例

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが 15 の位置までスクロールされます。

```
textBox.UIAVerticalScrollBar().ScrollToPosition(15)
```

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが一番下までスクロールされます。

```
textBox.UIAVerticalScrollBar().ScrollToMaximum()
```

UI オートメーション使用時の制限事項

UI オートメーション使用時における既知の制限事項を以下に示します。

UI オートメーション サポート使用時に IME が正しく処理されない

UI オートメーション サポートを有効にすると、Silk Test Workbench の記録/再生時に IME を伴うキーの入力が正しく処理されません。

UI オートメーション サポートを使ったテスト時のトラブルシューティング

Microsoft Windows 7 で記録した UI オートメーション コントロールを含むスクリプトを Microsoft Windows 8 以降で再生できない理由

Microsoft Windows 7 以前で UI オートメーション コントロールを含んだスクリプトを記録すると、Microsoft Windows 8 以降での再生に失敗する場合があります。これは、Microsoft がオートメーションの仕組みを変更し、UI オートメーションの動作が Windows のバージョン間で異なることが原因です。

たとえば、アプリケーションのある UI オートメーション コントロールでは、Microsoft Windows 7 では automationId 属性に値が設定されていますが、Microsoft Windows 10 では値が設定されていません。

このような場合、Microsoft Windows の新しいバージョンでスクリプトを記録しなおすことを Micro Focus では推奨しています。

Silk Test Workbench での Unicode コンテンツのサポートの概要

Open Agent は、Unicode 対応済みです。つまり、Open Agent は、2 バイト (ワイド) 言語を認識できません。

Silk Test Workbench を使用して、中国語、韓国語、日本語 (漢字) などの 2 バイト言語や、それらを組み合わせたコンテンツを含んだアプリケーションをテストできます。

Open Agent は、以下をサポートします。

- Windows のローカライズ版。
- 国際化キーボードとネイティブ言語の入力方式エディター (IME)。
- テストケース、メソッドなどにパラメータとして国際化文字列を渡す、および文字列の比較。
- 複数の形式でのテキスト ファイルの読み書き：ANSI、Unicode、UTF-8。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

Silk Test Workbench を使用した 2 バイト文字のテストを行う前に

国際化されたアプリケーション、特に 2 バイト文字を含んだアプリケーションをテストするのは、英語 (1 バイト文字) のみを含んだアプリケーションをテストするよりも複雑です。国際化アプリケーションのテストにおいては、オペレーティング システムのサポートから、言語パック、フォント、IME の動作、さらに複合した言語など、さまざまな問題を理解する必要があります。

Silk Test Workbench を使用してアプリケーションのテストを始める前に、以下を確認する必要があります。

- 必要なローカライズ OS、地域の設定、必要な言語パックがテスト対象アプリケーション (AUT) の要求を満たしているか。
- AUT を表示するのに必要なフォントがインストールされているか。
- データ入力に IME が必要なアプリケーションをテストする場合、適切な IME がインストールされているか。

テキスト解決のサポート

テキスト解決メソッドとイメージ解決メソッドを使用して、オブジェクト解決で識別できない、高度にカスタマイズされたコントロールを含むテスト アプリケーションを便利に操作できます。座標ベースのクリックの代わりにテキスト クリックを使用し、コントロール内に指定されたテキスト文字列をクリックできます。

たとえば、次の表の 2 行目の最初のセルを選択することをシミュレートできます。


CustomerName	FirstOrder	ID	IsActive	CreditCard
Bob Villa	01.01.2008	0	<input checked="" type="checkbox"/>	MasterCard
Brian Miller	02.01.2008	1	<input type="checkbox"/>	Visa
Caral Rudd	03.01.2008	2	<input checked="" type="checkbox"/>	American Ex...
Dan Rundgren	04.01.2008	3	<input type="checkbox"/>	MasterCard
Devie Yingstein	05.01.2008	4	<input checked="" type="checkbox"/>	Visa

セルのテキストを指定すると、次のコードが生成されます：

```
'VB code  
table.TextClick("Brian Miller")
```

テキスト解決メソッドは、次のテクノロジー ドメインでサポートされます。

- Win32
- WPF
- Windows Forms
- Java SWT と Eclipse
- Java AWT/Swing

 **注:** Java アプレット、および Java バージョンが 1.6.10 より前の Swing アプリケーションの場合、テキスト解決は追加設定なしでサポートされます。Direct3D をサポートしない Java バージョン 1.6.10 以降の Swing アプリケーションの場合は、アプリケーションの起動時に次のコマンドライン要素を追加する必要があります。


```
-Dsun.java2d.d3d=false
```


例：

```
javaw.exe -Dsun.java2d.d3d=false -jar mySwingApplication.jar
```

Direct3D をサポートする Java アプレットや Swing アプリケーションでは、テキスト解決はサポートされません。


- Internet Explorer
- WebDriver ベースのブラウザー


 **注:** テキスト解決は、画面に表示されていないコントロールに対しては機能しません。たとえば、スクロールして画面外にあるテキストに対しては、テキスト解決を使用できません。

 **注:** 対象のテキストで使用されたフォントがテストが実行されるマシン上にインストールされていない場合、テキスト解決が機能しない場合があります。

WebDriver ベースのブラウザー

テキスト解決メソッドは、BrowserWindow オブジェクトと DomElement オブジェクトに対して実行できます。

 **注:** <canvas> 要素に描画されたテキストに対しては、テキスト解決は機能しません。

 **注:** ::before や ::after などの CSS 擬似要素によって追加されたコンテンツに対しては、テキスト解決は機能しません。

テキスト解決メソッド

Silk Test Workbench では、AUT がスクリーン上に描画するテキストを処理してテストを実行できるように、次のメソッドを提供しています。

TextCapture コントロール内のテキストを返します。子コントロールのテキストも返します。

TextClick コントロール内の指定テキストをクリックします。テキストが検出されるか、同期オプションで定義できるオブジェクト解決タイムアウトに達するまで待機します。


TextRectangle コントロール内の特定テキストの矩形、またはコントロールの領域を返します。

TextExists コントロール内またはコントロールの領域内に特定テキストが存在するかどうかを判断します。

テキスト解決メソッドでは、部分的に一致する単語よりも完全に一致する単語が優先されます。Silk Test Workbench では、完全に一致する単語の前に部分的に一致する単語が画面に表示されていても、部分的に一致する単語よりも完全に一致した単語の出現が先に解決されます。完全に一致する単語がない場合は、部分的に一致する単語が画面に表示される順序で使用されます。

TextClick、TextRectangle、TextExists メソッドは、その内部で TextCapture を使用してアプリケーションから表示されたテキストを取得し、テキストを処理します。TextCapture メソッドは、2 種類の方法で実装されています。Silk Test Workbench は、テスト対象アプリケーションの種類に応じて実装を使い分けます。

- ネイティブ Windows アプリケーション (WPF、WinForms、Java アプリケーション)、および Internet Explorer の場合は、Silk Test Workbench は Windows API のテキスト描画関数をフックして、アプリケーションが画面に描画するテキストを取得します。
- Google Chrome、Mozilla Firefox、Microsoft Edge、Apple Safari の場合は、Silk Test Workbench は JavaScript ベースの方法を使って、ブラウザーが描画したテキストを取得します。

 **注:** これら 2 種類の実装方法の違いにより、同じ Web アプリケーションでも、使用するブラウザーに応じて異なるテキストを Silk Test Workbench が返す場合があります。

例

ユーザー インターフェイスには、テキスト「*the hostname is the name of the host*」が表示されているとします。次のコードでは、画面上では *host* の前に *hostname* が表示されていますが、*hostname* ではなく *host* がクリックされます。

```
'VB code  
control.TextClick("host")
```

次のコードでは、2 番目の出現箇所であることを指定することで、単語 *hostname* 中の部分文字列 *host* がクリックされます。

```
'VB code  
control.TextClick("host", 2)
```


カスタム コントロール

Silk Test Workbench では、カスタム コントロールを扱うときに、以下の機能がサポートされます。

- 動的呼び出しを使用すると、テスト対象アプリケーション (AUT) 内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を Silk Test Workbench で直接実行できます。
- クラス マッピング 機能によって、カスタム コントロール クラスの名前を標準 Silk Test クラスの名前にマップできます。このようにすると、標準 Silk Test クラスでサポートされる機能をテストで使用できます。


Silk Test Workbench は、次のテクノロジー ドメインに対する UI のカスタム コントロールの管理をサポートします。

- Win32
- Windows Presentation Foundation (WPF)
- Windows Forms
- Java AWT/Swing
- Java SWT
- AUT にコードを追加して、カスタム コントロールをテストできます。
- **カスタム コントロールの管理** ダイアログ ボックスを使用して、ロケータで利用できるカスタム コントロールの名前を指定したり、カスタム コントロールを操作する再利用可能なコードを作成することができます。

 **注:** カスタム コントロールでは、Click、TextClick、TypeKeys などのメソッドだけが、Silk Test Workbench で記録できます。Apache Flex アプリケーションをテストする場合を除き、カスタム コントロールのカスタム メソッドは記録できません。

動的呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある setTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

 **注:** DOM 要素のメソッドを動的に呼び出すことはできません。

動的呼び出しに関するよくある質問

このセクションでは、カスタム コントロールをテストするために動的にメソッドを呼び出すときの質問を示します。

Invoke メソッドを使用して呼び出せるメソッド

特定のテスト オブジェクトに対して、Invoke メソッドを使用して呼び出せるすべてのメソッドのリストを取得するには、GetDynamicMethodList を使用します。リストを表示するには、コンソールに出力したり、デバッガーで表示することなどができます。

呼び出しで複雑なオブジェクトが返されることが期待されるときに単純な文字列が返される理由

Invoke メソッドは単純なデータ型のみを返すことができます。複雑な型は文字列として返されます。Silk Test Workbench は ToString メソッドを使用して、戻り値の文字列表現を取得します。個々のメソッドを呼び出し、最初のメソッドの呼び出しで返される複雑なオブジェクトのプロパティを読み取るには、Invoke ではなく、InvokeMethods を使用します。

複数の InvokeMethods 呼び出しを使用するときにはスクリプトを単純化する方法

スクリプトで大量の InvokeMethods を使用すると、すべてのメソッド名を文字列として渡し、すべてのパラメータをリストとして渡す必要があるため、複雑になります。このような複雑なスクリプトを単純化するには、InvokeMethods を通じてコントロールを操作するのではなく、AUT の実際のコントロールを操作する静的メソッドを作成します。詳細については、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」を参照してください。

テスト対象アプリケーションにコードを追加してカスタムコントロールをテストする

Windows Forms アプリケーションまたは WPF アプリケーションをテストし、複雑なカスタム コントロールまたは Invoke および InvokeMethods メソッドを使用するだけではテストできないカスタム コントロールをテストする場合は、テスト対象アプリケーション (AUT) の実際のコントロールを操作する静的メソッドを作成し、このコードを AUT に追加できます。

AUT にコードを追加することのメリットは、AUT のコードで、動的呼び出しメソッドによるメソッド呼び出しのリフレクション形式ではなく、通常のメソッド呼び出しを使用してコントロールを操作できるという点です。そのため、コードを作成する時に、コード補完と IntelliSense を使用できます。その後、AUT のコードを単純な呼び出しで呼び出し、該当するコントロールをパラメータとして渡すことができます。

AUT にコードを追加するには、次の方法があります。

- AUT でコードをコンパイルします。実装は簡単ですが、意図しない AUT の変更を行うこととなります。
- テスト スクリプトの LoadAssembly メソッドを使用して、実行時にコードを AUT に挿入します。AUT でコードをコンパイルする場合よりも作業は多くなりますが、挿入されたコードはテスト コードの近くに配置されます。LoadAssembly は、WPFWindow クラスおよび FormsWindow クラスで使用できます。

例 : UltraGrid Infragistics コントロールのテスト

この例では、UltraGrid コントロールの内容を取得する方法を示します。UltraGrid コントロールは、Infragistics が提供する NETAdvantage for Windows Forms ライブラ

りに含まれています。ライブラリの試用版を <http://www.infragistics.com/products/windows-forms/downloads> からダウンロードできます。

UltraGridUtil クラスを作成するには、以下の操作を実行します。

1. 新しいテスト スクリプトを作成します。新しいスクリプト AUTExtensions を呼び出します。



注: AUT は、.NET バージョン 4 以降を使用する必要があります。.NET スクリプトをコンパイルすると .NET 4.0 アセンブリが作成されますが、このアセンブリは使用している .NET バージョンが同じかより新しい AUT でのみロードできます。AUT でバージョン 4.0 以前の .NET を使用している場合は、Visual Studio を使用して、UltraGridUtil クラスを含むクラス ライブラリ プロジェクトを作成してください。

2. 必要な依存関係への参照をスクリプトに追加します。たとえば、Infragistics バージョン 12.2 の場合、次のアセンブリへの参照が必要です。

- Infragistics4.Shared.v12.2
- Infragistics4.Win.UltraWinGrid.v12.2
- Infragistics4.Win.v12.2

AUT で使用している Infragistics のバージョンが不明な場合は、Microsoft の **Process Explorer** ツールを使用して、AUT にロードされているアセンブリを確認できます。

- a. AUTExtensions スクリプトの内容を次のコードに置き換えます。

```
' VB code
Public Class UltraGridUtil

    Public Shared Function GetContents(ultraGrid As
    Infragistics.Win.UltraWinGrid.UltraGrid) As List(Of List(Of String))
        Dim contents = New List(Of List(Of String))
        For Each row In ultraGrid.Rows
            Dim rowContents = New List(Of String)
            For Each cell In row.Cells
                rowContents.Add(cell.Text)
            Next
            contents.Add(rowContents)
        Next
        Return contents
    End Function

End Class
```



注: Shared 修飾子によって、GetContents メソッドが静的メソッドになります。

3. 再生中に、AUT にアセンブリをロードします。

- 既存のテスト スクリプトを開くか、新しいテスト スクリプトを作成します。
- AUTExtensions スクリプトへの参照を追加します。
- 次のコードをテスト スクリプトに追加します：

```
' VB code
mainWindow.LoadAssembly(GetType(UltraGridUtil).Assembly.Location)
```



注: Silk Test Workbench によって、各スクリプトが個別の .NET アセンブリにコンパイルされます。

4. 挿入したコードの静的メソッドを呼び出して、UltraGrid の内容を取得します :

```
'VB code
Dim ultraGrid = mainWindow.Control("@automationId='my grid'")
Dim contents As IList =
mainWindow.Invoke("UltraGridUtil.GetContents", ultraGrid)
```

AUT へのコードの追加に関するよくある質問

このセクションでは、カスタムコントロールをテストするために AUT にコードを追加するときの質問を示します。

LoadAssembly メソッドを使用して AUT に挿入したコードが AUT で更新されない理由

AUT 内のコードが、LoadAssembly メソッドを使用して AUT に挿入したコードによって置き換えられない場合、アセンブリがすでに AUT にロードされている可能性があります。アセンブリをアンロードすることはできないため、AUT を閉じてから、再開する必要があります。

メソッドを呼び出すと入力引数の型が一致しない理由

何らかのメソッドを呼び出したときに、入力引数の型が一致しないことを示すエラーが表示される場合は、呼び出すメソッドは見つかりましたが、引数が正しくありません。スクリプトで正しいデータ型を使用していることを確認します。

スクリプトで LoadAssembly メソッドを使用してアセンブリを AUT にロードする場合にこのエラーが発生するもう 1 つの理由として、AUT が使用するバージョンとは異なるサードパーティライブラリのバージョンに対してアセンブリが作成されている可能性があります。この問題を修正するには、スクリプトで参照されているアセンブリを変更します。AUT で使用されているサードパーティライブラリのバージョンが不明な場合は、Microsoft の **Process Explorer** ツールを使用できます。

アセンブリをコピーできないときにコンパイル エラーを修正する方法

LoadAssembly メソッドで AUT にコードを追加しようとしたときに、次のコンパイル エラーが発生することがあります。

```
Could not copy '<assembly_name>.dll' to '<assembly_name>.dll'. The process cannot access the file.
```

このコンパイル エラーは、アセンブリがすでに AUT にロードされていて、上書きできないために発生します。

このコンパイル エラーを修正するには、AUT を閉じて、再度スクリプトをコンパイルします。最初から問題が発生しないようにするには、一時ディレクトリなどを使用してアセンブリをスクリプト内にコピーし、コピーしたアセンブリを LoadAssembly メソッドで挿入します。例 :

```
' VB code
Dim typeThatContainsYourStaticMethod = GetType (UltraGridUtil)
Dim assemblyToLoadIntoAUT = typeThatContainsYourStaticMethod.Assembly.Location
Dim tempFile = IO.Path.GetTempFileName()
IO.File.Copy(assemblyToLoadIntoAUT, tempFile, True )
mainWindow.LoadAssembly(tempFile)
```

Apache Flex カスタム コントロールのテスト

Silk Test Workbench では、Apache Flex カスタム コントロールのテストがサポートされています。ただし、デフォルトでは、Silk Test Workbench は、カスタム コントロールの個別のサブコントロールを記録および再生することはできません。

カスタム コントロールをテストする場合、以下のオプションが存在します。

- 基本サポート

基本サポートでは、動的呼び出しを使用して、再生中にカスタム コントロールと対話します。作業量が少なく済むこのアプローチは、テスト アプリケーションにおいて、Silk Test Workbench が公開しないカスタム コントロールのプロパティおよびメソッドにアクセスする場合に使用します。カスタム コントロールの開発者は、コントロールのテストを容易にすることのみを目的としたメソッドおよびプロパティをカスタム コントロールに追加することもできます。ユーザーは、動的呼び出し機能を使用してこれらのメソッドやプロパティを呼び出すことができます。

基本サポートには以下のような利点があります。

- 動的呼び出しでは、テスト アプリケーションのコードを変更する必要がありません。
- 動的呼び出しを使用することによって、ほとんどのテストのニーズを満たすことができます。

基本サポートには以下のような短所があります。

- ロケーターには、具体的なクラス名が組み込まれません (たとえば、Silk Test Workbench では「//FlexSpinner」ではなく「//FlexBox」と記録されます)。
- 記録のサポートが限定されます。
- Silk Test Workbench では、イベントを再生できません。

例を含む動的呼び出しの詳細については、「*Apache Flex* メソッドの動的呼び出し」を参照してください。

• 高度なサポート

高度なサポートでは、カスタム コントロールに対して、特定のオートメーション サポートを作成できます。この追加のオートメーション サポートによって、記録のサポートおよびより強力な再生のサポートが提供されます。高度なサポートには以下のような利点があります。

- イベントの記録と再生を含む、高レベルの記録および再生のサポートが提供されます。
- Silk Test Workbench では、カスタム コントロールが他のすべての組み込み Apache Flex コントロールと同様に処理されます。
- Silk Test Workbench API とシームレスに統合できます。
- Silk Test Workbench では、ロケーターで具体的なクラス名が使用されます (たとえば、Silk Test Workbench では「//FlexSpinner」と記録されます)。

高度なサポートには以下のような短所があります。

- 実装作業が必要です。テスト アプリケーションを変更し、Open Agent を拡張する必要があります。

カスタム コントロールの管理

Silk Test Workbench が専用サポートを提供していないカスタム コントロールに対応するカスタム クラスを作成できます。カスタム クラスを作成すると、以下の利点があります。

- スクリプトおよびビジュアル テストのロケーターが効率化されます。
- カスタム コントロールと対話するための再利用可能コードを簡単に記述できます。

例 : UltraGrid Infragistics コントロールのテスト

カスタム グリッド コントロールが Silk Test Workbench で汎用クラス Control として認識されるとします。Silk Test Workbench のカスタム コントロール サポートを使用すると、以下の利点があります。

カスタム コントロール クラス名をロケーターで使用できるため、オブジェクトの認識率が高まります。

複数のオブジェクトが Control として認識されることがあります。ロケーターには、特定のオブジェクトを識別するためのインデックスが必要です。たとえば、オブジェクトはロケーター //Control[13] を使用して識別できます。このコントロールのカスタム クラス (クラス UltraGrid など) を作成する場合は、ロケーター //UltraGrid を

スクリプト内の
コントロール
に、再利用可能
な再生操作を実
行できます。

使用できます。カスタム クラスを作成することによって、テスト対象アプリケーションが変更された場合にオブジェクト識別子が変わりやすい、大きな数字のインデックスを使用する必要がなくなります。

カスタム クラスを使用している場合、ユーザー インターフェイスにカスタム コントロールを指定すると生成されるクラスであるカスタム クラスに以下のコードを追加することで、グリッドのコンテンツをメソッド内に取り込む動作をカプセル化できます。

通常は、以下のいずれかの方法で、メソッドをカスタム コントロール クラスに実装できます。

- Click、TypeKeys、TextClick、および TextCapture などのメソッドを使用できます。
- AUT のオブジェクトで動的にメソッドを呼び出せます。
- AUT に追加したメソッドを動的に呼び出せます。これは、この例で説明されている手法です。

以下のコードを使用して、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」の例で定義されている静的メソッドを呼び出すことができます。メソッド GetContents が、生成されたクラス UltraGrid に追加されます。

```
' VB code
Partial Public Class UltraGrid

    Public Function GetContents() As IList
        Return
        Invoke("UltraGridUtil.GetContents", Me)
    End Function

End Class
```

クラスをカスタム コントロールとして定義すると、Dialog クラスのように、すべての組み込みクラスの場合と同じ方法でそのクラスを使用できます。

```
' VB code
Dim ultraGrid As UltraGrid =
mainWindow.UltraGrid("@automationId='my
grid'")
Dim contents = ultraGrid.GetContents()
```

カスタム コントロールのサポート

Silk Test Workbench は、次のテクノロジ ドメインに対する UI のカスタム コントロールの管理をサポートします。


- Win32

- Windows Presentation Foundation (WPF)
- Windows Forms
- Java AWT/Swing
- Java SWT


Silk Test Workbench が専用サポートを提供していないカスタム コントロールに対応するカスタム クラスを作成するには、以下を実行します。

1. ツール > **カスタム コントロールの管理** をクリックします。**カスタム コントロールの管理** ダイアログ ボックスが開きます。
2. **カスタム コントロール用 .NET スクリプト** フィールドで、任意の名前を入力するか、**参照** をクリックして、カスタム コントロールを含めるスクリプト を選択します。

カスタム コントロール用の .NET スクリプトは、Common プロジェクトに保存されます。すべての新しいデータベースに対して、デフォルトの .NET スクリプトは *CustomControlScript* という名前です。このスクリプトは、ユーザーが最初にデータベースにログインしたときに自動的に作成されます。


 **注:** .NET スクリプトは必須です。**カスタム コントロール用 .NET スクリプト** フィールドでスクリプトが選択されていない場合、**OK** ボタンは使用できません。ボタンを有効にするには、スクリプトを選択します。既存のデータベースの場合、Common プロジェクトの書き込み権があり、カスタム コントロール セット用の .NET スクリプトがない場合には、Silk Test Workbench は Common プロジェクトに新しいスクリプトを保存しようとします。*CustomControlScript* という名前のスクリプトが既に存在し、Common プロジェクトの書き込み権がある場合、Silk Test Workbench はスクリプトの名前の終わりに番号を追加してスクリプトを保存しようとします (*CustomControlScript2*、*CustomControlScript3* など)。


3. 新しいカスタム クラスを作成するテクノロジー ドメインのタブをクリックします。
4. コントロールを自動的に選択する場合は、**識別** をクリックして Silk Test Workbench を識別モードに切り替えると、通常通りにコントロールを選択できるようになります。**カスタム コントロールのクラス名** と **Silk Test 基本クラス** フィールドは、特定されたコントロールに基づいて、現在のタグに入力されます。それ以外の場合、以下の **追加** ステップを続けます。
5. **追加** をクリックします。新しい行がカスタム コントロールのリストに追加されます。
6. カスタム コントロールを手動でリストに追加するように選択した場合は、以下を実行します。
 - a) **Silk Test 基本クラス** 列で、クラスの取得元となる既存の基本クラスを選択します。
このクラスは、ご使用のカスタム コントロールのタイプに最も一致率が高くなければなりません。
 - b) **カスタム コントロール クラス名** 列に、マップしているクラスの完全修飾クラス名を入力します。
たとえば、Infragistics.Win.UltraWinGrid.UltraGrid です。クラス名には、ワイルドカード ? および * を使用できます。
 - c) **Win32** で作業していて、クラスの参照方法を変更する場合は、次の列のチェック ボックスをオンにします。チェック ボックスをオフにした場合、名前はデフォルトで **Silk Test 基本クラス** と同じになります。**Silk Test クラス** 列が有効になります。
 - d) **Silk Test クラス** 列で、**F2** をクリックし、クラスの参照に使用する名前を入力します。
この名前は、ロケーターに表示されます。たとえば、//Control[13] でなく //UltraGrid を入力します。

 **注:** 有効なクラスを追加すると、そのクラスは **Silk Test 基本クラス** リストで使用できるようになります。追加したクラスは、基本クラスとして再使用できます。

7. Win32 アプリケーションの場合のみ：**クラスの宣言を使用する** 列で、値を **False** に設定して、カスタム コントロール クラスの名前を標準 Silk Test クラスの名前に単純にマップします。
カスタム コントロール クラスを標準 Silk Test クラスにマップすると、テストの際に標準 Silk Test クラスでサポートされている機能を使用できます。カスタム コントロール クラスのクラス宣言を追加して使用する場合は、この値を **True** にします。
8. **OK** をクリックします。
9. スクリプトの場合のみ：

- a) カスタム コントロール用のクラスにカスタム メソッドおよびプロパティを追加します。
- b) スクリプト内で新しいクラスのカスタム メソッドおよびプロパティを使用します。

 **注:** カスタム メソッドおよびプロパティは記録されません。

 **注:** スクリプト ファイル内のカスタム クラスまたは基本クラスの名前を変更しないでください。スクリプト内に生成されたクラスを変更した場合、予期しない動作を起こすことがあります。カスタム クラスにプロパティおよびメソッドを追加する場合にのみスクリプトを使用してください。それ以外の変更をカスタム クラスに加える場合は **カスタム コントロールの管理** ダイアログ ボックスを使用してください。

カスタム コントロール オプション


ツール > **カスタム コントロールの管理**。

Silk Test Workbench は、次のテクノロジー ドメインに対する UI のカスタム コントロールの管理をサポートします。

- Win32
- Windows Presentation Foundation (WPF)
- Windows Forms
- Java AWT/Swing
- Java SWT


カスタム コントロール用 .NET スクリプト で、新しいカスタム クラスをその中に生成するスクリプト ファイルを定義します。

カスタム コントロール用の .NET スクリプトは、Common プロジェクトに保存されます。すべての新しいデータベースに対して、デフォルトの .NET スクリプトは *CustomControlScript* という名前です。このスクリプトは、ユーザーが最初にデータベースにログインしたときに自動的に作成されます。

 **注:** .NET スクリプトは必須です。**カスタム コントロール用 .NET スクリプト** フィールドでスクリプトが選択されていない場合、**OK** ボタンは使用できません。ボタンを有効にするには、スクリプトを選択します。既存のデータベースの場合、Common プロジェクトの書き込み権があり、カスタム コントロール セット用の .NET スクリプトがない場合には、Silk Test Workbench は Common プロジェクトに新しいスクリプトを保存しようとします。*CustomControlScript* という名前のスクリプトが既に存在し、Common プロジェクトの書き込み権がある場合、Silk Test Workbench はスクリプトの名前の終わりに番号を追加してスクリプトを保存しようとします (*CustomControlScript2*、*CustomControlScript3* など)。

カスタム コントロール クラスを標準 Silk Test クラスにマップすると、テストの際に標準 Silk Test クラスでサポートされている機能を使用できます。次の **カスタム コントロール オプション** が使用できます。

オプション	説明
Silk Test 基本クラス	自分のクラスの派生元として使用する既存の基本クラスを選択します。このクラスは、ご使用のカスタム コントロールのタイプに最も一致率が高くなければなりません。
Silk Test クラス	クラスの参照に使用する名前を入力します。この名前は、ロケーターに表示されます。
カスタム コントロールのクラス名	マッピングされているクラスの完全修飾クラス名を入力します。クラス名には、ワイルドカード ? および * を使用できます。
クラスの宣言を使用する	このオプションは Win32 アプリケーションの場合のみ使用できます。デフォルト値は False で、カスタム コントロール クラスの名前が標準 Silk Test クラスの名前にマップされることを意味します。カスタム コントロール クラスのクラス宣言を追加して使用する場合は、この設定を True にします。

 **注:** 有効なクラスを追加すると、そのクラスは **Silk Test 基本クラス** リストで使用できるようになります。追加したクラスは、基本クラスとして再使用できます。

例 : UltraGrid Infragistics コントロールのオプションの設定

UltraGrid Infragistics コントロールをサポートするには、次の値を使用します。

オプション	値
Silk Test 基本クラス	Control
Silk Test クラス	UltraGrid
カスタム コントロールのクラス名	Infragistics.Win.UltraWinGrid. UltraGrid


実行時間の計測

Timer クラスが提供するメソッドやプロパティを使用して、テストの実行にかかる時間を計測できます。詳細については、「*Timer* クラス」を参照してください。

特に、これらのメソッドとプロパティは、Silk Performer から呼び出されるテスト実行の計測に使用できるという利点があります。Silk Test Workbench と Silk Performer の統合についての詳細は、『[Silk Performer ヘルプ](#)』を参照してください。

テスト実行の遅延

テスト対象アプリケーションによっては、UI でのアプリケーション データの読み込みに多くの時間を必要とするため、テストの再生に必要なオブジェクトのロードが時間内に終わらない場合があります。このような AUT でテストの再生を正しく行うには、操作を実行する前にオブジェクトの存在を確認したり、操作の実行前にスリープを挿入する必要があります。

 **注:** Micro Focus では、テストにスリープを追加することは基本的には推奨していません。たいていの場合、オブジェクトが利用可能かどうかを Silk Test Workbench が自動的に検出するので、スリープはテストのパフォーマンスを落とす結果になるためです。

1. オブジェクトが AUT で利用可能かどうかを確認するには、Exists メソッドを使用します。
たとえば、INPUT ボタンが利用可能になるまで 6 秒間待機する場合は、テストスクリプトに次の行を追加します。

```
'VB .NET code  
browserWindow.Exists("//INPUT", 6000)
```

2. コントロールの操作を実行する前にスリープを追加するには、Sleep メソッドを使用します。
たとえば、6 秒間スリープする場合は、テストスクリプトに次の行を追加します。

```
'VB .NET code  
System.Threading.Thread.Sleep(6000)
```

テストのデバッグ

ビジュアルテストおよびスクリプトのデバッグプロセスについて説明します。

ビジュアルテストのデバッグ

再生中のエラーは、テストアプリケーションの変更、ビジュアルテストの不適切なステップフロー、環境の変更など、さまざまな要因により発生します。デバッグ機能を使用してこれらのエラーを迅速に診断、修正することで、ビジュアルテストのメンテナンスの労力を最小限に抑え、チームによるテスト作業の効率を上げることができます。

ビジュアルテストのデバッグを使用すると、ビジュアルテストの再生を一時的に中断して、その再生の管理、検証、リセット、段階的な再生を実行することができます。これにより、エラーの発生箇所を正確に識別できます。デバッグ中にステップを編集できるため、多くの場合、ステップを再実行する前にその中のエラーを解決できます。

選択したポイントでのビジュアルテスト再生の停止

ビジュアルテストの再生中に特定のポイントで停止して、デバッグモードに入るようにビジュアルテストを構成できます。再生中にビジュアルテストが停止するように設定するポイントのことを、ブレークポイントと呼びます。ブレークポイントにより、デバッグしやすいようにビジュアルテストの再生を管理でき、ビジュアルテストが失敗する箇所を切り離すのに役立ちます。ブレークポイントを使用すると、ビジュアルテストにエラーがない場合でも、ビジュアルテストがどのように再生されるかを分析できます。ブレークポイントを設定すると、ブレークポイントを設定したステップを実行する前に再生が停止し、デバッグモードに入ります。

ブレークポイントの設定により、確認したい特定のポイントまでビジュアルテストを再生できるようになります。ブレークポイントは、再生を停止したい特定のステップに設定します。たとえば、変数に含まれるデータの状態をローカル変数ウィンドウで確認したいポイントに設定します。再生は最初のブレークポイントまで実行され、ビジュアルテストに続行を指示するまで中断されます。

1. ブレークポイントを設定するビジュアルテストを開きます。
2. 再生を停止するステップを選択します。
3. **デバッグ > ブレークポイントの設定/クリア** を選択します。ステップ番号の横の列に、ブレークポイントアイコンが表示されます。再生は、設定されたブレークポイントまで実行されます。再生がブレークポイントで停止すると、ビジュアルテストのブレークポイントステップが黄色で強調表示され、そのステップを指す黄色の矢印が表示されます。

ビジュアルテストがデバッグモードで表示され、再生を段階的に実行したり、ステップの再生を制御したり、再生をビジュアルテスト内の特定のポイントで開始または停止したりできるようになります。



ヒント: ブレークポイントが設定されたステップを選択し、**F9** を押すと、選択したステップからブレークポイントが削除されます。**Ctrl+Shift+F9** を押すと、すべての開いているビジュアルテストとVB.NET スクリプトのすべてのブレークポイントが削除されます。

選択したポイントからのビジュアルテスト再生の段階的実行

デバッグ中に、ブレークポイントによって再生が一時停止された場所から、1行ごとにビジュアルテストを実行できます。

1. ビジュアル テストの特定のステップにブレークポイントを設定します。
2. F5 を押して、ビジュアル テストを再生します。再生は、ブレークポイントまで実行されると停止し、デバッグ モードになります。**テスト ステップ** ペインで、次に再生されるステップが黄色で強調表示されます。
3. F8 を押して、強調表示されたステップを再生します。
ステップが正常に実行されたら、再生は再びデバッグ モードになり、次に再生されるステップが黄色で強調表示されます。ステップが正常に再生されなかった場合は、**再生エラー** ダイアログ ボックスが開きます。
4. F8 を押して、ビジュアル テスト内の残りのステップの再生を段階的に実行します。再生は、設定されたブレークポイントまで実行されます。ビジュアル テストがデバッグ モードで表示され、再生を段階的に実行したり、ステップの再生を制御したり、再生をビジュアル テスト内の特定のポイントで開始または停止したりできるようになります。ビジュアル テストの段階的実行中、ローカル変数 ウィンドウを使用して、テスト実行の重要なポイントにおける変数の使用状況をスナップショットとして取得できます。デバッグ中に、特定のステップから再生を開始するように、ビジュアル テストを構成することもできます。

ビジュアル テストの再生におけるステップ実行の制御

ビジュアル テストのデバッグ中に、またはブレークポイントで、ビジュアル テスト内で次に実行するステップを制御できます。実行ポイントを移動すると、他のステップを実行せずに、ビジュアル テスト内の別の箇所にある選択したステップから再生を再開できます。

実行ポイント/次のステートメントの設定 コマンドは、選択したステップに実行ポイントを設定します。この機能は、デバッグ中のブレークポイントでのみ使用可能です。

現在のプロシージャ内でステートメントを再実行する場合や、ビジュアル テスト内で再生したくないステップをスキップする場合は、**実行ポイント/次のステートメントの設定** コマンドを使用します。この機能を使用すると、ビジュアル テストのデバッグ中にフローを制御してエラーを認識しやすくできるため、エラーの診断と修正に要する時間を短縮することができます。

1. デバッグ中、または再生中のブレークポイントで、実行する次のステップを選択します。
ビジュアル テストで任意のステップを選択します。
2. **デバッグ > 実行ポイント/次のステートメントの設定** を選択します。選択したステップに黄色の実行矢印が移動し、そのステップが黄色で強調表示されます。
3. F5 を押して選択したステップから再生を再開するか、F8 を押して選択したステップのみを再生します。選択したステップが実行されると、再生が停止し、Silk Test Workbench がデバッグ モードに戻ります。実行ポイントは次のステップに移動します。

デバッグ中のビジュアル テストの編集

デバッグ中にビジュアル テストを編集および保存することができます。変更を保存すると、そのビジュアル テストの新しいバージョンが作成され、アクティブ ビジュアル テストになります。

デバッグ中にビジュアル テストを段階的に実行して、変更を加えることができます。加えた変更は、再生時に適用されます。デバッグ中に、ビジュアル テストで以下の変更を実行することができます。

- フラグの追加。
- 既存のステップのプロパティの編集。
- テストステップのコピーと貼り付け。
- ブレークポイントの設定と解除。
- 他のビジュアル テストに挿入されるビジュアル テストの編集。たとえば、*visualtest2* を *visualtest1* に挿入する場合、*visualtest2* を制限なしで編集できます。

デバッグモードでのビジュアルテストの段階的実行

デバッグモードでは、ビジュアルテストを1ステップずつ再生できます。ステップングと呼ばれるこの方法により、再生中にステップを段階的にトレースして、テストステップの実行順序、特定のステップによって返されるデータ、およびステップが正常に実行されるかどうかを確認できます。

デバッグモードでは、以下のコマンドを使用して再生の実行を制御します。

ステップイン (F8) 1つずつステップを再生します。**ステップイン** は各ステップのトレースに有効であり、また再生しているビジュアルテストに挿入された別のビジュアルテストやスクリプトにもステップインします。挿入された各ビジュアルテストやスクリプトについても、一度に1ステップまたは1行ずつ実行されます。

ステップイン はテストの詳細分析に有効であり、変数の使用やターゲットアプリケーションとの対話に対する各ステップの影響を確認することができます。

ステップオーバー (Shift + F8) 別のビジュアルテストに挿入されたビジュアルテスト全体またはスクリプト全体を、1つのステップとして実行します。**ステップオーバー** は、デバッグモードの再生中に、ビジュアルテストまたはスクリプトを1ステップとして再生する場合に使用します。これにより、挿入されたビジュアルテスト全体またはスクリプト全体が再生されます。挿入されたビジュアルテスト全体またはスクリプト全体が再生されると、元のビジュアルテスト内の次のステップで再生が停止し、デバッグモードになります。

別のビジュアルテストまたはスクリプトを再生するステップ以外のステップで**ステップオーバー**を使用すると、**ステップイン**を使用した場合と同じになります。次のステップのみが実行されたあと、再生は停止し、再びデバッグモードになります。

ステップアウト (Ctrl + Shift + F8) 別のビジュアルテストから再生しているビジュアルテストまたはスクリプト内の残りのステップをすべて実行したあと、元のビジュアルテスト内の次のステップで再生が一時停止します。

ステップアウト は、別のビジュアルテストに挿入されたビジュアルテスト内のステップまたはコード行で再生が停止されているときに、残りのビジュアルテストを再生して元のビジュアルテストに戻る場合に使用します。挿入されたビジュアルテストまたはスクリプトの残りの部分の再生が実行されたら、元のビジュアルテスト内の次の行で再生が停止し、再びデバッグモードになります。

カーソルまで実行 (Ctrl + F8) 再生を一時停止するステップを選択できます。これにより、ビジュアルテストまたはスクリプト内の選択したセクションを「ステップオーバー」することができます。

カーソルまで実行 は、ビジュアルテストまたはスクリプトを再生して、実行時エラーが発生したポイントの直前で停止する場合に使用します。これにより、ブレークポイントを挿入することなく特定の行またはステートメントで再生を停止できます。再生が停止したら、別のデバッグオプションのいずれかを使用して再生を継続できます。

カーソルから実行 現在選択されているテストステップからビジュアルテストを再生します。

ステップングコマンドには、**デバッグ**メニューからアクセスします。

ビジュアルテスト内でのエラー処理による確実な再生

アプリケーションのテストは、テスト対象アプリケーション内のエラーを明らかにするものですが、Silk Test Workbenchを使用したソフトウェアアプリケーションのテストでは、これらのエラーが原因でビジュアルテストの再生を完了できない場合があります。たとえば、テストステップで、再生中に検出されないTextFieldにテキストを入力しようとすると、オブジェクトが見つからないというエラーが発生し、再生が停止します。ユーザーはデバッグモードに切り替えて、エラーの原因を特定することができます。

エラー処理を設定すると、再生エラーをデバッグする必要がなくなります。エラー処理を設定するには、ビジュアルテストでエラーの検出ステップとその処理方法を設定します。エラー処理を設定することによって、ビジュアルテストを最後まで再生でき、再生中に発生したエラーの情報を得ることができます。

[プロパティ] ペインを使用して、エラーに自動的に対処するようにビジュアルテストを設定

ビジュアルテストは、再生エラーを予測して自動的に対処するように構成することができます。ビジュアルテストのステップ内でエラーを処理するように設定するか、特定のエラーを処理するステップを含む別のビジュアルテストを開くように設定できます。エラーを処理するようにビジュアルテストを設定するには、エラー処理ステップを作成します。

1. エラー処理を設定するビジュアルテストを開きます。
2. エラー処理ステップを作成する場所の直前のステップを選択します。
3. **挿入** > **エラー処理** を選択します。選択したステップのあとにエラー処理ステップが作成されます。エラー処理ステップのプロパティが**プロパティ** ペインに表示されます。
4. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
5. エラー処理ステップの**プロパティ** ペインでツールバー ボタンをクリックして、プロパティをカテゴリ別にグループ化するか、アルファベット順に並べ替えます。
6. **エラー時の移動先** プロパティを更新し、再生設定とその値を割り当てます。
エラーハンドラの**実行する操作** プロパティが**移動**である場合、ビジュアルテスト内でエラーハンドラの移動先になるラベルを指定します。

エラー処理ステップは、再生中にエラーが発生すると実行されます。

テストロジックデザイナーを使用してビジュアルテストを設定し、エラーに自動的に対処

ビジュアルテストは、再生エラーを予測して自動的に対処するように構成することができます。ビジュアルテストのステップ内でエラーを処理するように設定するか、特定のエラーを処理するステップを含む別のビジュアルテストを開くように設定できます。ビジュアルテストでエラーが処理されるように設定するには、**テストロジックデザイナー** ウィザードを使用します。

1. エラー処理を設定するビジュアルテストを開きます。
2. エラー処理ステップを作成する場所の直前のステップを選択します。
3. **挿入** > **テストロジック** > **エラー処理** を選択します。**テストロジックデザイナー** ウィザードが開き、**よろこ** ページが表示されます。
4. **次へ** をクリックします。**ロジックの種類を選択** ページが開きます。
5. **ビジュアルテストの再生中のエラー (再生エラー)** をクリックし、**次へ** をクリックします。**エラー処理の構築** ページが開きます。
6. **再生エラーが発生した場合、常に以下の操作を行う** リストで、以下のいずれかの操作を選択します。

デフォルトの再生エラーダイアログを表示します **再生エラー** ダイアログボックスを表示し、そこでそのエラーに対して実行する操作を決定します。**再生エラー** ダイアログボックスを使用してデバッグモードに入り、ビジュアルテストでエラーを診断します。

現在のビジュアルテストの再生を終了します エラーが発生したときに、ビジュアルテストの再生を終了します。ビジュアルテストのその他のステップは再生されません。再生が終了すると、**再生完了** ダイアログボックスが表示され、そこで再生の完了時に実行する操作を決定します。

エラーが発生したステップの再実行 **資産** プロパティに指定されているビジュアルテストを再生してから、エラーが発生したステップの実行を試行します。**資産** プロパティにビジュアルテストが指定されていない場合、単にエラーが発生したステップの再実行を試行します。**再試行の回数**

- 生を再試行します** プロパティには、再生を試行する回数を指定できます。デフォルト値は、5 です。指定した再試行の回数に到達した時点でステップ実行に失敗すると、再生エラーが生成されます。
- 次のステップで再生を再開します** **資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップの直後のステップの実行を試みます。エラーの原因となったステップは、再実行されません。
- ラベルへ移動します** **資産** プロパティに指定されているビジュアル テストを再生し、**ラベル** プロパティに指定されている事前定義のラベル ステップに移動して、再生を続行します。**移動** を使用すると、ビジュアル テストで、再生中に発生したエラーと同様のエラーが発生する可能性のあるステップをスキップして、テストの別のセクションに移動できます。
- 他のビジュアル テストを最初に再生します** 指定した別のビジュアル テストが再生されたあと、次に実行される操作をユーザーが指定します。別のテスト実行のあとに実行される操作を決定するには、テストの終了、エラーが発生したステップを再試行、ラベルへ移動、またはエラーが発生した元のテストの再開を選択します。

7. **ラベルへ移動します** を指定した場合は、**移動先のラベルを選択する** リスト ボックスから エラー ハンドラが移動するラベルを選択します。
8. **他のビジュアル テストを最初に再生します** を指定した場合は、以下のステップを実行します。
 - a) **<ビジュアル テストの挿入>** をクリックし、再生する**資産**を選択して、**OK** をクリックします。
 - b) **次の処理** リスト ボックスから、次に行う操作を選択します。
 - ラベルへ移動します** を指定した場合は、**移動先のラベルを選択する** リスト ボックスから エラー ハンドラが移動するラベルを選択します。
9. **次へ** をクリックします。 **要約** ページに、新しいエラー処理ステップのテスト ステップの説明が表示されます。
- 10 **終了** をクリックします。

ビジュアル テストの印刷

ビジュアル テストを印刷して、テストの内容を保存します。

1. 印刷するビジュアル テストを開きます。
2. **ファイル > 印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
3. リストからプリンタを選択します。
4. 省略可能：**プロパティ** をクリックすると、余白や印刷の向きなどの印刷ページ設定を変更できます。
5. 省略可能：**印刷範囲** セクションで、ドキュメント全体を印刷するか、または選択したページのみを印刷するかを指定します。
6. 省略可能：**部数** フィールドで、印刷する部数を指定します。
7. 省略可能：**部単位で印刷する** チェック ボックスをオンにすると、すべてのコピーの対応するページをまとめて印刷します。
 - このチェック ボックスは、複数の部数を印刷するように選択した場合にのみ有効になります。
8. 印刷するステップを指定します。
 - **テスト ステップ** ペインに表示されているすべてのステップを印刷する場合は、**現在の表示** をクリックします。
 - **テスト ステップ** ペインで選択されているステップだけを印刷する場合は、**選択** をクリックします。
9. 印刷出力をカスタマイズするには、**オプション** をクリックして、印刷出力に含める情報の種類を選択します。
 - 以下のオプションがあります。

- 以下を含める** 含める内容の種類を指定します。
- **ビジュアルテストの要約**：ビジュアルテストの名前、プロジェクト、作成者、作成日、バージョン、説明などのビジュアルテストの要約を含めます。この要約は、印刷出力の最初のページに表示されます。
 - **最新の結果の要約**：ビジュアルテストの最新結果の要約ページを含めます。この要約は、印刷出力の最後のページに表示されます。
 - **グループ画面**：現在のダイアログボックスまたは Web ページのメインコンテンツを含むグループ画面のスナップショットを印刷出力に含めます。グループ画面に続くテストステップは、現在のダイアログボックスまたは Web ページのコンテンツで発生する単一の操作に対応します。グループ画面は、**テストステップ** ペインの Using ステップに対応します。グループ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。
 - **無効なステップ**：無効化されたステップを含めます。この情報は、印刷出力で灰色のテキストで表示されます。

- 各ステップに含める** 各ステップに含める内容を指定します。
- **画面**：各ステップの関連画面スナップショットを含めます。個々の画面は (ステップ画面) に対応し、現在のダイアログボックスまたは Web ページでの特定の操作に関連付けられます。ステップ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。
 - **説明**：各ステップの説明を含めます。ステップの説明は、**プロパティ** ペインの **ステップの説明** プロパティに表示されるように、印刷出力に示されます。
 - **フラグの詳細**：フラグの説明、作成者、更新者、更新日、割り当てられたユーザーなどのフラグの詳細を含めます。

- 以下のプロパティを含める** 各ステップに含めるプロパティを指定します。
- **すべてのプロパティ**：各ステップのすべてのプロパティを含めます。

デフォルト デフォルトの印刷オプションを選択するには、このボタンをクリックします。

- 10**省略可能 : **プレビュー** をクリックして、結果情報がどのように印刷されるかを表示します。**プレビュー** ウィンドウが開きます。
- 11****OK** をクリックします。

スクリプトのデバッグ

再生中の実行時エラーは、さまざまな要因により発生します。これらの要因には、テストアプリケーションの変更、テストスクリプトのコーディングの誤り、または環境の変更などがあります。スクリプトのデバッグによってこれらのエラーを迅速に診断、修正することで、スクリプトのメンテナンスの労力を最小限に抑え、より効率的なチームによるテスト作業を推進することができます。

スクリプトのデバッグを使用すると、開発環境でのスクリプトの再生を一時的に中断して、その再生の管理、検証、リセット、段階的実行、継続を行えます。

Silk Test Workbench には **出力** ウィンドウがあり、自動処理コードの確認やユーザー定義の追加出力の追加を行うことができます。たとえば、`Console.Write` および `Console.WriteLine` コマンドを使用して、追加実行を出力できます。

さらに、ローカル変数ウィンドウを使用してドリルダウンし、スクリプト内のすべての変数の値を表示することもできます。

スクリプトの信頼性

アプリケーションのテストは、テスト対象アプリケーション内のエラーを明らかにします。ソフトウェアアプリケーションのテスト時に、これらのエラーが原因でスクリプトの実行を完了できない場合があります。

テストを成功させるには、スクリプトの再生を確実に完了させる必要があります。同期化の問題、あるいは予期されるまたは予期されないアプリケーション エラーによって、再生が完了する前にスクリプトが失敗することがあります。Silk Test Workbench に組み込まれているオプション設定とスクリプト作成機能により、同期化とアプリケーションの問題を調整して、任意のテスト環境でスクリプトを確実に再生できるようにします。

スクリプトを再生しているときに、アプリケーションの応答時間の遅延に起因するエラーが発生すると、アプリケーションとそのアプリケーションをテストするスクリプトとの同期が取れなくなります。これらのエラーは、Silk Test Workbench のオプション設定を調整してスクリプトとテスト対象アプリケーション間の同期を最適化することで処理できます。

デバッグ モードでのスクリプトの段階的実行

デバッグ モードでは、スクリプトを一度に 1 行ずつ再生できます。ステップングと呼ばれるこの方法により、テスト担当者は再生中にコードを段階的にトレースして、ステートメントの実行順序を確認できるため、デバッグに役立ちます。

ステップングにより、デバッグ モードでスクリプトを段階的に実行し、その実行を詳細に検証することができます。

デバッグ モードでは、以下のコマンドを使用して再生の実行を制御します。

ステップ イン (F8)

一度に 1 ステートメントずつスクリプトを実行します。**ステップ イン** はコードの各行のトレースに有効であり、関数または埋め込まれているスクリプトにもステップ インします。各関数または埋め込みスクリプトも、一度に 1 行ずつ実行されます。

ステップ イン はテストの詳細分析に有効であり、変数の使用やターゲット アプリケーションとの対話に対する各ステートメントの影響を確認することができます。



注: VB .NET スクリプトのデバッグ中に反復子 (*Iterator*) にステップ インすることはできません。反復子は実行されますが、デバッグは実行時に反復子が処理されていることを表示しません。反復子の内側にブレイクポイントを設定すると、実行はブレイクポイントで停止しますが、反復子内のパラメータやローカル変数は、**ローカル変数** ウィンドウに表示されません。詳細については、「[反復子](#)」を参照してください。

ステップ オーバー (Shift + F8)

各プロシージャを 1 つのステートメントとして実行します。関数を構成する各行を実行せずに、関数に直接移動する場合には、[ステップ イン] ではなくこのコマンドを使用します。

ステップ アウト (Ctrl + Shift + F8)

プロシージャ内の残りすべてのコードを 1 つのステートメントのように実行し、そのプロシージャを最初に呼び出したプロシージャ内の次のステートメントに移ります。

カーソルまで実行 (Ctrl + F8)

再生を一時停止するステップを選択できます。これにより、ビジュアル テストまたはスクリプト内の選択したセクションを「ステップ オーバー」することができます。

カーソルまで実行 は、ビジュアル テストまたはスクリプトを再生して、実行時エラーが発生したポイントの直前で停止する場合に使用します。これにより、ブレイクポイントを挿入することなく特定の行またはステートメントで再生を停止できます。再生が停止したら、別のデバッグ オプションのいずれかを使用して再生を継続できます。

実行ポイント/次のステートメントの設定 (Ctrl + F9) 現在ポイントがある場所のステートメントから、途中のステートメントを実行せずステートメントの に実行を続けます。

ステップング コマンドには、**デバッグ** メニューからアクセスします。

選択したポイントからのスクリプト再生の段階的実行

デバッグ中に、ブレークポイントによって再生が一時停止された場所から、スクリプトを実行できます。


1. スクリプトの特定の行にブレークポイントを設定します。
2. F5 を押して、スクリプトを再生します。再生は、ブレークポイントまで実行されると停止し、デバッグモードになります。
3. F8 を押して、スクリプトを次のブレークポイントまで再生します。
コードが正常に実行されたら、再生は再びデバッグ モードになり、次に再生されるコードが黄色で強調表示されます。コードが正常に再生されなかった場合は、**再生エラー** ダイアログ ボックスが開きます。
4. F8 を押して、スクリプト内の残りのコードの再生を段階的に実行します。再生は、設定されたブレークポイントまで実行されます。スクリプトがデバッグ モードで表示され、再生を段階的に実行したり、ステップの再生を制御したり、再生をスクリプト内の特定のポイントで開始または停止したりできるようになります。スクリプトの段階的実行中、**変数の監視** ウィンドウを使用して、テスト実行の重要なポイントにおける変数の使用状況をスナップショットとして取得できます。デバッグ中に、コード内の特定の行から再生を開始するように、スクリプトを構成することもできます。

スクリプト デバッグ中の行の実行の制御

スクリプトのデバッグ中にフローを制御して、エラーを認識しやすくし、エラーの診断と修正に要する時間を短縮することができます。

実行ポイント/次のステートメントの設定 コマンドは、スクリプト内の選択した行に実行ポイントを設定します。この機能は、デバッグ中のブレークポイントでのみ使用可能です。

現在のプロシージャ内でステートメントを再実行する場合や、スクリプト内で再生したくないステートメントをスキップする場合は、**実行ポイント/次のステートメントの設定** コマンドを使用します。この機能を使用すると、スクリプトのデバッグ中にフローを制御してエラーを認識しやすくできるため、エラーの診断と修正に要する時間を短縮することができます。

1. デバッグ中、または再生中のブレークポイントで、以下のいずれかのステップを実行します。
 - 黄色い矢印をクリックして次に再生するコード行へドラッグします。
 - 次に再生する行をクリックしてカーソルを配置してから、**デバッグ > 実行ポイント/次のステートメントの設定** を選択します。
-  **注:** スクリプトで、次のステートメントは、現在のステートメントと同じプロシージャの行にのみ設定できます。
2. F5 を押して選択した行から再生を再開するか、F8 を押して選択した行のみを再生します。選択したステップが実行されると、再生が停止し、Silk Test Workbench がデバッグ モードに戻ります。実行ポイントは次の行に移動します。

スクリプト例 - エラー処理

以下のスクリプトは、エラー処理を含むスクリプトを作成する方法を示しています。

基本的なエラー処理をスクリプトに含めるには、Catch ex As Exception ステートメントに続いて例外のテスト対象となるオブジェクトとパラメータをスクリプトに含めます。たとえば、メモ帳アプリケーションのフォントの種類をテストするには、以下のコードを入力します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop
```

```

Public Sub Main()
    With _desktop.Window("@caption='Untitled - Notepad'")
        .TextField().TypeKeys("test", 0, False)
        .MenuItem("@caption='Font']").Select()
    Try
        With .Dialog("@caption='Font1'") ' throws an exception because such a
dialog does not exist
            .ComboBox("@caption='Font style:']").Select("Regular")
            .ComboBox("@caption='Size:']").Select("14")
            .PushButton("@caption='Cancel']").Select()
        End With
    Catch ex As Exception
        With .Dialog("@caption='Font'")
            .ComboBox("@caption='Font style:']").Select("Regular")
            .ComboBox("@caption='Size:']").Select("14")
            .PushButton("@caption='Cancel']").Select()
        End With
    End Try
End With
End Sub
End Module

```



注: この例は、基本的なエラー処理の実装方法を示すことを意図しています。このコードを正しく実装するには、アプリケーション構成を追加して、テスト アプリケーションを必要に応じて調整する必要があります。このサンプル コードをスクリプトに貼り付けても、アプリケーション構成が存在しないため、エラーが発生します。

スクリプトへのコメントの挿入

出力 ウィンドウを使用して、自動処理コードを確認し、ユーザー定義の追加出力を付加して、スクリプトに関する補足情報を提供します。たとえば、コメントを追加して、スクリプトに関する補足情報を提供できます。スクリプトを再生すると、コメントが **出力** ウィンドウに表示されます。

1. スクリプトでは、`Console.Write` および `Console.WriteLine` コマンドを使用してコメントを追加できます。
`Comment` メソッドでは、文字列がパラメータとして処理されます。この文字列の内容は、スクリプトの再生後に **出力** ウィンドウに保存されます。
2. スクリプトを再生します。 **出力** ウィンドウに、自動処理コードおよび追加したコメントを表示できます。

選択したポイントでのスクリプト再生の停止

スクリプトの再生中に特定のポイントで停止して、デバッグ モードに入るようにスクリプトを構成できます。再生中にスクリプトが停止するように設定するポイントのことを、ブレークポイントと呼びます。ブレークポイントにより、デバッグしやすいように再生を管理でき、スクリプトが失敗する箇所を切り分けるのに役立ちます。ブレークポイントを使用すると、スクリプトにエラーがない場合でも、スクリプトがどのように再生されるかを分析できます。ブレークポイントを設定すると、再生はブレークポイントで停止し、デバッグ モードに入ります。

ブレークポイントの設定により、確認したい特定のポイントまでスクリプトを再生できるようになります。ブレークポイントは、再生を停止したい特定の行に設定します。たとえば、変数に含まれるデータの状態を **変数の監視** ウィンドウで確認したいポイントに設定します。再生は最初のブレークポイントまで実行され、スクリプトに続行を指示するまで中断されます。

1. ブレークポイントを設定するスクリプトを開きます。

- 再生を停止する行をクリックします。
- デバッグ > ブレークポイントの設定/クリア**. 行番号の横の列に、ブレークポイントアイコンが表示されます。再生は、設定されたブレークポイントまで実行されます。再生がブレークポイントで停止すると、スクリプトのブレークポイント行が黄色で強調表示され、その行を指す黄色の矢印が表示されます。

デバッグでは、再生中に行の実行を制御できます。また、特定の行まで再生を実行したり、特定の行から再生を開始したりするように、スクリプトを構成することもできます。



ヒント: ブレークポイントが設定された行を選択し、F9 を押すと、選択した行からブレークポイントが削除されます。Ctrl+Shift+F9 を押すと、スクリプトからすべてのブレークポイントが削除されます。

スクリプトの印刷

確認やトラブルシューティングに役立てるために、コード ウィンドウに表示されているコードを印刷できます。

- 印刷するスクリプトを開きます。
- ファイル > 印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
- リストからプリンターを選択するか、**プリンター検索** をクリックして他のプリンターを検索します。
- 省略可能：**ファイルへ出力** チェック ボックスをオンにすると、PDF などのファイルへ出力されます。
- 省略可能：**詳細設定** をクリックすると、余白や印刷の向きなどの印刷ページ設定を変更できます。
- 省略可能：**印刷範囲** セクションで、ドキュメント全体を印刷するか、または選択したページのみを印刷するかを指定します。
以下のいずれかのオプションをクリックします。
 - すべて**：ドキュメント全体を印刷します。
 - 選択した部分**：選択したコードを印刷します。
 - ページ指定**：指定したページを印刷します。
 - 現在のページ**：現在のページを印刷します。
- 省略可能：**部数** フィールドで、印刷する部数を指定します。
- 省略可能：**部単位で印刷する** チェック ボックスをオンにすると、すべてのコピーの対応するページをまとめて印刷します。
このチェック ボックスは、複数の部数を印刷するように選択した場合にのみ有効になります。
- 印刷** をクリックします。

テストの再生と結果の分析

テストの再生および再生結果の分析の手順について説明します。

ビジュアルテストの再生

ビジュアルテストを記録したら、いつでもそれを再生できます。テストステップは、上から下に順番に実行されます。再生中、ビジュアルテストは、記録または追加された操作に従ってテストアプリケーションを操作します。これらの操作は画面で確認できます。操作は、そのアプリケーションで可能な最短の応答速度で実行されます。ビジュアルテストは、手動での実行に比べはるかに短時間で再生できます。

ビジュアルテストは、オブジェクトの画面上の位置に関係なくオブジェクトレベルで記録されるため、再生もオブジェクト指向です。つまり、Silk Test Workbench はオブジェクトと直接対話します。そのため、環境やテストアプリケーションに変更があった場合でも、ビジュアルテストは正しく再生されます。

ビジュアルテストを再生するたびに、そのテストに記録されたすべての操作が Silk Test Workbench によって繰り返され、検証などの記録または追加された操作が実行されます。

ビジュアルテストの再生

ビジュアルテストを再生すると、記録または追加した操作が実行されます。

1. 次のいずれか 1 つのステップを行います：

- **開始画面** の **資産** セクションの **最新** リストで、再生するビジュアルテストの **ビジュアルテスト 'プロジェクト名 - テスト名' の再生** をクリックします。
- **アセットブラウザ** で、再生するビジュアルテストを右クリックして、**再生** をクリックします。
- **ビジュアルナビゲータ** で **操作 > 再生** を選択し、アクティブなビジュアルテストを再生します。

再生 ダイアログボックスが開きます。

2. 省略可能：テスト結果の説明を **結果の説明** フィールドに入力します。

3. **結果オプション** リストから、このテストの結果の保存方法を選択します。

以下のオプションのいずれかを選択します。

新しい実行番号で結果を保存 ビジュアルテストまたはスクリプトの再生の結果は、新しい増分実行番号で保存されます。たとえば、新規の結果は実行番号 1 で保存されます。同じビジュアルテストまたはスクリプトの後続の結果は、実行番号 2、3、4 など保存されます。これはデフォルトの設定です。

現在の実行番号に結果を追加 現在の再生の結果は、最新の実行番号の結果に付加されます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。

既存の実行番号を新しい結果で上書き 現在の再生の結果で、最新の既存の実行番号の結果が上書きされます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。

この結果を保存しない ビジュアルテストまたはスクリプトを再生したときに結果は作成されません。


4. 省略可能：**オプション** をクリックして、結果オプションを変更します。

詳細については、「再生結果オプションの設定」を参照してください。

5. 省略可能：**ローカル変数ウィンドウの表示** チェックボックスをオンにすると、再生中に **ローカル変数** ウィンドウが表示されます。

Silk Test Workbench は、再生が終わると、**ローカル変数** ウィンドウを非表示にします。

6. **OK** をクリックします。
7. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。


 **注:** 複数のアプリケーションが現在のスクリプトに対して設定されている場合、**ブラウザーの選択** ダイアログ ボックスは表示されません。

8. 省略可能: スクリプトを停止する場合は、**操作 > 停止** を選択するか、ツールバーの **停止** をクリックするか、両方の **Shift** キーを同時に押します。現在のステップで再生を一時停止する場合は、**操作 > 一時停止** を選択するか、ツールバーの **一時停止** をクリックするか、**Alt + F11** キーを押します。ビジュアル テストが再生されるときに、テスト アプリケーションに対して実行される各操作を確認できます。

ビジュアル テスト内に値が割り当てられていない受け渡し変数が見つかったら、**ビジュアル テスト変数** ダイアログ ボックスが表示され、これらの変数に値を割り当てることができます。

再生中にエラーが発生すると、**再生エラー** ダイアログ ボックスが開きます。

9. 再生が完了したら、**再生完了** ダイアログ ボックスを使用して、実行する操作を決定します。

 **注:** 再生中にエラーが発生した場合は、エラー メッセージが表示されます。


再生中、**再生** ダイアログ ボックスを非表示にするには、**このダイアログを再表示しない** チェック ボックスをオンにします。これにより、全般オプションの **[再生] ダイアログの表示** オプション設定が上書きされます。**再生** ダイアログ ボックスを表示するには、この設定を使用します。

ビジュアル テスト内でのビジュアル テストの実行


ビジュアル テストでは、別のビジュアル テストを呼び出して実行したり、再生したりできます。これにより、アプリケーションのテストに対するモジュール式アプローチが可能になり、テストの実行をより制御できるようになります。ビジュアル テスト内で別のビジュアル テストを実行する場合は、テスト サイトの整合性を保ち、テスト対象のターゲット アプリケーションが正しい初期再生状態になるようにすることが重要です。


この手順を使用して、ドライバ テストを作成することもできます。

1. 別のビジュアル テストを挿入するビジュアル テストを開きます。
2. テストの挿入箇所の直後にあるステップを選択します。
3. **挿入 > ビジュアル テスト** をクリックします。**ビジュアル テストの参照** ダイアログ ボックスが開きます。
4. すべての資産を表示するか、作成した資産だけを表示するかを選択します。

 **注:** 列見出し (**名前**、**プロジェクト**、**説明**、**更新日**) をクリックして、その内容を基準にビジュアル テストのリストをアルファベット順に並べ替えることができます。

5. 挿入するビジュアル テストを選択し、**OK** をクリックします。

 **注:** 現在のプロジェクトとすべての参照プロジェクトのビジュアル テストがリストに表示されます。

 **ヒント:** **資産の選択** リストに多くのビジュアル テストが表示されている場合、**名前フィルタ** テキスト フィールドに挿入したいビジュアル テストの名前を入力してから、ビジュアル テストをクリックすると簡単です。

Silk Test Workbench によって、選択したステップの下にステップが挿入されます。挿入されたステップによって、選択したビジュアル テストが呼び出されます。

再生中、上記のステップが実行されると、呼び出し元のビジュアル テスト内の次のステップが実行される前に、挿入されたビジュアル テストが最後まで再生されます。ビジュアル テストが別のビジュアル テストに挿入されたあと、再生に使用される設定を構成できます。


再生すると、テスト対象アプリケーションは、必要に応じて設定、フック、開始され、再生セッションの終了時までフックされたままになります。たとえば、呼び出し元のビジュアル テストにメモ帳用のアプリ

ケーション構成があり、挿入されたビジュアル テストに電卓用のアプリケーション構成がある場合には、次のようになります。

- 呼び出し元のビジュアル テストの実行の開始時に、メモ帳がフックされ、開始されます。
- 挿入されたビジュアル テストの実行の開始時に、電卓がフックされ、開始されます。メモ帳はフックされたままです。
- 挿入されたビジュアル テストの実行を停止しても、メモ帳と電卓の両方ともフックされたままになります。
- 再生セッションの終了時、つまり、呼び出し元のビジュアル テストの実行を停止すると、メモ帳と電卓の両方のフックが解除されます。

[ビジュアル テスト変数] ダイアログ ボックス

このダイアログ ボックスを使用すると、再生中のビジュアル テストで値が割り当てられていない変数が渡された場合に、値を割り当てることができます。このダイアログ ボックスは、再生中に、ビジュアル テストの <<開始>> ステップで定義されたパラメータまたは変数に値がない場合のみ表示されます。

 **注:** 変数名は、「st_」で始めることはできません。


ビジュアル テスト変数 ダイアログ ボックスには、以下の列で構成されるリストが含まれています。

ビジュアル テスト変数名: このビジュアル テスト内の値が割り当てられていないビジュアル テスト変数の名前が表示されます。この変数は、再生中にこのビジュアル テストを実行する他のビジュアル テストにも渡されます。この列のデータは表示専用で変更はできません。

型: ビジュアル テスト変数のデータ型が表示されます。この列のデータは表示専用で変更はできません。変数の型を以下に示します。

テキスト

変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

 **注:** テキスト データ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

数値 (Long)

変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。

数値 (Double)

変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。

ブール値 (True/False)

変数の値は True または False です。

数値 (Long Long)

64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙

この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

値: リストされたビジュアル テスト変数の値です。ビジュアル テスト変数の値を入力します。値は型に一致している必要があります。たとえば、型が数値 (Long) の場合、値は -2,147,483,648 から 2,147,483,647 の数値にする必要があります。入力した値が型と一致しない場合、または値を入力しなかった場合は、デフォルト値が使用されます。

- 数値 (Long)、数値 (Double)、数値 (Long Long)、および列挙の場合、デフォルト値は 0 です。
- テキストの場合、デフォルト値は NULL 文字列です。
- ブール値の場合、デフォルト値は False です。

スクリプトの再生

スクリプトを記録したら、いつでもそれを再生できます。通常、スクリプト内のコード行は上から下の順に実行されます。スクリプトの再生中、スクリプト内の操作に従ってテスト アプリケーションが操作されます。これらの操作は画面で確認できます。操作は、そのアプリケーションで可能な最短の応答速度で実行されます。これにより、手動での実行に比べはるかに短時間でテストを実行できます。

スクリプトは、オブジェクトの画面上の位置に関係なくオブジェクト レベルで記録されるため、再生もオブジェクト指向です。つまり、Silk Test Workbench は Windows のオブジェクトと直接対話します。そのため、環境やテスト アプリケーションに変更があった場合でも、スクリプトは正しく再生されます。

スクリプトの再生

スクリプトを再生すると、記録または記述された各操作が実行されます。

1. スクリプトを開きます。
2. **操作 > 再生** を選択するか、ツールバーの **再生** をクリックします。 **再生** ダイアログ ボックスが開きます。
3. 省略可能：テスト結果の説明を **結果の説明** フィールドに入力します。
4. **結果オプション** リストから、このテストの結果の保存方法を選択します。
以下のオプションのいずれかを選択します。


新しい実行番号で結果を保存 ビジュアル テストまたはスクリプトの再生の結果は、新しい増分実行番号で保存されます。たとえば、新規の結果は実行番号 1 で保存されます。同じビジュアル テストまたはスクリプトの後続の結果は、実行番号 2、3、4 などでも保存されます。これはデフォルトの設定です。

現在の実行番号に結果を追加 現在の再生の結果は、最新の実行番号の結果に付加されます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。


既存の実行番号を新しい結果で上書き 現在の再生の結果で、最新の既存の実行番号の結果が上書きされます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。

この結果を保存しない ビジュアル テストまたはスクリプトを再生したときに結果は作成されません。


5. 省略可能：**オプション** をクリックして、結果オプションを変更します。
詳細については、「再生結果オプションの設定」を参照してください。
6. 省略可能：**ローカル変数ウィンドウの表示** チェック ボックスをオンにすると、再生中に **ローカル変数** ウィンドウが表示されます。
Silk Test Workbench は、再生が終わると、**ローカル変数** ウィンドウを非表示にします。
7. **OK** をクリックします。
8. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

 **注：** 複数のアプリケーションが現在のスクリプトに対して設定されている場合、**ブラウザーの選択** ダイアログ ボックスは表示されません。

9. 省略可能：スクリプトを停止する場合は、**操作 > 停止** を選択するか、ツールバーの **停止** をクリックするか、両方の **Shift** キーを同時に押します。現在のステップで再生を一時停止する場合は、**操作 > 一時停止** を選択するか、ツールバーの **一時停止** をクリックするか、**Alt + F11** キーを押します。

 **注:** 再生中にエラーが発生した場合は、エラーメッセージが表示されます。

10再生が完了したら、**再生完了** ダイアログ ボックスを使用して、実行する操作を決定します。

 **注:** 再生中にエラーが発生した場合は、エラーメッセージが表示されます。

再生中、**再生** ダイアログ ボックスを非表示にするには、**このダイアログを再表示しない** チェック ボックスをオンにします。これにより、全般オプションの **[再生] ダイアログの表示** オプション設定が上書きされます。**再生** ダイアログ ボックスを表示するには、この設定を使用します。

再生中のオートメーション操作の表示

出力 ウィンドウにオートメーション操作を表示して、スクリプトを確認またはトラブルシューティングします。

1. **表示** > **出力** を選択します。**出力** ウィンドウが開きます。
2. **ツール** > **オプション** をクリックします。**オプション** ダイアログ ボックスが開きます。
3. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
4. **全般** をクリックします。
5. **出力ウィンドウにオートメーション操作を表示する** リスト ボックスから、**はい** を選択します。

スクリプトを再生して、関連するオートメーション操作を表示します。


テストの並列実行


複数の Workbench プロセスを使用して、複数のブラウザーやモバイル デバイスに対してテストを並列に実行することができます。たとえば、CI サーバーや Silk Central からテストを実行するときに、この機能を使用できます。

デフォルトでは、Silk Test は、次のブラウザーやプラットフォームでの並列テストをサポートします。

- Google Chrome
- Mozilla Firefox
- 以下のプラットフォーム上での Web、ネイティブ、ハイブリッド アプリ
 - 物理 Android デバイス
 - Android エミュレータ
 - 物理 デバイス

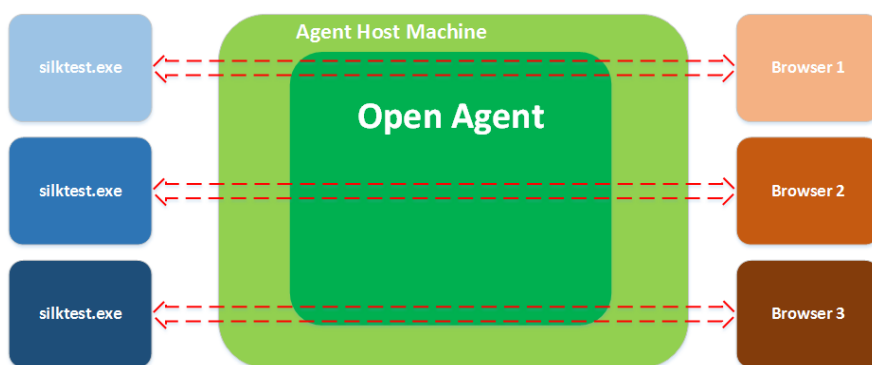
テストの並列再生を無効にするには、環境変数 `SILKTEST_ENABLE_PARALLEL_TESTING` を `false` に設定します。

 **注:** 並列テストを有効にすると、Open Agent はテストの実行プロセスのそれぞれを別々に処理しようとしています。ある Silk Test クライアントでテストされたアプリケーションは、最初のクライアントが実行している間、他のクライアントでテストすることはできません。たとえば、Silk4J と Silk4NET で交互に同じアプリケーションをテストすることはできません。

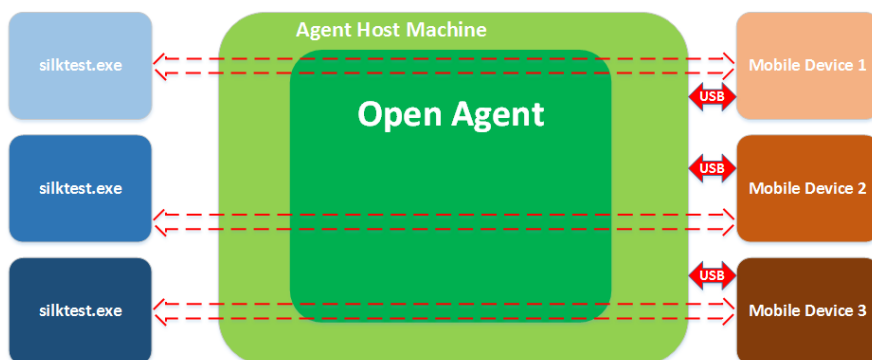
 **注:** 同じモバイル デバイス上で同時に複数のテストを実行することはできません。テストを並列実行する前に、十分なデバイスとエミュレータが利用できることを確認してください。モバイル デバイスやエミュレータが割り当てられないと、テストの実行は失敗します。

並列テストの実行それぞれが、1 個のブラウザーまたはモバイル デバイスに対して個々の `silktest.exe` を開始します。ブラウザーやモバイル デバイスを特定の `silktest.exe` と関連付けたい場合には、接続文字列を使用できます。詳細については、「[モバイル デバイスの接続文字列](#)」または「[リモートデスクトップ ブラウザーの接続文字列](#)」を参照してください。

次の図は、複数のブラウザーの並列テストのイメージを表します。



次の図は、複数のデバイスの並列テストのイメージを表します。



複数のプロセスが同時に開始されると、Silk Test Workbench を実行しているマシン上で Open Agent を何度も開始しようとする可能性があります。同じマシンで複数回 Open Agent を実行することはできないため、Silk Test Workbench は例外をスローすることになります。これを避けるため、テストを並列実行する前に、Open Agent が実行されていることを確認してください。

テスト結果は、複数の 結果ファイル（各テスト実行毎に 1 つ）に保存されます。

ビジュアル テスト再生の構成

ビジュアル テストで検証が処理される方法のようなビジュアル テストの再生動作、ビジュアル テストのテスト アプリケーションとのタイミングと同期、結果の使用法と結果に表示される内容など、さまざまな側面を構成できます。

ビジュアル テストの再生動作を構成する方法は主に 2 つあります。最初の方法は、さまざまな再生オプションの値を **オプション** ダイアログ ボックスで設定します。**オプション** ダイアログ ボックスで設定する再生オプションは、Silk Test Workbench によってその後のすべてのビジュアル テストの再生に自動的に適用されるグローバル オプションです。これらの構成オプションを設定するには、**ツール > オプション** を選択します。

2 番めの方法は、ビジュアル テストに再生設定の設定ステップを挿入し、グローバル再生オプション値を上書きします。再生設定の設定ステップに再生オプションを設定しても、**オプション** ダイアログ ボックスに表示されるデフォルトのグローバル再生オプションは永続的に変更されるわけではありません。再生設定の設定ステップでは、ステップが表示されているビジュアル テストの再生オプションのみが変更されます。

また、再生オプションの現在の値をビジュアル テストの変数に返して、この変数を式やテスト ロジックなどで使用することができます。

ビジュアルテスト内での再生設定値の設定

ビジュアルテスト内で再生設定を設定して、**オプション** ダイアログ ボックスのグローバル再生オプションのデフォルト値を上書きすることができます。ビジュアルテスト内で再生オプションを上書きしても、**オプション** ダイアログ ボックスに表示される再生オプションのデフォルト値が永続的に変更されるわけではありません。

1. 再生設定の値を設定するビジュアルテストを開きます。
2. 再生設定ステップを挿入する場所の直前のステップを選択します。
3. **挿入 > 再生設定 > 設定ステップ** を選択します。Silk Test Workbench によって、選択したステップのあとに再生設定の設定ステップが挿入されます。再生設定の設定ステップのプロパティが**プロパティ** ペインに表示されます。
4. 再生設定の設定ステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
5. **プロパティ** ペインで、目的の再生設定を選択し、その値を割り当てます。

以下の設定を利用できます。

- **ツール > オプション > 詳細設定** 以下のすべてのオプション（**ツール > オプション > 詳細設定 > アクティブデータ** 以下のオプションを除く）。
- **ツール > オプション > 再生 > 全般** 以下のすべてのオプション。
- **ツール > オプション > 再生 > タイミング** 以下のすべてのオプション。
- **ツール > オプション > 再生 > xBrowser** 以下のすべてのオプション。
- **ツール > オプション > 再生 > 閉じる** 以下のすべてのオプション。
- **ツール > オプション > 再生 > 再生ステータス ダイアログ** 以下のすべてのオプション。

Silk Test Workbench は、再生設定の設定ステップのラベルを次のように更新します：

再生設定 '[再生設定名]' に '[値]' を設定します

6. 他の再生設定の設定ステップを作成するには、必要に応じて上記の手順を繰り返します。ビジュアルテストを再生すると、再生設定の設定ステップで指定した再生設定値が Silk Test Workbench によって使用されます。

ビジュアルテスト内での再生設定値の取得

オプションの現在値を返して、ローカル変数に格納するステップをビジュアルテストに挿入することにより、再生設定を取得することができます。このローカル変数は、検証やロジックのステップで使用したり、再生設定値を元の値にリセットするために使用したりできます。

1. 再生設定の値を取得するビジュアルテストを開きます。
2. 再生設定値を保存するローカル変数を作成します。
3. 再生設定ステップを挿入する場所の直前のステップを選択します。
設定値を取得して変数に設定するステップは、その変数を使用するステップの直前に実行する必要があります。
4. **挿入 > 再生設定 > 取得ステップ** を選択します。Silk Test Workbench によって、選択したステップのあとに再生設定の取得ステップが挿入されます。再生設定の取得ステップのプロパティが**プロパティ** ペインに表示されます。
5. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
6. 適切な設定を選択して、その値を変数に割り当てます。

以下の設定を利用できます。

- **ツール > オプション > 詳細設定** 以下のすべてのオプション（**ツール > オプション > 詳細設定 > アクティブデータ** 以下のオプションを除く）。
- **ツール > オプション > 再生 > 全般** 以下のすべてのオプション。

- ツール > オプション > 再生 > タイミング 以下のすべてのオプション。
- ツール > オプション > 再生 > xBrowser 以下のすべてのオプション。
- ツール > オプション > 再生 > 閉じる 以下のすべてのオプション。
- ツール > オプション > 再生 > 再生ステータス ダイアログ 以下のすべてのオプション。

Silk Test Workbench は、再生設定の取得ステップのラベルを次のように更新します：

再生設定 '[再生設定名]' を取得して、ローカル変数 '[変数名]' に格納します

7. 他の再生設定の取得ステップを作成するには、必要に応じて上記の手順を繰り返します。

挿入されたビジュアル テストまたは VB .NET スクリプトの再生の構成

ビジュアル テストに VB .NET スクリプトまたはビジュアル テストを挿入したあとは、挿入した VB .NET スクリプトまたはビジュアル テストに適用する再生設定を構成することができます。

1. 挿入された VB .NET スクリプトまたはビジュアル テストが含まれているビジュアル テストを開きます。
2. VB .NET スクリプトまたはビジュアル テストを実行するステップを選択します。
ビジュアル テストのステップ テキストは、次のようになります：
ビジュアル テスト [資産名] の再生
ここで、[資産名] は、挿入されたビジュアル テストの名前です。
3. プロパティ ペインを開きます。
4. 再生設定 リストから、挿入された VB .NET スクリプトまたはビジュアル テストの再生時に適用する設定を次の中から選択します。
 - システムのデフォルトを使用
 - 親の設定を継承する
 - 親から設定をコピー

Silk Test Workbench がテストを同期する方法

ほとんどの予期しないテストの失敗は、同期問題に起因します。テストの再生時の同期が弱いと、予期しない結果が生成されるため、実際のアプリケーションの問題を見つけることが困難になります。同期エラーはタイミングの問題で、テストの環境に強く依存しており、このようなエラーを再現し解決することは困難です。たとえば、特定のテスト環境で発生する同期エラーが、開発環境では再現できないことがあります。同期が弱いと、自動化プロジェクトにおいて、自動化テスト セットの肥大化に伴い管理不能に陥る、最も一般的な理由のひとつになります。

Silk Test Workbench では、自動的なテスト同期をすべてのサポートするテクノロジーに対して提供しているため、ロバストで管理可能なテスト セットを構築できます。テストの再生中に、Silk Test Workbench は、AUT が次の操作を実行する準備ができるまで、常にテストが待機するようにします。テストの検証ステップでは、Silk Test Workbench は、テストのそれより前のテストが検証を実行する前に完了しているようにします。

AUT の特定の動作にテストを適用する場合は、次の同期タイムアウトの値を変更できます。

同期タイムアウト (OPT_SYNC_TIMEOUT)

再生中に AUT が準備完了状態になるまで Silk Test Workbench が待機する最大時間をミリ秒で指定します。デフォルト値は、300000 ミリ秒です。

オブジェクト解決タイムアウト (OBJ_WAIT_RESOLVE_OBJDEF)

Find メソッドがオブジェクトを検索する最大時間をミリ秒で指定します。デフォルト値は、5000 ミリ秒です。



注: 低速な接続でアクセスした仮想マシンなどの低速なシステムや高負荷な環境でテストを正しく実行できるようにするために、Silk Test Workbench は内部タイムアウトを増やすことがあります (AUT の開始時やダイアログやウィンドウの表示時など)。AUT、ダイアログ、ウィンドウが完全に表示されると、Silk Test Workbench は再びタイムアウト値を OPT_WAIT_RESOLVE_OBJDEF で指定された値に戻します。

オブジェクト解決再試行間隔 (OPT_WAIT_RESOLVE_OBJDEF_RETRY)

Silk Test Workbench がオブジェクトを直ちに見つけることができなかった場合、Silk Test Workbench はオブジェクト解決タイムアウトが経過するまでオブジェクトの検索を再試行します。オブジェクト解決再試行間隔は、Silk Test Workbench がオブジェクトの検索を再試行するまで待機する時間をミリ秒で指定します。デフォルト値は、1000 ミリ秒です。

オブジェクト有効化タイムアウト (OPT_OBJECT_ENABLED_TIMEOUT)

再生中にオブジェクトが有効になるまで Silk Test Workbench が待機する最大時間をミリ秒で指定します。デフォルト値は、1000 ミリ秒です。



注: タイムアウトは重なりません。

Silk Test Workbench が提供する自動的な同期のうち、特に Web アプリケーションに関する詳細な情報については、「xBrowser のページ同期」を参照してください。Silk Test Workbench が提供する同期のうち、特に AJAX アプリケーションに関する詳細な情報については、「[How to Synchronize Test Automation Scripts for Ajax Applications](#)」を参照してください。

自動的な同期に加えて、Silk Test Workbench では、VB.NET スクリプトに待機関数を手動で追加することもできます。Silk Test Workbench は、手動同期用に次の待機関数を提供します。

WaitForObject

指定したロケータに一致するオブジェクトを待機します。XPath ロケータやオブジェクト マップ識別子に対して動作します。

WaitForProperty

propertyName パラメータで指定したプロパティが expectedValue パラメータで指定した値を返すまで、またはタイムアウトに到達するまで待機します。

WaitForDisappearance

オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。

WaitForChildDisappearance

locator パラメータで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。

テストが ObjectNotFoundException で不規則に失敗する場合、オブジェクト解決タイムアウトを増やします (30 秒にするなど)。進行状況ダイアログが表示され、長時間計算が行われた後に表示されるオブジェクトをクリックする場合など、特定の操作に特別に時間のかかる場合は、WaitForObject メソッドをテスト スクリプトに手動で追加してオブジェクトが見つかるまで待機するようしたり、WaitForDisappearance メソッドをテスト スクリプトに追加して進行状況ダイアログが表示されなくなるまで待機するようにします。

自動的な同期の例

Silk Test Workbench がキャプション **Ok** を持つボタンをクリックする、次のサンプルコードを考えます。

```
'VB .NET code  
Dim button = _desktop.PushButton("@caption='Ok'")  
button.Click()
```

このサンプルコードの操作を再生するために、Silk Test Workbench は次の同期操作を実行します。

1. Silk Test Workbench はボタンを検索します。そして、オブジェクト解決タイムアウトが経過した場合、Silk Test Workbench は再生を停止して例外をスローします。
2. Silk Test Workbench はテスト対象アプリケーション (AUT) が準備完了状態になるまで待機します。AUT が準備完了状態になる前に、同期タイムアウトが経過した場合、Silk Test Workbench は再生を停止して例外をスローします。
3. Silk Test Workbench はボタンが有効になるまで待機します。オブジェクト有効化タイムアウトがボタンが有効になる前に経過した場合、Silk Test Workbench は再生を停止して例外をスローします。
4. Silk Test Workbench はボタンをクリックします。
5. Silk Test Workbench はテスト対象アプリケーション (AUT) が再び準備完了状態になるまで待機します。

結果の分析

結果はビジュアルテストやスクリプトの再生中に生成される資産で、これによって、テストプロセスを証明したり、任意の時点におけるテストアプリケーションの機能の状態を永続的に記録したりできます。結果には、ビジュアルテストやスクリプトの名前、実行番号、各ステップや行が実行された日時、コマンドやステップのステータス（合格か失敗か）、その他の重要な情報など、再生に関する情報が含まれます。結果は、再生情報を確認し、テストに失敗した領域を分析するのに役立ちます。

結果の表示は、再生後すぐに **結果** ウィンドウを開くか、または **アセットブラウザ** から結果を選択して行います。**結果** ウィンドウでは、**ビジュアルナビゲータ** の4つのペインを使用して結果情報が表示されます。また、**結果** ウィンドウでは結果情報がタブを使用して整理されるため、再生の要約レポート、合格/失敗の検証、フラグを付けられたステップ、およびビジュアルテストのすべてのステップやスクリプトのコード行の詳細な結果にすばやくアクセスできます。

結果の **画面プレビュー** で、ビジュアルテストの記録中にキャプチャされていた画面と再生中にキャプチャされた画面とを比較できます。比較して違いがあった場合は、**画面プレビュー** からすぐに再生画面のビジュアルテスト画面を更新できます。詳細については、「画面テストステップのキャプチャ画面の更新」を参照してください。

テスト手順の文書化や、他のテスト担当者との協業においては、結果を印刷したり、結果をファイルに保存したり、結果内の重要な問題について自分自身または他の担当者に注意を促すために、項目にフラグを付けたりすることができます。


また、フィルタを作成して適用すれば、最も関心のある結果情報のみを表示できます。たとえば、直前のテストの再生で失敗した検証のみを確認する場合、フィルタを作成して適用し、必要な情報のみを表示することができます。


結果を開いたあと、Silk Test Workbench では **結果** ウィンドウが表示されます。このウィンドウで、ビジュアルテストまたはスクリプトの再生に関するすべての側面についての詳細な情報を確認および分析できます。

[結果] ウィンドウ

ビジュアル テストまたはスクリプトを再生したあと、再生結果を **結果** ウィンドウに表示することができます。

Summary **Passed** Failed Flags Details (16 steps)

Overall result for run 1 :  **Passed** No verifications were executed

Browser:  Internet Explorer 11 - Windows 7

Reason for abort: Not applicable

Latest run number: 1

Recent runs: 1

Visual test executed: TestResultOS

Visual test description:

Result description:

Scripts (number of times each ran): [Project A.TestResultOS.version 1 \(1\)](#)

Verifications passed: 0 / 0

Verifications failed: 0 / 0

Flags: 0

Start time: 13.03.2018 14:14:39

End time: 13.03.2018 14:14:46

Total time: 7 s

Steps run: 16

Playback setting: System Defaults Edit...

結果 ウィンドウには以下の機能があります。

ビジュアル ナビゲータ テスト再生の全体の様子がすばやくわかります。詳細については、「[結果] ウィンドウのビジュアル ナビゲータ」を参照してください。

[結果] ウィンドウの ツールバー 結果の内容の表示と種類をカスタマイズすることができます。詳細については、「[結果] ウィンドウのツールバー」を参照してください。

[結果] ウィンドウの タブ タブには、結果の内容を特定の種類にまとめられています。詳細については、「[結果] ウィンドウのタブ」を参照してください。

[結果] ウィンドウのツールバー

結果 ウィンドウのツールバーには、結果に含まれる内容の表示と種類をカスタマイズするための以下のボタンとドロップダウン リストがあります。

- **種類の選択によるフィルタ処理の結果**：すべての事前定義済みおよびユーザー定義の結果フィルタにすばやくアクセスできます。リストからフィルタを選択します。
- **フィルタの管理**：結果フィルタを作成、編集、複製、および適用できる **フィルタの管理** ダイアログ ボックスが開きます。
- **条件**：条件 ダイアログ ボックスが表示されます。このダイアログ ボックスでは、今後のすべての実行の成功を定義する条件として、合格した検証のパーセンテージを設定できます。たとえば、合格条件を 90% に設定した場合は、ビジュアル テストまたはスクリプトにおける 10 件の検証のうち 9 件以上に

合格しないと、再生の結果が合格になりません。このオプションを設定すると、**結果合格条件（パーセンテージ）** オプションが更新されます。このパーセンテージは、今後のすべての結果に適用されます。

- **すべての実行を表示：実行の詳細** ダイアログ ボックスが開き、テストのすべてのテスト実行結果、およびテスト実行が合格したか失敗したかの一覧が表示されます。このダイアログ ボックスで、結果を開いたり削除したりすることができます。
- **更新**：現在の結果を更新します。
- **表示：テスト ステップ** ペインに表示されるステップのタイプを設定します。このボタンの横にあるドロップダウン矢印をクリックして、ステップのみを表示するか、画面のみを表示するか、またはその両方を表示するかを選択します。**ステップの説明** 列を表示するように選択することもできます。選択された表示は、**結果** ウィンドウの各タブ（**合格**、**失敗**、**フラグ**、および **詳細**）に適用されます。
- **基本表示：テスト ステップ** ペインの標準の列が表示され、**プロパティ** ペインに対応するプロパティが表示されます。
- **詳細表示：テスト ステップ** ペインに追加の列、および **プロパティ** ペインに対応するプロパティが表示されます。**テスト ステップ** ペインに追加の列を表示するには、右にスクロールすることが必要となる場合があります。

[結果] ウィンドウのタブ

結果 ウィンドウのタブは、ビジュアル テストの特定の種類のステップまたはスクリプトの特定の種類のコード行をすばやく表示できるフィルタとして機能します。

合格、**失敗**、**フラグ**、および **詳細** タブのすべてで同じ方法、つまり **ビジュアル ナビゲータ** のペインを使用して内容が整理され、表示されます。それぞれのタブでは、表示される内容の種類が異なります。**詳細** タブには、各ステップまたはコード行が含まれます。**合格** タブには、合格した検証のみが含まれます。**失敗** タブには、失敗した検証のみが含まれます。**フラグ** タブには、フラグを付けられた結果ステップのみが含まれます。

結果 ウィンドウには以下のタブがあります。

要約 タブ 結果を開くと、このタブがデフォルトで表示されます。デフォルトの結果表示を変更するには、[全般] オプションの **デフォルトの結果表示** オプションを変更します。**要約** タブには、以下の情報を含む一般的な概要レポートが表示されます。

- **総実行結果**：ビジュアル テストまたはスクリプトが正常に再生されて、結果合格条件（パーセンテージ）を満たしている場合は、[合格] と表示されます。結果合格条件（パーセンテージ）を満たしていない場合は、[失敗] と表示されます。ステップを正しく実行できなかった場合は、[再生エラー] と表示されます。
- **ブラウザー**：Web アプリケーションに対してテストが実行された場合、次の情報が行に表示されます。
 - テストが実行されたブラウザー。
 - ブラウザーのバージョン。
 - 再生時に使用するブラウザー サイズ。この情報は、デフォルト サイズではなく、ブラウザー サイズを変更した場合にのみ表示されます。
 - オペレーティング システムとバージョン。
 - モバイル デバイス、エミュレータ、シミュレータの名前。モバイル Web アプリケーションに対してテストが実行された場合にのみ、この情報が表示されます。
 - ブラウザー テストが実行されたモバイル デバイスが接続されている、またはエミュレータやシミュレータが実行されているマシンの名前。モバイル Web アプリケーションに対してテストが実行された場合にのみ、この情報が表示されます。
- **デバイス**：ネイティブ モバイル アプリケーションに対してテストが実行された場合、次の情報が行に表示されます。
 - モバイル デバイス、エミュレータ、シミュレータの名前。
 - モバイル デバイス、エミュレータ、シミュレータのオペレーティング システムとそのバージョン。

- モバイル デバイスが接続されている、またはエミュレータやシミュレータが実行されているマシンの名前。
- **モバイル アプリケーション**：テストを実行したモバイル アプリケーションへの完全パス。ネイティブ モバイル アプリケーションに対してテストが実行された場合にのみ、この情報が表示されます。
- **中止の理由**：ビジュアル テストまたはスクリプトの再生が中止された理由を表示します。
- **最新の実行番号**：最新の結果の実行番号を表示します。
- **最近の実行**：最近の実行を表示します。前の実行を表示するには、表示する実行をクリックします。このフィールドに表示されていない以前の実行を開くには、ツールバーの **すべての実行を表示** をクリックして **実行の詳細** ダイアログ ボックスを開きます。ここで、前の実行を開いたり削除したりできます。
- **実行されたビジュアル テスト/.NET スクリプト**：結果のビジュアル テストまたはスクリプトの名前を表示します。
- **ビジュアル テスト/.NET スクリプトの説明**：結果のビジュアル テストまたはスクリプトの説明を表示します。
- **結果の説明**：結果の説明を表示します。
- **ビジュアル テスト** または **スクリプト**：再生の一部として正しく実行されたビジュアル テストまたはスクリプトをすべて一覧します。これには、Workbench.RunScript() メソッドを使用して実行された、挿入されたビジュアル テストまたはスクリプトも含まれます。たとえば、ドライバ スクリプトでは、一度の再生により複数のスクリプトが実行されることがあります。さらに、ビジュアル テストまたはスクリプトのバージョンと、ビジュアル テストまたはスクリプトを実行した回数が表示されます。
- **合格した検証**：すべてのビジュアル テストまたはスクリプトで、正常に実行され合格した検証の合計数を表示します。このフィールドの数字をクリックすると **合格** タブが開くため、合格したすべての検証を確認できます。
- **失敗した検証**：すべてのビジュアル テストまたはスクリプトで、正常に実行されたが失敗した検証の合計数を表示します。このフィールドの数字をクリックすると **失敗** タブが開くため、失敗したすべての検証を確認できます。
- **フラグ**：ビジュアル テストについて、結果でフラグを付けられた検証ステップの総数です。このフィールドの数値をクリックして、**フラグ** タブを開きます。フラグは、スクリプトでは使用できません。
- **開始時刻**：最初のビジュアル テストまたはスクリプトの再生が開始された時刻です。
- **終了時刻**：最後のビジュアル テストまたはスクリプトの再生が終了した時刻です。
- **合計時間**：ビジュアル テストまたはスクリプトを再生した合計時間です。
- **実行したステップ数**：実行したステップまたはコード行の総数です。
- **再生設定**：結果の作成に使用される再生オプションのグループである再生設定を表示します。**編集** をクリックして再生オプションを表示し、再生設定を設定することができます。

合格 タブ： 再生中に合格した検証が表示されます。

失敗 タブ： 再生中に失敗した検証が表示されます。再生エラーとなったステップはこのタブには表示されません。

フラグ タブ： 結果内のフラグ付きステップが表示されます。結果が作成されたあとで手動でフラグを挿入するか、またはビジュアル テストの再生中にフラグを挿入する検証ロジックを使用することで、結果のステップにフラグを挿入できます。**フラグ** タブには、結果内のすべてのフラグ付きステップが動的に表示されます。たとえば、**合格**、**失敗**、または **詳細** タブのいずれかにフラグを挿入すると、Silk Test Workbench によって **フラグ** タブが自動的に更新され、フラグを付けられたステップが表示されます。



注： ビジュアル テストのフラグ付きのステップは、再生後はこのタブに表示されません。結果でフラグを付けられたステップと検証ロジックで挿入されたフラグのみが、結果の **フラグ** タブに表示されます。フラグは、スクリプトでは使用できません。

詳細タブ: ビジュアル テストの各ステップまたはスクリプトの各コード行に関する情報が表示されます。テスト ステップやコード行の名前、合格/失敗のステータス、説明、フラグのステータスなどです。キーワード駆動テストの場合、**詳細** タブにはキーワード駆動テストに含まれ、ビジュアル テストとして実装されているキーワードのステップが表示されます。**詳細** タブには、再生の詳細も表示されます。再生中にエラーが発生したとき、そのビジュアル テストまたはスクリプトが他のビジュアル テストから呼び出されていた場合、再生の詳細には、ビジュアル テストまたはスクリプトを呼び出したビジュアル テストのテスト ステップも含まれます。

[結果] ウィンドウのそれぞれのタブで、**テスト ステップ** ペインの **基本表示** または **詳細表示** のいずれかを選択できます。**詳細表示** では、**テスト ステップ** ペインに追加の列が表示されます。


実行ステータスの種類

Silk Test Workbench は、再生中のテストのパフォーマンスに基づいて、結果の各テスト実行に対して実行ステータスを割り当てます。実行ステータスは、結果の **要約** タブの **総実行結果** テキスト ボックスに表示されます。

実行ステータスは結果の合格条件を示す割合に依存します。結果の合格条件は、テストの合格を定義する、合格した検証の割合を指定したものです。たとえば、合格条件を 90% に設定した場合、テストの 10 件の検証のうち 9 件に合格しないと、再生の結果が合格になりません。

実行ステータスの種類

実行ステータスの種類には、以下のものがあります。

タイプ	説明
合格	合格 実行ステータスは、成功したテスト再生を示します。テストに検証が含まれている場合、 合格 ステータスは結果の合格条件の割合に達していることも示します。
失敗	失敗 実行ステータスは、再生中に失敗した検証がテストに含まれており、失敗した検証の割合が結果の合格条件の割合に達していないことを示します。  注: デフォルトでは、結果の合格条件は 100% に設定されています。このオプションを変更するには、 結果 ウィンドウのツールバーの 条件 をクリックします。
再生エラー	再生エラー 実行ステータスは、正常に実行されなかったステップが結果に含まれていることを示します。たとえば以下のような場合です。 <ul style="list-style-type: none"> アクティブ データ ファイル内の存在しない列を参照するステップ 存在しないローカル変数を参照するステップ 開始されていないタイマーを実行するステップ

結果のステップのステータスの詳細については、「**ステップのステータスの種類**」を参照してください。


ステップのステータスの種類

Silk Test Workbench は、再生中のステップのパフォーマンスに基づいて、結果の各ステップに対してステータスを割り当てます。ステップのステータスは **テスト ステップ** ペインの **結果** 列に表示され、ステップのテキストはそのステータスの種類に応じた色で表示されます。

ステップのステータスの種類

結果のステップのステータスの種類には、以下のものがあります。

タイプ	説明
合格	検証ステップがそのステップの検証ロジックで定義されている条件を満たしている場合は、ステップに 合格 ステータスが割り当てられます。

タイプ	説明
失敗	<p>このステータスの種類が割り当てられているステップは、結果の 合格 タブおよび 詳細 タブに表示されます。 結果 列に 合格 と表示され、ステップ テキストが緑で表示されます。</p> <p>検証ステップがそのステップの検証ロジックで定義されている条件を満たしていない場合は、ステップに 失敗 ステータスが割り当てられます。</p> <p>このステータスの種類が割り当てられているステップは、結果の 失敗 タブおよび 詳細 タブに表示されます。 結果 列に 失敗 と表示され、ステップ テキストが赤で表示されます。</p>
再生エラー	<p>再生エラー ステータスのステップは、正常に実行されなかったステップです。たとえば以下のような場合です。</p> <ul style="list-style-type: none"> • アクティブ データ ファイル内の存在しない列を参照するステップ • 存在しないローカル変数を参照するステップ • 開始されていないタイマーを実行するステップ <p>結果 列に 再生エラー と表示され、ステップ テキストが青で表示されます。</p>
エラー画面	<p>再生エラー ステータスのステップのあとに表示されるステップです。 エラー画面 ステータスが割り当てられたステップでは、テストで再生エラーが発生したときに表示されたテスト アプリケーションの画面が表示されます。</p> <p>結果 列に エラー画面 と表示され、ステップ テキストが青で表示されます。</p> <p> 注: このステップ タイプはテスト スクリプトでは利用できません。</p>
正しく実行されたステップ	<p>正しく実行されたすべてのステップ（検証ステップ以外）。</p> <p>結果 列は空で、ステップ テキストの色は黒です。</p>

結果の実行ステータスの詳細については、「実行ステータスの種類」を参照してください。

[結果] ウィンドウのビジュアル ナビゲータ

結果 ウィンドウには [ビジュアル ナビゲータ] があり、テストの再生をあらゆる面からすばやく確認できます。

ビジュアル テストの場合は、**合格** タブ、**失敗** タブ、**フラグ** タブ、**詳細** タブのすべてに [ビジュアル ナビゲータ] の以下の 4 つのペインが表示されます。

- テスト ステップ
- 画面プレビュー
- プロパティ
- ストーリーボード

結果 ウィンドウの **合格**、**失敗**、**フラグ**、および **詳細** タブでは、ビジュアル ナビゲータの 4 つのペインで内容が整理されて表示されます。**テスト ステップ**、**画面プレビュー**、**ストーリーボード**、および **プロパティ** ペインは互いに同期し、**テスト ステップ** ペインで選択したステップに固有の情報が表示されます。**結果** ウィンドウのビジュアル ナビゲータ ペインには、以下のセクションで説明する追加機能が含まれていません。

テスト ステップ

結果を表示すると、**テスト ステップ** ペインには、ビジュアル テストと同様の情報と各テスト ステップの結果の追加データが表示されます。ステップを選択すると、選択したステップに固有の情報でその他のペインが更新されます。また、フラグを挿入したり、選択したステップにステップの説明を追加したりできます。結果を [基本表示] または [詳細表示] で表示することを選択できます。

画面プレビュー

結果の [画面プレビュー] で、ビジュアル テストの記録中にキャプチャされていた画面と再生中にキャプチャされた画面とを比較できます。**テスト ステップ** ペインで比較する画面を含むステップを選択してから、**画面プレビュー** で **操作 > 相違点の表示 > 横に並べる** の順にクリックします。再生中にキャプチャされた画面を含むようにビジュアル テストを更新する場合は、**操作 > 画面の更新** の順にクリックします。

ストーリーボード

結果の **ストーリーボード** ペインでは、アイコンで検証ロジック ステップの実行ステータスが示されます。

プロパティ

結果の **プロパティ** ペインには、再生中にキャプチャされた結果のプロパティと値が表示されます。

プロパティ ペインには **再生前のビジュアル テストのステップ プロパティを表示/非表示** ツールバー ボタンがあり、選択したステップのビジュアル テストのプロパティを表示または非表示にすることができます。

スクリプトでは、**結果** ウィンドウには **テスト ステップ** ペインおよび **プロパティ** ペインのみ表示されません。**テスト ステップ** ペインには、各テスト ステップの再生ステータスと結果の詳細情報を示す列があります。

結果データの表示をカスタマイズするには、**全般** オプションから **デフォルトの結果表示** を変更します。再生結果オプションを変更し、目的のデータを表示するフィルタを作成することもできます。

結果 ウィンドウのツールバーの [基本表示] または [詳細表示] オプションを使用することにより、**プロパティ** ペインに特定のプロパティを表示できます。

結果の管理

デフォルトでは、Silk Test Workbench によりビジュアル テストまたはスクリプトの再生後に結果が作成され、固有の資産として保存されます。結果を表示するには、以下のいずれかの方法を使用できます。

- 再生後、**再生完了** ダイアログ ボックスから、手動で結果を開きます。
- 再生直後、自動的に結果を開く
- あとでアセット ブラウザから結果を開く

結果を開くと、**結果** ウィンドウが表示され、デフォルトで結果の **要約** タブが表示されます。デフォルトの結果表示を変更するには、**全般** オプションの **デフォルトの結果表示** オプションを変更します。

デフォルトの結果名は、再生したビジュアル テストまたはテスト スクリプトの名前です。**再生結果** オプションで **デフォルトの結果名** オプションを変更できます。

結果の作成

1. ビジュアル テストまたはスクリプトを開きます。
2. **再生** をクリックします。**再生** ダイアログ ボックスが開きます。
3. **結果の説明** テキスト ボックスに結果の説明を入力します。
4. **結果オプション** リストからオプションを選択することにより、結果を保存する方法を選択します。

以下のオプションのいずれかを選択します。

- **新しい実行番号で結果を保存** : 再生の結果は、増分された実行番号で保存されます。これはデフォルトの設定です。
- **現在の実行番号に結果を追加** : 現在の再生の結果は、最新の既存の実行番号の結果に追加されます。新しい実行番号は作成されません。この設定は、結果が存在している場合のみ表示されます。
- **既存の実行番号を新しい結果で上書き** : 前の再生は、新しい再生結果によって前の再生の実行番号を使って上書きされます。この設定は、ビジュアル テストまたはスクリプトがはじめて再生されたときのみ表示されます。

- **この結果を保存しない**：結果ファイルを作成せずに再生します。既存の結果がないビジュアルテストまたはスクリプトに対して **現在の実行番号に結果を追加** または **既存の実行番号を新しい結果で上書き** が選択された場合は、**新しい実行番号で結果を保存** が使用されます。

5. **OK** をクリックします。

結果の動作と表示のカスタマイズ

Silk Test Workbench には、結果の動作と表示を設定するためのさまざまな方法があります。

オプション ダイアログ ボックスでは、結果オプションを含むさまざまな再生オプションの値を設定できます。**オプション** ダイアログ ボックスで設定する再生オプションは、Silk Test Workbench によってその後のすべてのビジュアルテストの再生に自動的に適用されるグローバル オプションです。詳細については、「再生オプションの変更」を参照してください。

ビジュアルテストの場合は、再生設定の設定ステップを挿入し、目的の再生設定を選択することにより、デフォルトの結果オプションを上書きできます。詳細については、「ビジュアルテスト内での再生設定値の設定」を参照してください。

スクリプトとビジュアルテストの場合は、結果の記録に関するいくつかの特性を **再生** ダイアログ ボックスから直接制御できます。詳細については、「結果の記録のオン/オフの切り替え」を参照してください。

結果の記録のオン/オフの切り替え

デフォルトでは、Silk Test Workbench は、ビジュアルテストまたはスクリプトの再生ごとに結果を作成します。再生後に Silk Test Workbench が結果を作成しないようにするには、**再生結果** オプションまたは **再生** ダイアログ ボックス（ビジュアルテストまたはスクリプトを再生するとデフォルトで表示されます）で **結果オプション** を設定します。

1. ツール > **オプション** を選択します。 **オプション** ダイアログ ボックスが開き、デフォルトで **全般** オプションがリストされます。
2. ツリー ビューで **再生** オプションをダブルクリックし、**結果** をクリックします。使用可能な結果オプションがダイアログ ボックスの右側のウィンドウに表示されます。
3. 結果の記録をオフにするには、**結果オプション** リストから **この結果を保存しない** を選択します。
4. 結果の記録をオンにするには、**結果オプション** リストから以下のいずれかのオプションを選択します。
 - **新しい実行番号で結果を保存**：再生の結果は、増分された実行番号で保存されます。これはデフォルトの設定です。
 - **現在の実行番号に結果を追加**：現在の再生の結果は、最新の既存の実行番号の結果に追加されます。新しい実行番号は作成されません。この設定は、結果が存在している場合にのみ表示されます。
 - **既存の実行番号を新しい結果で上書き**：前の再生は、新しい再生結果によって前の再生の実行番号を使って上書きされます。この設定は、ビジュアルテストまたはスクリプトがはじめて再生されたときにのみ表示されます。

5. **OK** をクリックします。

再生後に結果を手動で開く

再生後に手動で結果を開いて、結果を表示することができます。

1. ツール > **オプション** を選択します。 **オプション** ダイアログ ボックスが開き、デフォルトで **全般** オプションがリストされます。
2. **[再生完了] ダイアログの表示** オプションが **はい** になっていることを確認するか、または設定します。
3. ビジュアルテストまたはスクリプトを実行します。 **再生完了** ダイアログ ボックスが開きます。
4. **結果へ移動** をクリックします。 **結果** ウィンドウが開き、結果が表示されます。

再生後に結果を自動的に開く

自動的に結果を開くように Silk Test Workbench を設定して、結果を表示することができます。

1. ツール > オプション を選択します。オプション ダイアログ ボックスが開き、デフォルトで 全般 オプションがリストされます。
2. 結果を自動的に開く リスト ボックスから、はい を選択します。
3. [再生完了] ダイアログの表示 リスト ボックスから、いいえ を選択します。
これにより、再生後に 再生完了 ダイアログ ボックスが表示されなくなります。
4. ビジュアル テストまたはスクリプトを実行します。再生後、Silk Test Workbench によって自動的に 結果 ウィンドウが表示され、結果が表示されます。

アセット ブラウザからの結果の表示

あとでアセット ブラウザから結果を開き、結果を表示することができます。

1. 表示 > アセット ブラウザ を選択します。アセット ブラウザ ウィンドウが開きます。
2. 資産の種類 リストから、結果 を選択します。項目ペインに既存の結果がリストされます。
3. 表示する結果をダブルクリックします。結果 ウィンドウが開き、結果が表示されます。

複数のビジュアル テストのエラーのトレース

ビジュアル テストまたは VB .NET スクリプトが他のビジュアル テストから呼び出され、実行中にエラーが発生した場合、呼び出したビジュアル テストの結果の 詳細 タブにエラーを起こしたテストステップが表示され、エラーが発生した呼び出されたビジュアル テストまたは VB .NET スクリプトそれぞれのテストステップまたはコードの行数と共に表示されます。呼び出したビジュアル テストが、さらに他のビジュアル テストのステップから呼び出されていた場合、呼び出したビジュアル テストの結果には呼び出されビジュアル テストのステップがさらに表示されます。これらのテスト ステップを参照するには、つぎの作業を行います。

1. 実行されたビジュアル テストの結果を開きます。
2. 詳細 タブを選択します。
3. エラーが発生したステップを選択します。
4. プロパティ ペインを開きます。再生の詳細 セクションの 呼び出し元 リストに、呼び出したビジュアル テスト、および呼び出されたビジュアル テストまたはテスト スクリプトにおける、エラーが発生したステップが対応するビジュアル テストまたはテスト スクリプトの名前と共にリスト表示されます。
5. リストから資産を選択し、資産を開く をクリックすると、エラーが発生したステップまたは行に直接アクセスできます。

例

ビジュアル テスト *Test1* の 2 番目のステップがビジュアル テスト *Test2* を呼び出し、**Test2** の 5 番目のステップがビジュアル テスト *Test3* を呼び出している場合に、*Test1* の実行時に *Test3* の 4 番目のステップでエラーが発生すると、*Test1* の結果ファイルには次の項目が含まれます。

呼び出し元	
Test1	2
Test2	5
Test3	4

結果からビジュアル テストまたはテスト スクリプトへの切り替え

特定の結果にリンクされているビジュアル テストを表示します。

1. ビジュアル テストの結果を開きます。

2. **詳細** タブをクリックし、**テスト ステップ** ペインのタイトルバーで **操作** をクリックします。
3. **ビジュアル テストのウィンドウ** を選択します。
4. **結果** ウィンドウに戻るには、**ビジュアル テストの テスト ステップ** ペインのタイトルバーで **操作** をクリックして、**[結果] ウィンドウへ移動** を選択します。

ビジュアル テストへの結果コメントの挿入

結果コメントを使用すると、結果内のビジュアル テストに関する補足情報を提供できます。ビジュアル テストを再生すると、**結果** ウィンドウの **詳細** タブにコメントがステップとして表示されます。



ヒント: 結果コメントの挿入以外に、フラグを使用して選択したステップのコメントを挿入することもできます。

1. ビジュアル テストの場合、**ビジュアル テストの テスト ステップ** ペインで、結果コメントを表示するステップを選択します。
2. **挿入 > 結果コメント** を選択します。**テスト ステップ** ペインに結果コメントステップが表示され、このステップのプロパティが **プロパティ** ペインに表示されます。
3. **プロパティ** ペインで、以下のステップのいずれかを実行します。
 - **式** プロパティ値テキスト ボックスにコメントを入力します。
 - テキスト ボックスをクリックして、**選択** ボタンを有効にすると、変数を選択したり、コメント値として使用する式を作成できます。
4. ビジュアル テストを再生します。結果の **詳細** タブで、結果コメントステップの **結果の詳細** 列にコメントが表示されます。

ビジュアル テストの結果の印刷

1. 印刷する結果を開きます。
2. **ファイル > 印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
3. リストからプリンタを選択します。
4. 省略可能：**プロパティ** をクリックすると、余白や印刷の向きなどの印刷ページ設定を変更できます。
5. 省略可能：**印刷範囲** セクションで、ドキュメント全体を印刷するか、または選択したページのみを印刷するかを指定します。
6. 省略可能：**部数** フィールドで、印刷する部数を指定します。
7. 省略可能：**部単位で印刷する** チェック ボックスをオンにすると、すべてのコピーの対応するページをまとめて印刷します。
このチェック ボックスは、複数の部数を印刷するように選択した場合にのみ有効になります。
8. 印刷出力をカスタマイズするには、**オプション** をクリックして、印刷出力に含める結果情報の種類を選択します。

以下のオプションがあります。

以下を 含める要約の種類を指定します。

含める

- **ビジュアル テストの要約**：要約にはビジュアル テストの名前、プロジェクト、作成者、作成日、バージョン、および説明が含まれています。この要約は、印刷出力の最後のページに表示されます。
- **結果の要約**：最新の結果の要約ページを含めます。この要約は、印刷出力の最初のページに表示されます。
- **グループ画面**：現在のダイアログ ボックスまたは Web ページのメイン コンテンツを含むグループ画面のスナップショットを印刷出力に含めます。グループ画面に続くテストステップは、現在のダイアログ ボックスまたは Web ページのコンテンツで発生する単一の操作に対応します。グループ画面は、**テスト ステップ** ペインの Using ステップに対応

します。グループ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。

タブ 印刷対象の内容を含むタブを指定します。印刷出力では、**詳細** タブの内容が先に示され、そのあとに **合格** タブ、**失敗** タブ、および **フラグ** タブの内容が示されます。

- **合格** : 結果の **合格** タブの内容を含めます。
- **失敗** : 結果の **失敗** タブの内容を含めます。
- **フラグ** : 結果の **フラグ** タブの内容を含めます。
- **詳細** : 結果の **詳細** タブの内容を含めます。

各ステップに含める 各ステップに含める内容を指定します。

- **画面** : 各ステップの関連画面スナップショットを含めます。個々の画面は (ステップ画面) に対応し、現在のダイアログ ボックスまたは Web ページでの特定の操作に関連付けられます。ステップ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。
- **説明** : 各ステップの説明を含めます。ステップの説明は、**プロパティ** ペインの **ステップの説明** プロパティに表示されるように、印刷出力に示されます。
- **フラグの詳細** : フラグの説明、作成者、更新者、更新日、割り当てられたユーザーなどのフラグの詳細を含めます。

以下のプロパティを含める 各ステップに含めるプロパティを指定します。

- **拡張プロパティ** : ステップに関する補足情報を提供する拡張プロパティを含めます。
- **その他すべてのプロパティ** : 各ステップの拡張プロパティ以外のすべてのプロパティを含めます。

デフォルト デフォルトの印刷オプションを選択するには、このボタンをクリックします。

9. 省略可能 : **プレビュー** をクリックして、結果情報がどのように印刷されるかを表示します。**プレビュー** ウィンドウが開きます。

100K をクリックします。

スクリプトの結果の印刷

1. 印刷する結果を開きます。
2. **ファイル** > **印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
3. 印刷オプションを設定します。
4. 印刷出力をカスタマイズするには、**オプション** をクリックして、印刷出力に含める結果情報の種類を選択します。

以下のオプションがあります。

以下を含める 含める内容の種類を指定します。

- **.NET スクリプトの要約** : スクリプトの名前、プロジェクト、作成者、作成日、バージョン、および説明を含むスクリプトの要約を含めます。この要約は、印刷出力の最後のページに表示されます。
- **結果の要約** : スクリプトの最新の結果の要約ページを含めます。この要約は、印刷出力の最初のページに表示されます。
- **グループ画面** : スクリプトでは利用できません。

タブ 印刷対象の内容を含むタブを指定します。印刷出力では、**詳細** タブの内容が先に示され、そのあとに **合格** タブ、**失敗** タブ、および **フラグ** タブの内容が示されます。

- **合格** : 結果の **合格** タブの内容を含めます。

- **失敗** : 結果の **失敗** タブの内容を含めます。
- **フラグ** : 結果の **フラグ** タブの内容を含めます。
- **詳細** : 結果の **詳細** タブの内容を含めます。

各ステップに含める 各ステップに含める内容を指定します。

- **画面** : 各ステップに関連したスクリーンショットを含めます。個々の画面は、テスト対象アプリケーションのコントロールに対する固有の操作に関連付けられています。
- **説明** : スクリプトでは利用できません。
- **フラグの詳細** : フラグの説明、作成者、更新者、更新日、割り当てられたユーザーなどのフラグの詳細を含めます。

以下のプロパティを含める 各ステップに含めるプロパティを指定します。


- **拡張プロパティ** : スクリプトでは利用できません。
- **その他すべてのプロパティ** : スクリプトのすべてのプロパティを含めます。

デフォルト デフォルトの印刷オプションを選択するには、このボタンをクリックします。

5. 省略可能 : **プレビュー** をクリックして、結果情報がどのように印刷されるかを表示します。**プレビュー** ウィンドウが開きます。
6. 余白、向きなどのページ設定情報を変更するには、**プロパティ** をクリックします。
7. **OK** をクリックします。

ファイルへの結果の保存


テスト手順を文書化したり、他のテスト担当者と協業するために、ビジュアル テストまたはテスト スクリプトの結果を XML ファイル、タブ区切りファイル、またはプレーンなテキスト ファイルに保存することができます。ファイルに保存したら、それをレポート ツールにインポートして、表示、管理、フォーマット、印刷などを行えます。

 **注:** テキスト ファイルに保存したビジュアル テストまたはテスト スクリプトの結果をすばやく参照するには、ファイルをタブ区切りファイルとして Microsoft Excel にインポートします。結果が列見出しで整理されて表示されます。

1. ビジュアル テストまたはスクリプトの結果を開きます。
2. 以下の結果タブのいずれかをクリックします。
 - **要約**
 - **合格**
 - **失敗**
 - **フラグ**
 - **詳細**

選択したタブに表示されているデータのみが指定したファイルに保存されます。フィルタ処理されたデータは表示されず、結果の他のタブに存在しているデータは保存されません。

3. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開き、デフォルトの結果名が **ファイル名** テキスト ボックスに表示されます。

 **注:** デフォルトの結果名を変更するには、**再生結果** オプションの **デフォルトの結果名** オプションを変更します。

4. ファイルの保存場所を選択します。
5. ファイル名を入力するか、デフォルト名を使用します。
6. ファイルの種類を選択します。

任意のタブの結果を、XML 形式、タブ区切り形式、またはプレーンなテキスト形式で保存できます。

7. **保存** をクリックします。

Web ページ コンテンツのキャプチャ

ブラウザ ウィンドウに現在表示されている Web ページの一部をスクリーンショットにキャプチャする場合は、CaptureBitmap メソッドを使用できます。画像ファイルの場所と名前は、絶対パスまたは相対パスでパラメータで指定します。例：

```
'VB code  
browserWindow.CaptureBitmap("C:¥Temp¥MyPage.png")
```

Web ページ コンテンツ全体のスクリーンショットを 1 つの画像としてキャプチャする場合は、CaptureFullPageBitmap メソッドを使用できます。画像ファイルの場所と名前は、絶対パスまたは相対パスでパラメータで指定します。例：

```
'VB code  
browserWindow.CaptureFullPageBitmap("C:¥Temp¥MyPage.png")
```

結果フィルタ

多くの場合、結果には大量のデータが含まれます。最も重要なデータに焦点を合わせるために、合格/失敗のステータス、日付/時刻、フラグが付いたステップなどの基準に基づいて、あらかじめ定義したフィルタを適用するか、または専用のフィルタを定義できます。Silk Test Workbench は、現在表示されているビューに関係なく、**結果** ウィンドウの **合格**、**失敗**、**フラグ**、および **詳細** タブにフィルタを適用します。

結果フィルタの作成

結果フィルタを使用すると、**結果** ウィンドウに表示されるデータの量が減少するため、重要な結果をすばやく見つけることができます。フィルタを設定したら、**フィルタ** リストでそれを選択して適用します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. **新規作成** をクリックします。**新しいフィルタ** ダイアログ ボックスが開き、デフォルトで **全般** タブが選択されます。
4. **名前** テキスト ボックスに、フィルタの名前を入力します。
ここで指定した名前がツールバーの **フィルタ** リストに表示されます。
5. 省略可能：**説明** テキスト ボックスにこのフィルタの使用方法的説明を入力します。
6. フィルタの種類を選択します。
 - このフィルタを現在のユーザーにのみ使用可能にするには **個人用** を選択します。
 - すべてのユーザーに使用可能にするには **公開用** を選択します。
7. ダイアログ ボックスの **フィルタ条件** セクションで、**結果** リストから以下のいずれかのオプションを選択します。
 - すべて** ビジュアル テストのすべてのステップ、またはスクリプトのすべてのコード行を表示します。
 - 失敗** 正しく実行されなかったステップまたはコード行のみを表示します。
 - 合格** 正しく実行されたステップまたはコード行のみを表示します。
8. **日付/時刻** オプションを選択します。
 - 任意の日付の任意の時刻に発生した結果データを表示するには、**すべて** を選択します。
 - 特定の時刻または日付に発生した結果データを表示するには、**開始** を選択して目的の日付/時刻の範囲を指定します。
9. **ユーザー** オプションを選択します。
 - すべてのユーザーによって作成された結果データを表示するには、**すべて** を選択します。

- 選択したユーザーによって作成された結果データのみを表示するには、**選択済み** を選択してから目的のユーザーを選択します。
- 10** 省略可能：フラグが割り当てられていないすべての結果データをフィルタ処理で非表示にするには、**フラグが割り当てられているステップのみ表示する** チェックボックスをオンにします。
- 11** より綿密なフィルタを作成するには、**詳細** タブをクリックし、必要な検証のみをフィルタ対象として指定します。
デフォルトでは、すべての検証が Silk Test Workbench によってフィルタの対象に含まれます。
- 左側のペインには、コマンドの種類のアとに、かっこで囲まれた数が表示されます。これらの数は、使用可能なコマンドの総数と選択されたコマンドの数を表します。たとえば、**検証 (3/7)** は、使用可能な 7 種類の検証から 3 種類の検証が選択されていることを示します。また、太字のチェックは、特定のコマンドの種類に関してすべてのコマンドが選択されていることを示します。オペレーティング システムの設定に応じて、チェックボックス内の四角、または無効になっているチェックが、特定のコマンドの種類に関して一部のコマンドが選択されていることを示します。
- 12** **OK** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
- 13** アクティブな **結果** ウィンドウにフィルタを適用するには、フィルタを選択して **適用** をクリックし、**OK** をクリックします。
- 14** **閉じる** をクリックします。

結果フィルタの編集

フィルタを編集して、**結果** ウィンドウに表示されるデータを変更します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. ユーザー定義フィルタを選択します。
デフォルトのフィルタ (**検証** と **失敗した検証**) は編集できません。
4. **編集** をクリックします。**フィルタの編集** ダイアログ ボックスが開き、デフォルトで **全般** タブが選択されます。
5. **名前** テキスト ボックスに、フィルタの名前を入力します。
ここで指定した名前がツールバーの **フィルタ** リストに表示されます。
6. 省略可能： **説明** テキスト ボックスにこのフィルタの使用法の説明を入力します。
7. フィルタの種類を選択します。
 - このフィルタを現在のユーザーにのみ使用可能にするには **個人用** を選択します。
 - すべてのユーザーに使用可能にするには **公開用** を選択します。
8. ダイアログ ボックスの **フィルタ条件** セクションで、**結果** リストから以下のいずれかのオプションを選択します。
 - すべて** ビジュアル テストのすべてのステップ、またはスクリプトのすべてのコード行を表示します。
 - 失敗** 正しく実行されなかったステップまたはコード行のみを表示します。
 - 合格** 正しく実行されたステップまたはコード行のみを表示します。
9. **日付/時刻** オプションを選択します。
 - 任意の日付の任意の時刻に発生した結果データを表示するには、**すべて** を選択します。
 - 特定の時刻または日付に発生した結果データを表示するには、**開始** を選択して目的の日付/時刻の範囲を指定します。
- 10 **ユーザー** オプションを選択します。
 - すべてのユーザーによって作成された結果データを表示するには、**すべて** を選択します。
 - 選択したユーザーによって作成された結果データのみを表示するには、**選択済み** を選択してから目的のユーザーを選択します。

11 省略可能：フラグが割り当てられていないすべての結果データをフィルタ処理で非表示にするには、**フラグが割り当てられているステップのみ表示する** チェック ボックスをオンにします。

12 より綿密なフィルタを作成するには、**詳細** タブをクリックし、必要な検証のみをフィルタ対象として指定します。

デフォルトでは、すべての検証が Silk Test Workbench によってフィルタの対象に含まれます。

左側のペインには、コマンドの種類のとに、かっこで囲まれた数が表示されます。これらの数は、使用可能なコマンドの総数と選択されたコマンドの数を表します。たとえば、**検証 (3/7)** は、使用可能な 7 種類の検証から 3 種類の検証が選択されていることを示します。また、太字のチェックは、特定のコマンドの種類に関してすべてのコマンドが選択されていることを示します。オペレーティング システムの設定に応じて、チェック ボックス内の四角、または無効になっているチェックが、特定のコマンドの種類に関して一部のコマンドが選択されていることを示します。

13 **OK** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。

14 アクティブな **結果** ウィンドウにフィルタを適用するには、フィルタを選択して **適用** をクリックし、**OK** をクリックします。

15 **閉じる** をクリックします。

結果フィルタの適用

結果フィルタは、以下の 2 通りの方法で適用できます。結果フィルタは、アクティブな **結果** ウィンドウにのみ適用することも、開いているすべての **結果** ウィンドウに対して適用することもできます。

1. 結果を開きます。

2. 結果フィルタをアクティブな **結果** ウィンドウにのみ適用するには、**結果** ウィンドウのツールバーの **種類の選択によるフィルタ処理の結果** リストからフィルタを選択します。Silk Test Workbench によってフィルタがアクティブな **結果** ウィンドウに適用され、タイトル バーがフィルタの名前で更新されます。例：Silk Test Workbench - [Common - VisualTest1 (結果：フィルタ = 失敗した検証)] のように表示されます。

3. 結果フィルタを開いているすべての **結果** ウィンドウに適用するには、以下の手順を実行します。

a) 結果を開きます。

b) 適用するフィルタを選択します。

c) **適用** をクリックして、**OK** をクリックします。Silk Test Workbench によってフィルタがアクティブな **結果** ウィンドウに適用され、タイトル バーがフィルタの名前で更新されます。例：Silk Test Workbench - [Common - VisualTest1 (結果：フィルタ = 失敗した検証)] のように表示されます。

結果フィルタを作成すると、**結果** ウィンドウのツールバーの **フィルタの管理** ツールバー ボタンのすぐ左にあるドロップダウン リストに表示されます。このリストからフィルタを選択すると、Silk Test Workbench は開いているすべての **結果** ウィンドウにそれを適用し、タイトルバーが更新されてフィルタの名前が表示されます。

結果フィルタの複製

フィルタの管理 ダイアログ ボックスを使用して、結果フィルタのコピーを作成できます。

1. 結果を開きます。

2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。

3. **フィルタ** リストから、複製するフィルタを選択します。

4. **複製** をクリックします。**フィルタの複製** ダイアログ ボックスが開きます。

適用された結果フィルタの削除

結果フィルタを削除して、**結果** ウィンドウにすべての結果データを表示します。

1. 結果を開きます。

2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。

3. **(なし)** を選択します。
4. **適用** をクリックします。
5. **閉じる** をクリックします。アクティブな **結果** ウィンドウからフィルタが削除されます。

結果フィルタの削除

フィルタを適用しないで結果を表示するには、ユーザー定義フィルタを削除します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. 削除するユーザー定義フィルタを選択します。
ユーザーが定義したフィルタのみ削除できます。あらかじめ定義されているフィルタは削除できません。
4. **削除** をクリックし、**はい** をクリックして、フィルタの削除を確認します。
5. **閉じる** をクリックします。現在のユーザーの Silk Test Workbench データベースからフィルタが削除され、デフォルト フィルタ **なし** が開いているすべての結果に適用されます。

[条件] ダイアログ ボックス

条件 ダイアログ ボックスを使用して、今後すべてのテスト実行の合格条件として、ユーザー定義の検証の合格の割合を設定できます。

このダイアログ ボックスには、以下のオプションがあります。

合格の条件：今後すべての実行の合格条件として、検証の合格の割合を設定します。たとえば、合格条件を 90% に設定した場合、ビジュアル テストの 10 件の検証のうち 9 件に合格しないと、再生の結果が合格になりません。

また、このオプションを設定すると、再生結果の **結果合格条件** オプションも設定されます。


デフォルト：このボタンをクリックすると、合格条件がデフォルト値（100%）にリセットされます。

再生ステータス ダイアログ ボックスの有効化

再生ステータス ダイアログ ボックスを有効にすると、テストの再生中に実行される操作を表示できます。このダイアログ ボックスは、リモート Mac やモバイル デバイスなど、他のマシン上でテストを再生する場合に非常に役立ちます。

再生ステータス ダイアログ ボックスを有効化するには：

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。
3. **再生ステータス ダイアログ** をクリックします。
4. **再生ステータス** ダイアログ ボックスを有効化するには、**再生ステータス ダイアログの表示** オプションを **はい** に設定します。
5. **再生ステータス** ダイアログ ボックスにテスト対象アプリケーションのビデオまたはスクリーンショットを表示するには、**テスト対象アプリケーションの表示** オプションを **はい** に設定します。
6. **OK** をクリックします。

 **注**：リモート Mac やモバイル デバイス上でテストする場合は、Mac またはデバイスの画面がテスト中にオフにならないようにしてください。オフになると、**再生ステータス** ダイアログ ボックスには何も表示されなくなります。

[再生完了] ダイアログ ボックス

このダイアログ ボックスを使用して、再生が正常に完了したあとに実行する操作を指定できます。

次回からこのメッセージを表示しない チェック ボックスがオンになっているか、または **全般** オプションの **[再生完了] ダイアログの表示** ダイアログ オプションが **いいえ** に設定されている場合は、再生が正常に完了しても、このダイアログ ボックスは表示されません。この場合、再生が正常に完了すると、ビジュアル テストが **ビジュアル ナビゲータ** に表示され、スクリプトがスクリプト ウィンドウに表示されます。

再生完了 ダイアログ ボックスには、以下のオプションがあります。

もう一度再生する : ビジュアル テストまたはスクリプトを再生するには、このボタンをクリックします。このオプションを選択すると、再生が正常に完了したあと、**再生完了** ダイアログ ボックスが再び表示されます。

結果へ移動 : **結果** ウィンドウを開くには、このボタンをクリックします。結果の表示と操作の詳細については、「結果の概要」を参照してください。

ビジュアル テストへ移動 : ビジュアル テストの再生後にのみ表示されます。変更を保存せずに、**ビジュアル ナビゲータ** で現在のビジュアル テストを開くには、このボタンをクリックします。

.NET スクリプトへ移動 : スクリプトの再生後にのみ表示されます。変更を保存せずに、スクリプト ウィンドウで現在のテスト スクリプトを開くには、このボタンをクリックします。

保存 : 変更を保存したあとで現在のビジュアル テストまたはテスト スクリプトを表示するには、このボタンをクリックします。

次回からこのメッセージを表示しない : このチェック ボックスをオンにすると、**再生** ダイアログ ボックスが今後の再生で表示されなくなります。**全般オプション** の **[再生完了] ダイアログの表示** を **はい** に設定すると、このオプションを再度選択できるようになります。



ヒント : **記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、このダイアログ ボックスは表示されません。この場合、ビジュアル テストが記録されたときは **ビジュアル ナビゲータ** が表示され、スクリプトの場合はスクリプト ウィンドウが表示されます。

[再生エラー] ダイアログ ボックス

このダイアログ ボックスを使用して、再生エラーが発生したときに実行する操作を指定できます。

このダイアログ ボックスは、ビジュアル テストまたはスクリプトの再生中に Silk Test Workbench でエラーが発生した場合に表示されます。

再生エラー ダイアログ ボックスには、以下のボタンがあります。

続行 : エラーが発生したステップを再試行して、再生を再開します。エラーは、ビジュアル テストの再生結果の **詳細** タブに表示されます。エラーの解決後、このオプションを使用して、再生を再開します。このボタンは、ビジュアル テストでのみ有効になります。

終了 : エラーが発生したステップで再生を停止します。**再生完了** ダイアログ ボックスが表示され、ビジュアル テストまたはスクリプトを再生したり、再生結果を表示したり、ビジュアル テストまたはスクリプトに戻ったりすることができます。

デバッグ : デバッグ モードに切り替えます。デバッグ モードでは、再生を段階的に実行しながら、ビジュアル テストのエラーの診断や修正を行うことができます。このボタンは、ビジュアル テストでのみ有効になります。

ヘルプ : オンライン ヘルプ内のこのヘルプ トピックを表示します。

資産の管理

資産はテストプロジェクトの基本要素です。資産とは、定義してデータベースに保存するテスト要素のことです。資産は **アセットブラウザ** を使用して表示および管理できます。

ツールバー ボタンおよび **アセットブラウザ** を使用して資産を管理します。**開始画面** の **入門ガイド** ペインに移動して、ペインの **タスク** 領域の **アセットブラウザ** をクリックします。**アセットブラウザ** では、テストプロジェクトの資産を一元的に作成、管理、表示できます。

ほとんどの資産は、それを参照している他の資産を変更せずに、再利用や再定義ができます。

資産の種類

Silk Test Workbench は、次の種類の資産をサポートします。

ビジュアルテスト ビジュアルテストは、アプリケーションのテスト時にスクリーンショットを使用してユーザーの操作を模倣する、自動化されたテストです。ビジュアルテストは、自動テストソリューションの基本要素です。

ビジュアルテストの作成方法についての情報は、「[ビジュアルテストの作成](#)」を参照してください。

.NET スクリプト VB .NET スクリプトも、アプリケーションのテスト時にユーザーの操作を模倣する、自動化されたテストです。このスクリプトは、Microsoft Visual Basic .NET で作成されています。

VB .NET スクリプトの作成方法についての情報は、「[スクリプトの作成](#)」を参照してください。

キーワードテスト キーワード駆動テストは、実行可能なキーワードのコレクションです。キーワード駆動テストは、他のテストと同様に再生することができます。

キーワード駆動テストについての追加の情報は、「[キーワード駆動テスト](#)」を参照してください。

キーワード キーワードは、テストオブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。Silk Test Workbench でのキーワードは、ビジュアルテストまたはメソッド名の前に Keyword 属性を持つメソッドです。キーワードは、キーワード資産として保存されます。

キーワードシーケンスは、他のキーワードを組み合わせたものです。キーワードシーケンスは、頻繁に使用するキーワードの組み合わせを 1 つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。

キーワードについての追加の情報は、「[キーワード](#)」を参照してください。

結果 結果は、ビジュアルテストまたは .NET スクリプトの再生時に生成される資産です。結果は、テストプロセスを証明し、任意の時点におけるテストアプリケーションの機能の状態を永続的に記録します。

結果についての追加の情報は、「[結果の分析](#)」を参照してください。

オブジェクトマップ オブジェクトマップには、論理名とコントロールおよびウィンドウを関連付ける項目が格納されています。ロケータの代わりにこれらの論理名を使用して、アプリケーションのコントロールおよびウィンドウを識別できます。



注: オブジェクトマップ項目を含むオブジェクトマップの名前を変更すると、ルート項目名が変更されます。このオブジェクトマップを使用する既存のビジュアルテストも

またはスクリプトを、新しいオブジェクト マップのルート項目名を使用するように手動で変更する必要があります。

オブジェクト マップについての追加の情報は、「[オブジェクト マップ](#)」を参照してください。

イメージ検証 イメージ検証は、イメージがテスト対象アプリケーションの UI に含まれているかどうかをチェックします。

イメージ検証についての追加の情報は、「[イメージ検証](#)」を参照してください。

アクティブデータ アクティブ データは、ビジュアル テストでデータ ドリブン テストを作成するために使用されます。アクティブ データによって、強力で包括的なアプリケーション テスト ソリューションの入力として外部ファイルの既存のデータを利用できます。トランザクションごとに異なるデータセットを使用することにより、複雑なコードを記述したり、既存のデータを損なうことなく、テスト アプリケーションで複数のトランザクションを実行できます。

アクティブ データについての追加の情報は、「[アクティブ データ](#)」を参照してください。

イメージ資産 イメージ資産は、データベースに資産として格納されるイメージです。イメージ資産を使用して、Silk Test Workbench で認識されないコントロールをクリックできるようになります。

イメージ資産についての追加の情報は、「[イメージ資産](#)」を参照してください。

資産のバージョンをパージ

Silk Test Workbench は、資産を保存するたびに新しいバージョンを作成します。したがって、資産のバージョンの数が増加し、データベースのサイズも大きくなります。

データベースから資産のバージョンを削除するには、**資産のバージョンをパージ** ダイアログ ボックスを使用して、資産の種類、資産が属するプロジェクト、保持するバージョンの数、資産の変更日などの条件を指定します。



注: 資産の現在のバージョンは削除できません。この機能は、資産を完全に削除するためのものではなく、資産のバージョンのみを削除するものです。資産のバージョンをパージするには、スクリプト作成者かフル アクセス権限が必要です。

資産のバージョンをパージ

資産のバージョンをパージする前に、ログイン中のユーザーが必要なアクセス権を持っていることを確認してください。パージの実行に必要なアクセス権は、グローバル オプションで変更できます（管理者権限が必要）。実行に必要なアクセス権の変更についての情報は、[グローバル オプション](#) を参照してください。

1. **ファイル > 資産のバージョンをパージ** をクリックします。 **資産のバージョンをパージ** ウィザードが表示されます。
2. **次へ** をクリックします。
3. 資産をパージする方法を選択します。
 - 特定の種類の資産をパージするには、**資産の種類によるパージ** をクリックします。
 - 個々の資産をパージするには、**個々の資産のパージ** をクリックします。
4. [資産の種類によるパージ] を選択した場合は、次の手順に従います。
 - a) 資産のバージョンをパージする資産の種類をチェックします。
 - b) 選択した資産の種類のバージョンをパージするプロジェクトをチェックします。
5. [個々の資産のパージ] を選択した場合は、次の手順に従います。
 - a) プロジェクト ツリーで、バージョンをパージする資産を含むプロジェクト ノードを展開します。
 - b) 資産の種類をクリックします。
 - c) リストから個々の資産をチェックします。

6. 省略可能：[資産のバージョンをパージ]のプロジェクトの選択状態を保持 オプションをオンにすると、資産のバージョンをパージ ウィザードで選択したプロジェクトを保持できます。

a) メニューから ツール > オプション を選択します。

b) オプション メニューで、グローバル > 資産管理 を選択します。

c) [資産のバージョンをパージ]のプロジェクトの選択状態を保持 オプションを はい に設定します。

このオプションを いいえ に設定すると、資産のバージョンをパージ ウィザードを開いたときに、プロジェクトは選択されません。

7. 次へ をクリックします。

8. 指定した数の資産のバージョンを保持するか、指定した日より前に変更されたすべての資産のバージョンを削除するかを選択します。

9. 指定した数の資産のバージョンを保持したい場合は、保持する資産のバージョンの数を指定します。

指定した数よりも古い資産のバージョンがデータベースから削除されます。たとえば、2 を指定すると、最新の2つのバージョンが維持され、残りのバージョンが削除されます。維持するバージョンの最低数は1で、最大数は32767です。



注：すべての実際の検証はパージ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除されません。パージ後に、指定したものより多くの予想した検証が保持される可能性があります。たとえば、予想した検証の5バージョンがテスト結果によって参照されており、パージ後に各資産ごとに3バージョンのみ残したい場合、結果によって参照されている検証の5バージョンすべてが保持されます。

10 指定した日より前に変更されたすべての資産のバージョンを削除したい場合は、定義済みの日付を選択するか、1970/01/01 から 3000/12/31 の間の日付を指定します。



注：すべての実際の検証はパージ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除されません。これによって、指定した日付よりも古い予想した検証が残されるという状況が生じる可能性があります。

11 次へ をクリックします。資産のバージョンをパージ ウィザードは、削除する資産のバージョンの要約を表示します。

12 次へ をクリックします。



注：資産または複数の資産をパージできなかった場合は、メッセージボックスにパージできなかった資産の数が示されます。通常、これは別のユーザーが資産の現在のバージョンを開いているか、パージ中にその資産がすでに削除されていることが原因です。


13 完了 をクリックします。


パージのコマンド ライン パラメータ

GUI を使用する代わりに、コマンド ラインを使用して資産のバージョンをパージできます。

コマンド ライン プログラムの名前は STWPurge.exe であり、次の場所にあります：¥Silk¥SilkTest¥ng ¥gui¥。

パラメータ	構文	例	説明
ユーザー名	u	-u admin	データベースへの接続を許可されているユーザーの名前を指定します。このパラメータは必須です。
	username	/username JohnSmith	
パスワード	p	-p admin123	データベースへの接続を許可されているユーザーの大文字小文字が区別されるパスワードを指定します。このパラメータはパスワードが Nothing 以外の場合に必要です。
	password	/password admin123	

パラメータ	構文	例	説明
DSN	d dsn	-d mydsn /dsn mydsn	データベースへの接続に使用するデータソース名 (DSN) の名前を指定します。このパラメータは省略可能です。省略した場合は、デフォルトの DSN が使用されます。
資産の種類	s assettypes	-s visualtest script /assettypes visualtest	<p>資産バージョンを削除する資産の種類を指定します。このパラメータは省略可能です。省略した場合は、すべての資産の種類が対象となります。このパラメータは、以下の値を受け入れます (構文はカギ括弧で示します、複数値は空白で区切って指定できます)。</p> <ul style="list-style-type: none"> • ビジュアル テスト [visualtest] • オブジェクト マップ [objectmap] • 結果 [result] • アクティブデータ [activedata] • .NET スクリプト [script]
プロジェクト	r prj projects	-r commonprojectA /projects common projects	<p>資産のバージョンを削除するプロジェクト、資産の種類、個々の資産を指定します。このパラメータは省略可能です。省略した場合は、すべてのプロジェクトが対象となります。複数の項目を指定する場合は、空白で区切って入力します。キャレット記号 (^) を含む項目は、引用符で囲む必要があります。</p> <p>次の例は、プロジェクトパラメータを使用して、<i>Login</i> という名前のビジュアルテストから古い資産のバージョンをパージします。</p> <pre>-projects Common.VisualTest.Login</pre> <p>次の例は、プロジェクトパラメータを使用して、<i>MyProject</i> という名前のプロジェクトのすべてのアクティブデータ資産から古い資産のバージョンをパージします。</p> <pre>-projects MyProject.ActiveData</pre> <p>projects パラメータを使用して資産の種類や個々の資産を指定する場合は、assettypes パラメータは指定できません。ただし、プロジェクトを指定する場合は同時に指定することができます。例：</p> <pre>-projects Common -assettypes script</pre>
数によるパージ	pc purgebycount	-pc 10 /purgebycount	<p>指定された数を超える古い資産バージョンは削除されます。このパラメータまたは日付によるパージパラメータのいずれかが必要です。両方のパラメータを使用することはできません。維持するバージョンの最低数は 1 で、最大数は 32767 です。資産のアクティブなバージョンは削除できません。このパラメータは、単独の数値を受け入れます。たとえば、10 を指定した場合、資産の 10 の最新バージョンが維持され、それより古いバージョンは削除されます。</p> <p> 注: すべての実際の検証はパージ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除され</p>

パラメータ	構文	例	説明
			<p>ません。ページ後に、指定したものより多くの予想した検証が保持される可能性があります。たとえば、予想した検証の 5 バージョンがテスト結果によって参照されており、ページ後に各資産ごとに 3 バージョンのみ残したい場合、結果によって参照されている検証の 5 バージョンすべてが保持されます。</p>
日付による ページ	pd purgebydate	-pd 2011-03-28 /purgebydate 2012-03-28	<p>指定された日付以前に変更された資産バージョンを削除します。このパラメータまたは数によるページパラメータのいずれかが必要です。両方のパラメータを使用することはできません。このパラメータは、1970-01-01 から 3000-12-31 までの日付を次の形式で受け入れます：[YYYY-MM-DD]。</p> <p> 注: すべての実際の検証はページ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除されません。これによって、指定した日付よりも古い予想した検証が残されるという状況が生じる可能性があります。</p>
情報モード	i info	-i silent /info verbose	<p>プログラムの実行中に表示する情報の種類を設定します。このパラメータは省略可能です。省略した場合は、デフォルトモードの「通常」が使用されます。このパラメータは、以下の値を受け入れます (構文はカギ括弧で示します)。</p> <p>通常 [n normal] すべてのエラーと一般情報と出力します。</p> <p>サイレント [s silent] 実行が停止するエラーのみを出力します。</p> <p>詳細 [v verbose] すべての警告、エラー、および一般情報と出力します。</p>
確認	o prompt	-o /prompt	<p>資産ページレポートを表示した後に、確認メッセージを表示します。このパラメータは省略可能です。</p>
ヘルプ	h ? help	-h /help	<p>ヘルプの内容を表示し、その他のすべてのパラメータを無視します。</p>

例 1

この例は、*Common* プロジェクトのすべての結果から、結果ごとに 5 バージョンを残してページします。

```
stwpurge /dsn SilkTest /username admin /password admin /purgebycount  
5 /projects Common /assettypes result
```

例 2

この例は、*LoginTest* ビジュアルテストから、2017年12月31日より前に作成されたすべてのバージョンをパージします。

```
stwpurge /dsn SilkTest /username admin /password admin /purgebydate  
2016-12-31 /projects Common.VisualTest.LoginTest
```

アセットブラウザを使用した資産の作成

ほとんどの資産は、既存のビジュアルテストまたはスクリプトに挿入することによって作成されるか、または記録時に作成されます。ただし、**アセットブラウザ** を使用して資産を作成することもできます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. 資産のリストで、作成する資産を右クリックします。この種類の資産を新規作成するには、**新規作成 > 資産の種類** を選択します。

すべての資産の種類をリストするには、**新規作成** を選択します。

すべての資産は、資産の命名規則に従う必要があります。

新しい資産は、現在アクティブになっているプロジェクトに追加され、そのプロジェクト内の他の資産に対して、またはアクティブプロジェクトを参照しているプロジェクトの資産に対してのみ使用可能になります。アクティブプロジェクトを変更する場合は、ウィンドウの**プロジェクト**領域の**アクティブ**リストから目的のプロジェクトを選択します。複数のプロジェクトで使用する資産を格納するには、Commonプロジェクトを使用します。Commonプロジェクトの資産は、Commonプロジェクトを参照しているすべてのプロジェクトで使用できます。

スクリプトから資産を開く

スクリプトを編集しているときに、資産を右クリックして**資産を開く**を選択し、資産を開くことができます。これにより、GUIで資産が開きます。

資産がシステム上のファイルへの参照である場合 (ImageClickFile によって参照される場合など)、ファイルはシステムのデフォルトエディターで開かれます。

資産を開く

アセットブラウザ を使用して、資産を開きます。

1. **ファイル > 開く** を選択するか、**開始画面** の **入門ガイド** ペインに移動して、**タスク** 領域の **アセットブラウザ** をクリックします。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. 右側のペインで、開く資産をダブルクリックします。

資産の比較

同じ種類の2つの異なる資産や、同じ資産の2つのバージョンを比較して、これらの間の相違点を調べることができます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。

2. 資産の種類 リストから資産の種類を選択します。

次の資産の種類に対して、資産や資産のバージョンを比較できます。

- ビジュアル テスト
- .NET スクリプト
- オブジェクト マップ

選択したプロジェクトの選択した種類の既存の資産が、右側のペインに表示されます。



ヒント: 調査したい資産が表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 同じ種類の 2 つの異なる資産を比較するには :

- a) **Ctrl + クリック** を使用して、比較する 2 つの資産を選択します。
- b) 選択した項目を右クリックします。
- c) **相違点の表示** をクリックします。

4. 同じ資産の 2 つのバージョンを比較するには :

- a) 調査したい資産を右クリックして、**すべてのバージョンを表示** を選択します。
- b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
- c) **相違点の表示** をクリックします。

Silk Test Workbench は、次の相違点を表示します。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

資産の複製

複製した資産には、元の資産のすべての属性が保持されます。資産を複製することにより、似たような属性の同じ種類の資産が複数必要な場合に、時間を節約することができます。資産を作成して保存したあとは、いつでもそれを複製できます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. **アセット ブラウザ** の右側のペインで、複製する資産を右クリックし、**複製** を選択します。**資産の複製** ダイアログ ボックスが開きます。
4. 複製資産の名前を入力し、必要に応じて資産を保存するプロジェクトを変更し、新しい資産の説明を入力します。
5. **OK** をクリックします。

資産名の変更

作成して保存した資産は、いつでも名前を変更できます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. **アセット ブラウザ** の右側のペインで、名前を変更する資産を右クリックし、**名前の変更** を選択します。
名前を変更したい資産を選択して、**F2** を押しても変更できます。
資産名の変更 ダイアログ ボックスが開きます。
4. 資産の新しい名前と、説明を入力します。
5. 省略可能 : 資産を他のプロジェクトに移動する場合は、資産の移動先のプロジェクトを選択します。

ビジュアルテストやVB.NET スクリプトを他のプロジェクトに移動する場合に、この機能を使用できません。

6. 省略可能：資産の説明を入力します。
7. **OK** をクリックします。
8. 資産が他の資産から参照されている場合は、その処理方法を選択します。
 - 資産名を変更して、参照を更新する場合は、**この資産の名前を変更して参照を更新する** をクリックします。
 - 参照を更新せずに資産名を変更する場合は、**この資産の名前のみ変更する** をクリックします。



注: 参照している資産は、正しく再生できなくなる可能性があります。

- 資産名の変更をキャンセルする場合は、**キャンセル** をクリックします。

資産を別のプロジェクトに移動する場合に、新しいプロジェクトが参照している資産のプロジェクトから参照されていないと、参照している資産にエラー アイコンが表示されます。エラー アイコンにマウスカーソルを移動させると、エラーの情報が表示されます。

資産の削除

資産を作成して保存したあとは、いつでもそれを削除できます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** で適切な資産の種類を選択します。
3. 資産のリストで削除する資産を右クリックし、**削除** を選択します。**削除** ダイアログ ボックスが開きます。資産に複数のバージョンがある場合は、Silk Test Workbench により最新のバージョンを削除するかすべてのバージョンを削除するかのオプションが表示されます。
4. 資産に複数のバージョンがあり、そのすべてを削除する場合は、**すべてのバージョンを削除** を選択して **はい** をクリックします。複数のバージョンがない場合、現在（つまり唯一）のバージョンの資産を削除するには **はい** をクリックし、資産を削除せずに **削除** ダイアログ ボックスを閉じるには **いいえ** をクリックします。

資産リストのフィルタ処理

フィルタ リストを使用すると、特定の条件を満たす資産だけを表示できます。

1. **アセットブラウザ** ウィンドウで、**フィルタ** をクリックします。**フィルタ** ダイアログ ボックスが開きます。
2. 使用するフィルタを 1 つまたは複数指定します。

複数のフィルタを指定した場合、すべての条件を満たす資産が表示されます。以下は、フィルタの種類です。

作成者 ユーザーの名前を選択すると、そのユーザーによって作成されたすべての資産が表示されます。

作成日 開始日と終了日を指定すると、その期間中に作成されたすべての資産が表示されます。


最終更新者 ユーザーの名前を選択すると、そのユーザーによって更新されたすべての資産が表示されます。

最終更新日 開始日と終了日を指定すると、その期間中に更新されたすべての資産が表示されます。

3. **アセットブラウザ** で特定の名前を持つ資産のみを表示するには、**名前フィルタ** テキスト ボックスに名前を入力します。

アセットブラウザ には、**名前フィルタ** と他の任意のフィルタ条件に一致する資産が表示されます。

4. **OK** をクリックします。Silk Test Workbench により、現在表示されている資産項目のリストにフィルタが適用されます。

 **注:** フィルタを解除して、再びすべての項目を表示するには、**フィルタ** をクリックし、**フィルタ** ダイアログ ボックスで **デフォルト** をクリックします。

アセット ブラウザ で特定の名前を持つ資産のみを表示するには、**名前フィルタ** テキスト ボックスに名前を入力します。他の任意のフィルタも表示される資産セットに影響を与えます。

資産の検索

アセット ブラウザ の **名前フィルタ** フィールドを使用して、名前を指定して資産を検索できます。

1. **アセット ブラウザ** を開きます。
2. 資産の種類を選択します。
3. **名前フィルタ** フィールドをクリックします。
代わりに、**F3** を押すこともできます。
4. 検索する資産の名前を指定します。
資産の名前を入力していくと、**名前フィルタ** フィールドの文字に一致する資産だけが Silk Test Workbench に表示されます。

資産の並べ替え

アセット ブラウザ の項目は、列の内容に従って並べ替えることができます。デフォルトでは、資産のリストは昇順に並んでいます。

1. **アセット ブラウザ** に表示された項目の並べ替えに使用する列の見出しをクリックします。
たとえば、項目を日付順に並べるには、**作成日** 列の見出しをクリックします。
2. 並べ替えの方向を逆にするには、**アセット ブラウザ** に表示された項目の並べ替えに以前使用した列の見出しをクリックします。
たとえば、作成日による並べ替えの順序を逆にするには、**作成日** 列の見出しを 2 回クリックします。

資産保存時のデフォルト動作の設定

資産を保存するときの、デフォルトの保存動作を設定します。動作は、資産の新しいバージョンを作成する、または資産の現在のバージョンを置き換える、のいずれかから選択します。


1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **グローバル** の隣にあるプラス記号 (+) をクリックし、**資産管理** をクリックします。**資産管理** オプションが右側のパネルに表示されます。
3. **デフォルトの保存動作** リスト ボックスから、**新しいバージョンとして保存** または **現在のバージョンとして保存** を選択します。ツールバーのボタンやショートカット キー (Ctrl+S) で資産を保存するときに、Silk Test Workbench によって、資産の新しいバージョンが作成されるか、資産の現在のバージョンが置き換えられます。
4. **OK** をクリックします。

資産のバージョンの最大数の設定

テスト結果を含む資産のストレージの容量を制限するために、保持する資産のバージョンの最大数を指定できます。

1. メニューで、**ツール > オプション** を選択します。**オプション** ダイアログ ボックスが開きます。
2. **オプション** ツリーで、**グローバル** を展開します。
3. **資産管理** を選択します。
4. **資産管理** ページの **資産のバージョンの最大数** テキスト ボックスに保持するバージョンの数を入力します。
デフォルトの設定は 0 で、すべての資産のバージョンが保存されることを意味します。
5. **OK** をクリックします。

バージョンの最大数が設定され、その数に到達すると、資産の新しいバージョンが保存され、最も古い資産のバージョンが削除されます。資産をインポートする場合、バージョンの数が最大数を超過していたとしても、すべてのバージョンがインポートされます。このようにしてインポートされた資産に対して新しいバージョンが保存されるとき、指定した資産の最大数と一致するように、最も古いバージョンの資産から削除されます。

 **注:** 資産のバージョンの数を制限すると、古いテスト結果は自動的に削除されます。

インクルードする資産の表示

他の複数のテスト資産のテスト資産を再利用することができます。たとえば、ビジュアル テストや VB .NET スクリプトで同じオブジェクト マップを使用できます。各スクリプトには、使用する他のテスト資産を明示的に含める必要があります。Silk Test Workbench は VB .NET 以外の資産に対しては自動的にインクルード参照を作成しますが、VB .NET 資産に対しては、それを使用する各スクリプトにユーザーが明示的に含める必要があります。

インクルード情報 ペインを使用して、選択した資産に含まれる資産と選択した資産を含む資産を表示できます。たとえば、スクリプトを編集したときに、このペインを使ってスクリプトを利用するほかのすべてのスクリプトを探して実行し、動作することを確認できます。

また、オブジェクト マップを変更したときにも、**インクルード情報** ペインを使用できます。たとえば、オブジェクト マップ ID の名前を変更する場合、Silk Test Workbench は、そのオブジェクト マップを使用するスクリプトで ID を自動的に更新しません。**インクルード情報** ペインを使うと、変更したオブジェクト マップ ID に必要なスクリプトを探し出すことができます。

1. **アセットブラウザ** で資産 を選択します。
2. **表示 > インクルード** を選択します。**インクルード情報** ペインが開きます。
3. **インクルード情報** ペインで、**インクルード** タブをクリックして選択した資産に含まれる資産を表示するか、**インクルード元** タブをクリックして選択した資産を含む資産を表示します。

複数バージョンの資産の表示

Silk Test Workbench では、すべての種類の資産のバージョンが自動的に格納されます。資産を変更するたびに、その資産の新しいバージョンがデータベースに格納されます。資産のすべてのバージョンのリストと、資産を最後に更新したユーザー、更新日時、バージョン番号などのバージョン情報をいつでも表示できます。

Silk Test Workbench では、資産のアクティブ バージョンだけを参照します。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. 資産のリストで表示する資産を右クリックし、**すべてのバージョンを表示** を選択します。**バージョンの詳細** ダイアログ ボックスが開きます。
4. **バージョンの詳細** ダイアログ ボックス下部に並んでいるボタンを使用して、資産のバージョンを開いたり、複製したり、削除したり、閉じたりできます。

- バージョンを選択し、**アクティブバージョンにする** をクリックすると、任意のバージョンを現在アクティブなバージョンにできます。

資産の命名規則

資産の新規作成 ダイアログ ボックスで資産を追加すると、Silk Test Workbench によって名前がチェックされ、所定の命名規則に従っているかどうかを検証されます。

Silk Test Workbench では、以下の命名規則が使用されます。

- 資産名の長さは 1 から 126 文字である必要があります。
- 資産名に ¥ / < > " ' : * ? | = の各文字を使用することはできません。
- 同一プロジェクト内で同じ種類（またはグループ）の他の資産と同じ名前を付けることはできません。

テスト資産のインポートとエクスポート

インポートおよびエクスポートユーティリティを使用して、異なるデータベース間で資産を転送できます。資産のインポートおよびエクスポートによって、プロジェクト管理が簡単になります。また、WAN を介して資産を共有したり、外部のバージョン管理システムを活用したりできます。資産をインポートまたはエクスポートするには、**資産インポート** ウィザードまたは **資産エクスポート** ウィザードを使用できます。いずれの方法でも、必要に応じて資産のインポートとエクスポートをカスタマイズできます。

インポートおよびエクスポートが可能な資産は、プロジェクト資産とシステム資産です。

プロジェクト資産 詳細については、「資産の種類」を参照してください。

システム資産 記録と再生のオプション。

資産のインポート

資産のデータベース間でのインポートは、**資産インポート** ウィザードを使用して行います。

1. **ファイル > 資産のインポート** を選択します。 **Silk Test Workbench 資産インポート** ウィザードが開きます。

このウィザードおよびそれ以外のすべての Silk Test Workbench ウィザードを今後使用するとき、**よろこ** ページをスキップするには、このページの最下部にあるチェックボックスをオンにします。このチェックボックスをオンにすると、**全般** オプションの **ウィザードの [よろこ] 画面の表示** オプションが更新されます。

ウィザードのサイズは、任意のページの右下隅をドラッグして変更できます。

2. **次へ** をクリックします。 **インポート ファイル名** ページが開きます。
3. インポートする資産を含んだ ZIP ファイルを指定します。
4. **次へ** をクリックします。 **インポートする項目** ページが開きます。
5. 以下のいずれかのステップを実行して、インポートする項目を選択します。

- すべての資産をインポートするには、**すべての資産をインポートする** をクリックします。

資産のすべてのバージョンをインポートするには、**各資産の最新バージョンだけをインポートする** チェックボックスをオフにします。


- 特定の資産をインポートするには、**選択した資産だけをインポートする** をクリックします。

各資産の特定のバージョンをインポートするには、**各資産のすべてのバージョンを表示する** チェックボックスをオンにします。その後、**次へ** をクリックします。

インポートする資産のチェックボックスをオンにします。


6. **次へ** をクリックします。 **プロジェクトの選択** ページが開きます。

7. **省略可能** : ソースプロジェクトの資産をインポートするターゲットプロジェクトを指定するには、**ターゲット** 列でプロジェクトを選択します。

 **注:** 現在のユーザーが管理者であれば、新しプロジェクトを作成する項目を **ターゲット** リストから選択して、資産を新しいプロジェクトにインポートできます。

ターゲット リストにプロジェクトが含まれていない場合は、次の原因が考えられます。

- インポートファイルのプロジェクトがデータベースに存在せず、現在のユーザーが管理者でない
- 現在のユーザーが、既存のプロジェクトに対して実行者以上の権限を持たない

 **注:** 管理者であれば、新しいプロジェクトを作成して資産をインポートすることができます。

8. **次へ** をクリックします。 **競合の解決** ページが開きます。

9. インポートする資産名が既存の資産名と競合しているプロジェクト資産について、競合の解決方法を選択します。

次のいずれかを選択します。

- **インポートした資産を既存の資産の最新バージョンとして追加する** : このオプションは、データベース内の資産の新しいバージョンを作成します。インポートされる資産のバージョンが複数ある場合、資産は最小のバージョンが先に、最大のバージョンが最後にインポートされます。つまり、最大のバージョン番号を持つ資産がデータベースの最新の資産バージョンになります。
- **インポートした資産を追加、名前変更、またはスキップするか確認する** : これは対話オプションで、各競合に対してダイアログボックスを生成して、インポート資産を追加するか、名前変更するか、またはスキップするよう要求します。
- **資産をインポートしない** : 資産のインポートをスキップします。

10. インポートする資産名が既存の資産名と競合しているシステム資産について、競合の解決方法を選択します。

次のいずれかを選択します。


- **インポートした資産を名前変更またはスキップするか確認する** : これは対話オプションで、各競合に対してダイアログボックスを生成して、インポート資産を名前変更するか、またはスキップするよう要求します。
- **資産をインポートしない** : 資産のインポートをスキップします。

11. **次へ** をクリックします。 **資産のインポートの開始** ページが開きます。

12. 要約を確認します。

資産のインポートでは、次の点にご注意ください。


- ZIP ファイルに含まれるプロジェクトのうち、ターゲットデータベースに存在しないプロジェクトが作成されます。
- データベースに既に存在するプロジェクトには、プロジェクト参照は追加されません。要約には、このような競合が表示されます。必要に応じて、プロジェクト間の参照を手動で追加してください。

 **注:** 19.0 より前のバージョンの Silk Test Workbench で作成した ZIP ファイルをインポートする場合は、プロジェクト間の参照は ZIP ファイルに含まれておらず、Silk Test Workbench は新しく作成したプロジェクトに Common プロジェクトへの参照を追加します。

13. **次へ** をクリックして、インポートを開始します。 **インポートの進捗状況** ウィンドウに、完了レベルを示す進行状況バーが表示されます。インポートが完了すると、**インポートウィザードの完了** ページが開きます。

14. **省略可能** : **保存** をクリックして、詳細レポートを保存します。

15. **完了** をクリックします。

 **注:** Silk Test Workbench 18.5 以降を使用している場合、18.5 より前のバージョンの Silk Test Workbench でエクスポートした資産をインポートできません。以前のバージョンでエクスポートした資産をインポートするには、このような資産をインポートする場合は、

StwConvertExportedXML プログラムを使用してください。このコマンドラインプログラムを使用すると、資産を新しい形式に変換できます。このプログラムには、次の引数を指定できます。

```
STWConvertExportedXML [-overwrite] [-?] path_to_old_export path_to_new_export.zip
```

- path_to_old_export: Silk Test Workbench の以前のバージョンで生成したファイルまたはフォルダへのパス。
- path_to_new_export.zip: 変換したエクスポート ファイルを保存する ZIP ファイルへのパス。
- -overwrite: ZIP ファイルが既に存在するときに上書きする場合に指定します。-overwrite を指定しない場合、ZIP ファイルが存在すると、エラー メッセージを出力してプログラムが終了します。
- -?: ヘルプ メッセージを出力してプログラムが終了します。



注: 資産のインポートでは、履歴データはすべてインポート時の情報に置き換えられます。たとえば、作成者、作成日、最終更新者、更新日には、インポートを実行したユーザーおよびインポートの日付と時刻が反映されます。

資産のエクスポート

資産のデータベース間でのエクスポートは、**資産エクスポート** ウィザードを使用して行います。



注: エクスポート ファイルは変更しないでください。変更すると、ファイルのインポート時に Silk Test Workbench データベースが破損する可能性があります。資産の編集中にエクスポートが発生した場合、編集中の資産はエクスポートされません。そのため、エクスポートを実行する前に、資産に対するすべての編集を完了し、保存していることを確認してください。

1. **ファイル > 資産のエクスポート** を選択します。Silk Test Workbench **資産エクスポート** ウィザードが開きます。

このウィザードおよびそれ以外のすべての Silk Test Workbench ウィザードを今後使用するとき、**よろこ** ページをスキップするには、このページの最下部にあるチェックボックスをオンにします。このチェックボックスをオンにすると、**全般** オプションの **ウィザードの [よろこ] 画面の表示** オプションが更新されます。

ウィザードのサイズは、任意のページの右下隅をドラッグして変更できます。

2. **次へ** をクリックします。 **エクスポート対象** ページが開きます。
3. すべての資産をエクスポートするには、以下のステップを実行します。
 - a) **すべての資産をエクスポートする** をクリックします。
 - b) **各資産の最新バージョンだけをエクスポートする** チェックボックスをオフにして各資産のすべてのバージョンをエクスポートするか、このチェックボックスをオンにして最新バージョンだけをエクスポートに含めます。
4. 特定の資産をエクスポートするには、以下のステップを実行します。
 - a) **選択した資産だけをエクスポートする** をクリックします。
 - b) 各資産の特定のバージョンをエクスポートするには、**各資産のすべてのバージョンを表示する** チェックボックスをオンにします。



注: 資産の複数のバージョンをエクスポートするよう選択した場合は、選択するバージョンに応じて、バージョン情報がインポート時に変更することがあります。たとえば、資産のバージョン 1 と 5 をエクスポートするよう選択した場合、インポート時にバージョン番号が 1 と 2 に更新されます。

- c) **ビジュアル テストおよび .NET スクリプトに関連付けられた資産をエクスポートする** チェックボックスをオンにすると、すべての参照プロジェクト内のすべてのオブジェクト マップを含む、すべての関連資産をエクスポートできます。
- d) **次へ** をクリックします。 **エクスポートする項目** ページが開きます。
- e) 左側のペインでエクスポートする項目を含むフォルダを展開し、エクスポートする資産のチェックボックスをオンにします。

青いチェック マークは、項目とそのすべてのサブ項目がエクスポート対象として選択されていることを示します。灰色のチェック マークは、サブ項目の一部だけが選択されていることを示します。

5. **次へ** をクリックします。 **データの格納場所** ページが開きます。
6. 資産をエクスポートする ZIP ファイルの場所を指定します。
エクスポートしたすべての資産が、指定したファイルにまとめられます。
7. **次へ** をクリックします。 **資産のエクスポートの開始** ページが開きます。このページでは、エクスポートする資産の概要が表示されます。
8. 要約を確認し、**次へ** をクリックしてエクスポートを開始します。 **エクスポートの進捗状況** ウィンドウに、完了レベルを示す進行状況バーが表示されます。エクスポートが完了すると、**エクスポートウィザードの完了** ページが開きます。
9. 省略可能： **保存** をクリックして、詳細レポートを保存します。
- 10 **完了** をクリックします。

資産のインポートとエクスポートの権限

ユーザーに割り当てられている権限に応じて、資産のインポート機能を制限したり、無効にしたりすることができます。権限は、プロジェクトごとに固有に設定できます。たとえば、*ProjectA* に対する読み取り専用権限を持つユーザーに対して、*ProjectB* ではフル アクセス権限を割り当てることができます。この結果、このユーザーにとってインポート機能は、*ProjectA* で無効であっても、*ProjectB* では使用可能になります。

資産のエクスポート機能は、通常は制限されていません。読み取り専用以上の権限を持つすべてのユーザーは、プロジェクト資産とシステム資産の両方をエクスポートできます。

以下の表に、各ユーザー ロールが、プロジェクト資産およびシステム資産をインポートできるかどうかを示します。プロジェクト資産とシステム資産は、以下のように定義されます。

- プロジェクト資産：詳細については、「資産の種類」を参照してください。
- システム資産：記録オプションと再生オプション。

資産のインポート権限

管理者またはフル アクセス権限を持つユーザーのみが、すべてのプロジェクト資産とシステム資産をインポートできます。インポート権限の完全なリストについては、以下の表を参照してください。

ユーザー ロール	プロジェクト資産のインポート	システム資産のインポート
アクセス不可	いいえ	はい
読み取り専用	いいえ	はい
実行者	はい ¹	はい
スクリプト作成者	はい	はい
フル アクセス	はい	はい

¹：結果のインポートのみが許可されます。

資産のエクスポート権限

読み取り専用以上の権限を持つユーザーは、すべてのプロジェクト資産とシステム資産をエクスポートできます。

オブジェクト解決

Silk Test Workbench は、コントロール クラスの名前と優先付けられた属性のコレクションの組み合わせによって、テスト対象アプリケーション (AUT) のコントロールをユニークな XPath ロケーターに識別します。組み合わせた XPath ロケーターでコントロールをユニークに識別できない場合には、Silk Test Workbench は、ロケーターにインデックスを追加します。Silk Test Workbench で記録中に、**操作の選択** ダイアログの **ロケーター** フィールドのリストから、コントロールの他のロケーターを選択できます。

Silk Test Workbench では、識別されたオブジェクトのリテラル参照はロケーターと呼ばれます。Silk Test Workbench は、ロケーターを使用して、テスト対象アプリケーション (AUT) のオブジェクトを検索して識別します。ロケーターは、W3C (World Wide Web Consortium) によって定義された 共通の XML ベース言語である XPath クエリー言語のサブセットです。

ビジュアル テストまたはスクリプトを記録すると、Silk Test Workbench によってロケーターが自動的に記録されます。デフォルトでは、Silk Test Workbench は、コントロールのリテラル参照またはウィンドウのロケーター名を使用する代わりに、エイリアスまたは論理名 (オブジェクト マップ項目と呼ばれる) を使用します。ビジュアル テストおよびスクリプトでは、実際のロケーター名ではなく、オブジェクト マップ項目に基づいてオブジェクトが参照されます。

オブジェクト マップを無効にすると、オブジェクト マップ項目ではなくロケーターがビジュアル テストまたはスクリプト コンテキストに組み込まれます。

ロケーターの基本概念

Silk Test Workbench は、XPath クエリー言語のサブセットをサポートしています。XPath の詳細については、<http://www.w3.org/TR/xpath20/> を参照してください。

XPath 式は現在のコンテキスト、つまり、Find メソッドを呼び出したオブジェクトの階層上における位置に依存します。ファイル システムと同じように、すべての XPath 式は、この位置に依存します。例：

- `"//Shell"` は、現在のコンテキストから始まるすべての階層にあるすべての Shell を見つけます。
- `"Shell"` は、現在のコンテキストの直下の子であるすべての Shell を見つけます。

さらに、ある XPath 式は、コンテキストの影響を受けます。たとえば、`myWindow.find(xpath)` は、`myWindow` が現在のコンテキストとなります。

動的オブジェクト解決は、テスト ケース内でオブジェクトを識別するために、Find または FindAll メソッドを使用します。

オブジェクト タイプと検索範囲

典型的なロケーターには、検索するオブジェクトのタイプと検索範囲が含まれます。検索範囲は以下のいずれかです。

- `//`
- `/`

ロケーターは、ロケーターを指定する対象となるオブジェクトである、現在のオブジェクトに依存します。現在のオブジェクトは、アプリケーション UI のオブジェクト階層における位置を特定します。ファイル システムと同じように、すべてのロケーターは、この階層における現在のオブジェクトの位置に依存します。

XPath 式は、現在のコンテキスト、つまり、Find メソッドを呼び出したオブジェクトの階層上における位置に依存します。ファイル システムと同じように、すべての XPath 式は、この位置に依存します。

**注:**

HTML 要素に対するロケータにおけるオブジェクト タイプは、HTML タグ名または、このオブジェクトに対して Silk Test Workbench が使用するクラス名のいずれかになります。たとえば、ロケータ `//a` と `//DomLink` (ここで、`DomLink` は Silk Test Workbench でのハイパーリンクに対する名前です) は同じです。HTML ベースでないテクノロジーの場合は、Silk Test Workbench クラス名だけが使用されます。

例

- `//a` は、現在のオブジェクトに相対的なすべての階層にあるハイパーリンク オブジェクトを識別します。
- `/a` は、現在のオブジェクトの直下の子であるハイパーリンク オブジェクトを識別します。



注: `<a>` は、Web ページのハイパーリンクを表す HTML タグです。

例

以下のコード例は、ブラウザ内の最初のハイパーリンクを識別します。この例では、実行中のブラウザ インスタンスを参照するスクリプトに `browserWindow` という名前の変数が存在することを仮定しています。ここで、タイプは "a" で、現在のオブジェクトは `browserWindow` です。

```
Dim link As DomLink = browserWindow.DomLink("//a")
```

属性を使用したオブジェクトの識別

プロパティに基づいてオブジェクトを識別するために、ロケータ属性を使用できます。ロケータ属性は、オブジェクトの型の後にかぎ括弧で指定します。

例

次の例では、テキスト `Home` を持つハイパーリンクを識別するために、`textContent` 属性を使用します。同じテキストを持つハイパーリンクが複数存在する場合は、最初のハイパーリンクがロケータによって識別されます。

```
Dim link as DomLink = browserWindow.DomLink("//a[@textContent='Home']")
```

ロケータの構文

Silk Test Workbench では、UI コントロールを特定するために、XPath クエリ言語のサブセットをサポートしています。

以下の表に、Silk Test Workbench がサポートする構成子を一覧します。



注: `<a>` は、Web ページのハイパーリンクを表す HTML タグです。

サポートするロケータ構成子	例	説明
<code>//</code>	<code>//a</code>	現在のオブジェクトより下層にあるオブジェクトを識別します。

サポートするロケータ構成子	例	説明
/	/a	この例は、Web ページのハイパーリンクを識別します。 現在のオブジェクトの直下にあるオブジェクトを識別します。それより下層にあるオブジェクトは解決されません。
属性	例 1 : // a[@textContents='Home'] 例 2 : // div[@textContents='Price: * USD']	この例は、現在のオブジェクトの直下にある Web ページのハイパーリンクを識別します。 特定の属性によってオブジェクトを識別します。属性値には、ワイルドカード * および ? を使用できます。 例 1 は、テキスト <i>Home</i> を持つハイパーリンクを識別し、例 2 は、ワイルドカードを使って任意の価格を持つ div を識別します。
インデックス	例 1 : //a[3] 例 2 : // a[@textContents='Home'][2]	複数のオブジェクトがある場合に、インデックスを指定して特定のオブジェクトを識別します。ロケータのインデックスは 1 から始まります。 例 1 は、3 番目のハイパーリンクを識別し、例 2 は、テキスト <i>Home</i> を持つ 2 番目のハイパーリンクを識別します。
論理演算子 : and、or、not、=、!=	例 1 : // a[@textContents='Remove' or @textContents='Delete'] 例 2 : //a[@textContents! ='Remove'] 例 3 : // a[not(@textContents='Delete' or @id='lnkDelete') and @href='*/delete']	論理演算子を使って、属性を組み合わせてオブジェクトを識別します。 例 1 は、 <i>Remove</i> か <i>Delete</i> のテキストを持つハイパーリンクを識別し、例 2 は、 <i>Remove</i> 以外のテキストを持つハイパーリンクを識別し、例 3 は、複数の論理演算子を組み合わせた例を示しています。
ancestor	例 1 : //input[@id='username']/ ancestor::form 例 2 : //input[@id='username']/ ancestor::div[@className='c ontainer']	現在のオブジェクトより上層にあるオブジェクトを識別します。 例 1 は、識別子 <i>username</i> を持つ input 要素を子に持つ form 要素を識別し、例 2 は、識別子 <i>username</i> を持つ input 要素を子に持ち、クラス名が <i>container</i> を持つ div を識別します。
..	例 1 : // a[@textContents='Edit']/..	オブジェクトの親を識別します。 例 1 は、テキスト <i>Edit</i> を持つハイパーリンクの親を識別し、例 2 は、テキスト <i>Edit</i> を持つハイパーリンクと同

サポートするロケータ構成子	例	説明
	例 2 : // a[@textContents='Edit']/..// a[@textContents='Delete']	列にあるテスト <i>Delete</i> を持つハイパーリンクを識別します。
following-sibling	例 : // td[@textContents='John']/ following-sibling::td[2]	現在のオブジェクトより後ろにある同列オブジェクトを識別します。 この例は、テキスト <i>John</i> を持つテーブルセルの右側から 2 番目にあるテーブルセルを識別します。
preceding-sibling	例 : // td[@textContents='John']/ preceding-sibling::td[2]	現在のオブジェクトより前にある同列オブジェクトを識別します。 この例は、テキスト <i>John</i> を持つテーブルセルの左側から 2 番目にあるテーブルセルを識別します。
*	例 1 : // *[@textContents='Home'] 例 2 : /*/a	オブジェクトの型 (ハイパーリンク、テキストフィールド、ボタンなど) を考慮せずにオブジェクトを識別します。 例 1 は、オブジェクトの型は任意で、指定したテキストを持つオブジェクトを識別し、例 2 は、現在のオブジェクトより 2 つ下層にあるハイパーリンクを識別します。

以下の表には、Silk Test Workbench がサポートしないロケータ構成子を一覧します。

サポートしないロケータ構成子	例
右辺、左辺ともに属性を指定して比較する。	//a[@textContents = @id]
属性名を右辺に指定することはサポートされません。属性名は左辺に指定する必要があります。	//a['abc' = @id]
複数のロケータを and あるいは or で結合する。	//a[@id = 'abc'] or ../Checkbox
複数の属性をかぎ括弧で指定する。	//a[@id = 'abc'] [@textContents = '123'] (代わりに //a [@id = 'abc' and @textContents = '123'] を使用)
複数のインデックスをかぎ括弧で指定する。	//a[1][2]
クラスあるいはクラス名の一部にワイルドカードを含むクラスワイルドカードを明示的に指定しない構成子。	//[@id = 'abc'] (代わりに //*[@id = 'abc'] を使用) "/*//a[@id='abc']"

スクリプトでのロケータの使用

Silk Test Workbench では、識別されたオブジェクトのリテラル参照はロケータと呼ばれます。必要に応じて、ロケータ文字列の短縮形をスクリプトで使用できます。スクリプトを再生すると、Silk Test Workbench によって自動的に構文が展開されて完全なロケータ文字列が使用されます。スクリプトを手動でコーディングする場合は、次の順番で次の部分を省略できます。

- 検索スコープ「//」。
- オブジェクトの型名。Silk Test Workbench のデフォルトはクラス名です。
- 属性を囲む角かっこ「[]」。

スクリプトを手動で記述する場合は、使用可能な最も短い形式を使用することをお勧めします。



注: .NET スクリプトを使用する場合にのみ、ロケータ文字列の短縮形を使用できます。ビジュアルテストでは、完全なロケータ文字列を使用する必要があります。オブジェクトを識別する場合は、完全なロケータ文字列がデフォルトでキャプチャされます。

以下のロケータは同じです。

- 最初の例では、完全なロケータ文字列が使用されています。

```
_desktop.DomLink("//BrowserApplication//BrowserWindow//a[@textContents='Home']").Select()
```

完全なロケータ文字列を確認するには、**オブジェクトの識別** ダイアログ ボックスを使用します。

- 2 番目の例は、ブラウザー ウィンドウが既に存在する場合に機能します。

```
browserWindow.DomLink("//a[@textContents='Home']").Select()
```

または、短縮形を使用することができます。

```
browserWindow.DomLink("@textContents='Home']").Select()
```

識別のための実際の属性がないオブジェクトを検索するには、インデックスを使用します。たとえば、Web ページの 2 つめのハイパーリンクを選択するには、以下のように入力します。

```
browserWindow.DomLink("[2]").Select()
```

さらに、その種類の最初のオブジェクトを検索する（このことは、オブジェクトに実際の属性がない場合に便利です）には、以下のように入力します。

```
browserWindow.DomLink().Select()
```

Find メソッドの使用

.DomLink のようなメソッドを使用する代わりに、Find メソッドを使用してロケータを持つ単一のオブジェクトを識別することができます。



注: Find メソッドは、完全ロケータのみ使用できます。省略形のロケータはサポートされません。

以下のように入力する代わりに：

```
_desktop.DomLink("//a[@textContents='Home']").Select()
```

以下のように入力することができます。

```
_desktop.Find(Of DomLink)("//a[@textContents='Home']").Select()
```

.DomLink メソッドは、Find メソッドで内部的に使用されます。 .DomLink メソッドを使用した方が、Find メソッドよりも簡潔に表現できるため、好まれます。

ロケータを使用したオブジェクトの存在確認

Exists メソッドを使用して、オブジェクトがテスト対象アプリケーションに存在するかどうかを確認できます。

次のコードは、「Log out」というテキストのハイパーリンクが Web ページに存在するか確認します。

```
If (browserWindow.Exists( "//a[@textContents='Log out']" )) Then
    ' do something
End If
```

Find メソッドの使用

Find メソッドや FindOptions メソッドを使用して、後で使用したいオブジェクトが存在するか確認できます。

次のコードは、ウィンドウを検索し、ウィンドウが見つかった場合にウィンドウを閉じます。

```
Dim mainWindow As Window
mainWindow = _desktop.Find("//Window[@caption='My Window']", New FindOptions(False))
If (mainWindow IsNot Nothing) Then
    mainWindow.CloseSynchron()
End If
```

1 つのロケーターで複数のオブジェクトを識別する

FindAll メソッドを使用して、ロケーターに一致する最初のオブジェクトのみを識別するだけでなく、ロケーターに一致するすべてのオブジェクトを識別できます。

例

次のコードの例は、FindAll メソッドを使用して、Web ページのすべてのハイパーリンクを取得します。

```
Dim links As IList(Of DomLink) = browserWindow.FindAll(Of DomLink)
("//a")
```

ロケーターのカスタマイズ

このセクションでは、テスト対象アプリケーション (AUT) のコントロールを Silk Test Workbench が確実に解決できるようにするために、安定したロケーターを作成する方法について説明します。

Silk Test Workbench は、AUT がその UI コントロールに対して公開する識別子を利用して、非常に柔軟で強力な UI コントロールの識別方法を提供します。Silk Test Workbench は、任意の UI コントロールに対して宣言された任意のプロパティを使用して、UI コントロールの階層を使ってロケーターを作成できます。Silk Test Workbench は、それぞれの UI コントロールを識別するのに最も適した項目とプロパティを階層から選択します。

Silk Test Workbench は、UI コントロールの階層から多くのコントロールを動的に除外するため、AUT の変更に対して非常に影響を受けにくいオブジェクト解決方法を Silk Test Workbench は提供します。Web ページの書式要素のよな UI コントロール ツリーの階層を変更する中間のグループ化されたコントロールは、オブジェクト解決から除外することができます。

UI コントロールによっては、それを固有に識別できるようにする有意義なプロパティを公開しません。このようなコントロールを含んだアプリケーションは低いテスト容易性を持つアプリケーションとみなされます。階層、とくに動的な階層は、このようなアプリケーションに対する固有のロケーターを作成するのに、重要な意味を持ちます。高いテスト容易性を持つアプリケーションは、固有の UI コントロールを識別するための単純な仕組みを常に提供します。

AUT のテストを容易にする最も単純で最も効果的な慣例のひとつが、コントロールに対する安定した識別子を導入し、アプリケーションの既存のインターフェイスを通して、これらの安定した識別子を公開することです。

安定した識別子

UI コントロールの安定した識別子 とは、コントロールの呼び出しごとに、または UI コントロールが存在するアプリケーションのバージョンが変わっても変更されない識別子を言います。安定した識別子は、その使用されるコンテキストにおいて一意である必要があります。つまり、同じ識別子を持つコントロール

が同時にアクセス可能でないことが求められます。つまり、グローバルなコンテキストで一意である GUID 形式の識別子を使用する必要はありません。コントロールの識別子は、可読性が高く、有意な名前であるべきです。これらの識別子の命名規則によって、実際のコントロールに識別子を関連付けるのがより容易になります。

例：キャプションはコントロールの良い識別子と言えるか

ほとんどのテストツールでは、UI コントロールのデフォルト識別子として キャプションを使用します。キャプションは、コントロールに関連付けられた UI のテキストです。しかし、UI コントロールを識別するためにキャプションを使用することには、次のような欠点があります。

- キャプションは安定していません。キャプションは開発プロセスの間に頻繁に変更されます。たとえば、AUT の UI が開発プロセスの終わりにレビューされる場合があります。UI が安定していないため、開発プロセスの初期の段階で UI テストを導入することが困難になります。
- キャプションは一意ではありません。たとえば、アプリケーションには **OK** というキャプションを持つボタンが複数存在する可能性があります。
- 多くのコントロールはキャプションを表示しないため、識別するために、ほかのプロパティを使用する必要があります。
- ローカライズしたアプリケーションのテストにキャプションを使用する場合、各言語ごとにコントロールのキャプションを保守する必要があるため、非常に扱いにくく、さらに言語ごとに適切なキャプションを動的に割り当てることができるように、複雑なスクリプト ロジックを保守する必要があります。

安定したロケータを作成する

Silk Test Workbench の主要なメリットのひとつが、柔軟で強力なオブジェクト解決の仕組みです。UI コントロールを特定するために XPath 記法を使用することによって、UI コントロールが適切な属性を持っていない場合でも、適切な属性を持つ対象要素のそばにある限り、Silk Test Workbench は確実に識別できます。Silk Test Workbench の XPath ロケータは、UI コントロールを識別するために、UI コントロールの階層全体を使用することもできれば、その一部を使用することもできます。特に最近の AJAX ツールキットは、とても複雑なドキュメント オブジェクト モデル (DOM) を動的に生成するので、UI コントロールを特定するために使用できる適切なコントロール属性を提供しません。

このような場合、インテリジェントなオブジェクト解決の仕組みを提供しないテスト ツールでは、UI コントロールを識別するために、ほとんどの場合インデックス ベースの解決方法を使用することが必要になります。たとえば、展開 アイコンの n 番目のコントロールを識別します。このようなテスト スクリプトは保守することが容易でなく、アプリケーションにほんのわずかな変更を加えるだけでテスト スクリプトが動作しなくなってしまうことが良くあります。

有用な属性を提供しない UI コントロールに対して安定したロケータを作成する良い方法は、階層の中で安定したロケータを持つアンカー要素を見つけることです。そして、そのアンカー要素からロケータを作成したい要素まで辿っていくことができます。

Silk Test Workbench は、この方法を使用してロケータを作成しますが、ときにはコントロールからの安定したロケータを手動で作成することが必要となる場合もあります。

例：コントロールの同列要素の指定

このトピックでは、ロケータに使用することのできる安定した属性を提供しないコントロールを、そのコントロールの同列要素に対する安定したロケータが利用できる場合に、それを使用して指定する方法を説明します。

次の安定したロケータを持つコントロール **Item 0.0** が既に識別されていると想定します。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']
```

Item 0.0 が、タイプ *a* の後ろに同列要素を持っていることがわかっている場合、次のコードを使用して、同列要素の安定したロケータを構築できます。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']/following-sibling::a
```

テキスト フィールドを識別するときにも、この同列要素による方法を使用できます。テキスト フィールドは、たいいていの場合、ロケータで使用できる有用な属性を提供しません。テキスト フィールドのラベルを使用すると、テキスト フィールドの有用なロケータを作成できます。同列要素による方法を使って、テキスト フィールドのロケータの一部として、ラベルを使用することは簡単です。たとえば、テキスト フィールドがテキスト **User Name** を持つラベルの前に同列要素である場合、次のロケータを使用できます。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='User Name']/preceding-sibling::input[@type='text']
```

例：動的 GWT ツリーの展開アイコンの検索

Google Widget Toolkit (GWT) は、とても人気のある強力なツールキットですが、テストしにくいです。動的ツリー コントロールは、とても一般的に使用されている GWT の UI コントロールです。ツリーを展開するには、**展開** アイコン要素を識別する必要があります。

動的 GWT ツリーのサンプルは、<http://samples.gwtproject.org/samples/Showcase/Showcase.html#!CwTree> にあります。

Silk Test Workbench が生成するデフォルトのロケータは次のようになります。

```
/BrowserApplication//BrowserWindow//DIV[@id='gwt-debug-cwTree-dynamicTree-root-child0']/DIV/DIV[1]//IMG[@border='0']
```

次の理由で、このデフォルトのロケータは、**Item 0.0** の **展開** アイコンを識別するロケータとしては信頼できるものではありません。

- ロケータは複雑で、複数の階層から構成されています。AJAX で動的に DOM 構造が少し変わるとロケータは使えなくなります。
- ロケータには、階層のいくつかのコントロールにインデックスが含まれています。インデックスベースのロケータは、その出現番号によってコントロールを検索するため、一般にもろく、たとえば、ツリーの 6 番目の展開アイコンを見つけるなど、うまく特定のコントロールを定義できません。このルールの例外は、たとえばグリッドの 6 番目のデータ行など、識別するさまざまなデータ セットを表すためにインデックスが使用される場合です。

多くの場合、より良いロケータを見つける良い方法は、検索する要素の同列要素を探し出すことです。より良いロケータの同列要素を見つけると、XPath は、これらの同列要素を識別してロケータを構成することができます。この場合、ツリー項目 **Item 0.0** は、**展開** アイコンよりも良いロケータです。ツリー項目 **Item 0.0** のロケータは、コントロールの @textContents プロパティを使用するため、安定した単純なロケータです。

デフォルトでは、Silk Test Workbench は @id プロパティを使用しますが、GWT では @id には 'gwt-uid-<nnn>' のような値 (ここで、<nnn> は同じ要素でも呼出し毎に頻繁に変わります) が含まれるため、たいいてい安定したプロパティではありません。

@textContents プロパティを @id の代わりに使用してロケータを手動で変更できます。

元のロケータ：

```
/BrowserApplication//BrowserWindow//DIV[@id='gwt-uid-109']
```

別のロケータ：

```
/BrowserApplication//BrowserWindow//DIV[@textContents='Item 0.0']
```

もしくは、@id='gwt-uid-<nnn>' を使用しないように Silk Test Workbench を設定できます。この場合、Silk Test Workbench は自動的に安定したロケータを記録します。たとえば、@id プロパティで使用されるテキスト パターンをロケータ属性値除外リストに追加します。この場合、gwt-uid* を除外リストに追加します。

要素の階層を調べると、**Item 0.0** コントロールと **展開** アイコン コントロールは、共通のルートノードとして DomTableRow コントロールを持つことが分かります。

展開 アイコンの安定したロケータを作成するには、次のロケータを使って **Item 0.0** をまず検索する必要があります。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']
```

そして、要素の階層を 2 レベル上がって DomTableRow 要素まで移動します。これは、XPath では、ロケータに `../..` を追加して表現します。最後に、DomTableRow から **展開** アイコンを検索します。**展開** アイコンは、サブツリー内では唯一の IMG コントロールであるので、容易に検索できます。これは、XPath では、ロケータに `//IMG` を追加して表現します。**展開** アイコンの最終的な安定したロケータは次のようになります。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']/../..//IMG
```

XPath の ancestor 軸を使うと、より簡単に **展開** アイコンを指定できます。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']/ancestor::tr//IMG
```

カスタム属性

多くの UI テクノロジーは、UI コントロールのあらかじめ定義された属性のセットをカスタム属性で拡張する方法を提供します。アプリケーション開発者は、コントロールを一意に識別する安定した識別子を導入するためにカスタム属性を使用できます。Silk Test Workbench は、UI コントロールのカスタム属性にアクセスでき、UI コントロールを識別するために、これらのカスタム属性を使用することもできます。

UI コントロールを識別するために特別に自動化用属性を使用すると、caption のような定義済み属性を使用する場合と比較して、いくつかのメリットを享受できます。アプリケーション コードで安定した識別子を指定でき、カスタム属性や定義済みの自動化用プロパティの何れかを通して識別子を公開することで、テスト自動化スクリプトが理解しやすくなり、保守性も高まり、開発プロセスの初期の段階からテストの自動化を開始することができるようになります。

Silk Test Workbench はロケータ生成の柔軟性が高く、識別に使用する属性を設定することができます。


Apache Flex アプリケーションのカスタム属性

Apache Flex アプリケーションは、あらかじめ定義されたプロパティ automationName を使用して、次のように Apache Flex コントロールに対して安定した識別子を指定します。

```
<?xml version="1.0" encoding="utf-8"?>
  <s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" width="400" height="300">
    <fx:Script>
      ...
    </fx:Script>
    <s:Button x="247" y="81" label="Button" id="button1" enabled="true"
      click="button1_clickHandler(event)"
      automationName="AID_buttonRepeat"/>
    <s:Label x="128" y="123" width="315" height="18" id="label1" verticalAlign="middle"
      text="awaiting your click" textAlign="center"/>
  </s:Group>
```

Apache Flex アプリケーションのロケータは次のようになります。

```
...//SparkApplication//SparkButton[@caption='AID_buttonRepeat']
```

 **注目:** Apache Flex アプリケーションの場合、Silk Test Workbench では automationName はロケータ属性 caption に常にマップされます。automationName 属性が指定されていない場合、Silk Test Workbench は属性 ID をロケータ属性 caption にマップします。

Java SWT カスタム属性

カスタム属性をテスト アプリケーションに追加して、テストをより安定させることができます。たとえば、Java SWT では、GUI を実装する開発者が属性 ('silkTestAutomationId' など) をウィジェットに対して定義することによって、アプリケーション内でそのウィジェットを一意に識別することができます。これにより、Silk Test Workbench を使用するテスト担当者は、その属性 (この場合は 'silkTestAutomationId') をカスタム属性のリストに追加すると、その一意の ID によってコントロールを識別できるようになります。カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のウィジェットが追加されると変更されるためです。

複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として 'loginName' を 2 つの異なるテキスト フィールドに割り当てた場合は、'loginName' 属性を呼び出したときに、両方のフィールドが返されます。

Java SWT の例

以下のコードを使用して、テストするアプリケーションにボタンを作成する場合：

```
Button myButton = Button(parent, SWT.NONE);  
myButton.setData("SilkTestAutomationId", "myButtonId");
```

テストの XPath クエリ文字列に属性を追加するには、以下のクエリを使用します。

```
Dim button =  
desktop.PushButton("@SilkTestAutomationId='myButton'")
```

Java SWT アプリケーションをカスタム属性のテストに対して有効化にするには、開発者はカスタム属性をアプリケーションに含める必要があります。属性を含めるには org.swt.widgets.Widget.setData(String key, Object value) メソッドを使用します。

Web アプリケーションのカスタム属性

HTML は、安定した識別子を表すことができる一般的な属性 ID を定義します。定義により、ID は文書内の要素を一意に識別します。特定の ID を持つ要素は文書内で 1 つだけ存在します。

ただし、多くの場合 (特に AJAX アプリケーションでは)、ID は HTML 要素に関連付けられたサーバー ハンドラを動的に識別するために使用されます。つまり、Web 文書の作成のたびに ID は変わるようになります。このような場合、ID は安定した識別子ではなく、Web アプリケーションの UI コントロールを識別するのに適しません。

Web アプリケーションの場合、より確実にするには、Silk Test Workbench に UI コントロールの情報を公開するためだけに使用されるカスタム HTML 属性を新たに導入することです。

カスタム HTML 属性はブラウザーは無視するため、AUT の動作は変わりません。ブラウザーの DOM を通じてアクセスすることができます。Silk Test Workbench では、このような属性を (属性がコントロール クラスのカスタム 属性であっても) 識別時のデフォルト属性として使用するように設定することができます。特定のテクノロジ ドメインのデフォルト識別属性としてカスタム属性を設定するには、**ツール > オプション > 記録** をクリックして、テクノロジ ドメインを選択します。

アプリケーション開発者は、Web 要素にさらに HTML 属性を追加することが必要です。

元の HTML コード：

```
<A HREF="http://abc.com/control=4543772788784322..." <IMG  
src="http://abc.com/xxx.gif" width=16 height=16> </A>
```

新しいカスタム HTML 属性 *AUTOMATION_ID* を持つ HTML コード :

```
<A HREF="http://abc.com/control=4543772788784322..."
AUTOMATION_ID = "AID_Login" <IMG src="http://abc.com/xxx.gif"
width=16 height=16> </A>
```

カスタム属性を設定すると、Silk Test Workbench は、できる限りカスタム属性を使用して、一意のロケーターを構成しようとします。Web ロケーターは次のようになります。

```
...//DomLink[@AUTOMATION_ID='AID_Login']
```

例 : 変化する ID

変化する ID の 1 例は、Google Widget Toolkit (GWT) で、ID は Web 文書の作成のたびに変化する動的な値を保持します :

```
ID = 'gwt-uid-<nnn>'
```

この場合、<nnn> が頻繁に変化します。

Windows Forms アプリケーションのカスタム属性

Windows Forms アプリケーションは、あらかじめ定義された自動化用プロパティ *automationId* を使用して、Windows Forms コントロールに対して安定した識別子を指定します。

Silk Test Workbench は、ロケーターを識別するために、自動的にこのプロパティを使用します。Windows Forms アプリケーションのロケーターは次のようになります。

```
/FormsWindow//PushButton[@automationId='btnBasicControls']
```

WPF アプリケーションのカスタム属性

WPF アプリケーションは、あらかじめ定義された自動化用プロパティ *AutomationProperties.AutomationId* を使用して、次のように WPF コントロールに対して安定した識別子を指定します。

```
<Window x:Class="Test.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="MainWindow" Height="350" Width="525">
  <Grid>
    <Button AutomationProperties.AutomationId="AID_buttonA">The
Button</Button>
  </Grid>
</Window>
```

Silk Test Workbench は、ロケーターを識別するために、自動的にこのプロパティを使用します。WPF アプリケーションのロケーターは次のようになります。

```
/WPFWindow[@caption='MainWindow']//WPFButton[@automationId='AID_buttonA']
```

XPath のパフォーマンス問題のトラブルシューティング

複雑な Web アプリケーションの場合など、オブジェクトの構造が複雑なアプリケーションをテストする場合、パフォーマンスの問題や、スクリプトの信頼性に関連する問題が発生することがあります。このトピックでは、Silk Test Workbench が記録中に自動的に生成したロケーターとは異なるロケーターを使用して、スクリプトのパフォーマンスを向上させる方法について説明します。



注: 一般に、複雑なロケーターを使用することは推奨しません。複雑なロケーターを使用すると、テストの信頼性を損なう恐れがあります。複雑なロケーターは、テスト アプリケーションの構造をほんの少し変更しただけで機能しなくなってしまう可能性があります。それにもかかわらず、スクリプト

のパフォーマンスが要求を満たしていない場合には、より固有のロケータを使用することによってテストのパフォーマンスを向上できる可能性があります。

例として、MyApplication アプリケーションの要素ツリーを以下に示します。

```
Root
  Node id=1
    Leaf id=2
    Leaf id=3
    Leaf id=4
    Leaf id=5
  Node id=6
    Node id=7
      Leaf id=8
      Leaf id=9
    Node id=9
      Leaf id=10
```

以下の最適化手法のいくつかを使用して、スクリプトのパフォーマンスを改善させることができます。

- 複雑なオブジェクト構造内の要素を特定したい場合は、オブジェクト構造全体ではなく、その特定の部分だけを検索するようにします。たとえば、サンプルツリーの識別子 4 を持つ要素を検索する場合に `Root.Find("//Leaf[@id='4']")` というクエリーを使用している場合、`Root.Find("/Node[@id='1']/Leaf[@id='4']")` というクエリーで置き換えます。最初のクエリーでは、識別子 4 を持つリーフが、アプリケーションの要素ツリー全体から検索されます。そして、最初のリーフが見つかった時点で返されます。2 番目のクエリーでは、識別子 1 を持つノードと識別子 6 を持つノードがある最初のレベルのノードがまず検索された後、識別子 4 を持つすべてのリーフが識別子 1 を持つノードのサブツリー内から検索されます。
- 同じ階層内の複数の項目を特定したい場合は、まずは階層を特定してからループ内で項目を特定します。`Root.FindAll("/Node[@id='1']/Leaf")` というクエリーを使用している場合、次のようなループで置き換えます。

```
Public Sub Main()
  Dim node As TestObject

  node = _desktop.Find("//Node[@id='1']")
  For i As Integer = 1 To 4 Step 1
    node.Find("/Leaf[@id='"+i+"']")
  Next
End Sub
```


Silk Test Open Agent

Silk Test Open Agent は、スクリプトのコマンドを GUI 固有のコマンドに翻訳するソフトウェア プロセスです。つまり、Open Agent がテストするアプリケーションを動かす、監視しています。

ホストマシン上で 1 つのエージェントをローカルに実行できます。ネットワーク環境では、任意の数のエージェントがリモート マシン上でテストを再生できます。ただし、記録はローカル マシン上でのみ実行できます。

Silk Test Open Agent の起動

テストの作成またはサンプル スクリプトの実行前に、Silk Test Open Agent が実行されている必要があります。通常は、製品を起動したときにエージェントが実行されます。Open Agent を手動で開始しなければならない場合には、次のステップを実行してください。

(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > ツール > Silk Test Open Agent**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Open Agent** をクリックします。Silk Test Open Agent アイコン  が、システムトレイに表示されます。


テスト実行後に Open Agent を停止する

コマンドライン オプション `-shutDown` を使うか、スクリプトから Open Agent を停止して、テスト実行が終わってもエージェントが実行し続けることがないようにすることができます。次のコマンドをコマンドラインから実行して、Open Agent を停止することができます：

```
openAgent.exe -shutDown
```


スクリプトからエージェントを停止するには：

1. テスト実行が完了したときに実行するスクリプトを開く、または作成します。
たとえば、テスト実行後にクリーンアップのために使用する既存のスクリプトを開きます。
2. スクリプトに ShutDown メソッドを追加します。

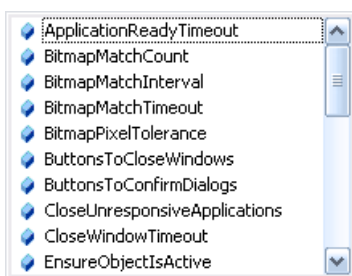
 **注:** 他のスクリプトが Open Agent を必要になった時点で、エージェントは再起動されます。

エージェント オプション

このセクションでは、GetOption および SetOption メソッドで操作できるオプションについて説明します。

 **注:** デフォルトでは、メニューから **ツール > オプション** を選択してグローバル オプションを設定できます。

スクリプトでエージェント オプションを設定または取得するには、GetOption または SetOption メソッドを使用します。たとえば、`Agent.SetOption(Options. と入力すると、オートコンプリートと構文ヒントテクノロジーを使用して、エージェント オプションがアクティブなエディター ウィンドウに自動的に表示されます。たとえば、以下のオプションがアクティブなエディター ウィンドウに表示されます。`



スクリプトでオプションを挿入する場合は、次のように記述します。

```
'VB .NET code
Agent.SetOption(Options.ApplicationReadyTimeout, 100000)
```


あるいは、スクリプトでエージェント オプション名を使用することもできます。たとえば、以下のように入力できます。

```
'VB .NET code
Agent.SetOption("OPT_APPREADY_TIMEOUT", 100000)
```


Open Agent の設定 を使用して、ビジュアル テストにカスタム オプションを追加することができます。**ツール > オプション** に移動し、設定するオプションの種類に依存して **記録** または **再生** を選択し、新しいオプションを名前値ペアとして追加します。この機能は特定の場合にのみ使用すべきです（サポートから提案された場合など）。

.NET オプション名	エージェント オプション名	定数の型	説明
ApplicationReadyTimeout	OPT_APPREADY_TIMEOUT	数値	新しく起動したアプリケーションが準備完了状態になるまで待機する時間をミリ秒で指定します。指定したタイムアウト時間内にアプリケーションの準備が整わない場合は、Silk Test Workbench によって例外がスローされます。
BitmapMatchCount	OPT_BITMAP_MATCH_COUNT	整数	ビットマップが安定していると判断するために、連続して同じビットマップでなければならないスナップショットの数を指定します。スナップショットは、OPT_BITMAP_MATCH_TIMEOUT で指定された秒数まで取得され、各スナップショット間で OPT_BITMAP_MATCH_INTERVAL で指定された時間間隔で一時停止します。 デフォルトで、これは 0 です。
BitmapMatchInterval	OPT_BITMAP_MATCH_INTERVAL	実数	ビットマップ画像が安定していることを保証するために使用するスナップショット間の時間間隔を指定します。スナップショットは、OPT_BITMAP_MATCH_TIMEOUT で指定された時間まで取得されます。 デフォルトで、これは 0.1 です。
BitmapMatchTimeout	OPT_BITMAP_MATCH_TIMEOUT	実数	ビットマップ画像が安定するまでの合計許容時間を指定します。

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>この期間に Silk Test Workbench は画像の複数のスナップショットを取得し、スナップショット間で OPT_BITMAP_MATCH_TIMEOUT で指定された秒数だけ待機します。OPT_BITMAP_MATCH_COUNT で指定されたビットマップ数と一致する前に、OPT_BITMAP_MATCH_TIMEOUT から返される値に到達すると、Silk Test Workbench でスナップショットの取得が停止し、例外 E_BITMAP_NOT_STABLE が発生します。</p> <p>デフォルトで、これは 5 です。</p>
BitmapPixelTolerance	OPT_BITMAP_PIXEL_TOLERANCE	整数	<p>2 つのビットマップが一致しているとみなされる許容差異 (ピクセル単位) を指定します。差異ピクセル数がこのオプションで指定された値より小さい場合、ビットマップは同一であるとみなされます。最大許容値は 32767 ピクセルです。</p> <p>デフォルトで、これは 0 です。</p>
ButtonsToCloseWindows	OPT_CLOSE_WINDOW_BUTTONS	文字列のリスト	<p>CloseSynchron メソッドでウィンドウを閉じるために使用するボタンを指定します。</p>
ButtonsToConfirmDialogs	OPT_CLOSE_CONFIRM_BUTTONS	文字列のリスト	<p>CloseSynchron メソッドでウィンドウを閉じる際に表示された確認ダイアログ ボックスを閉じるために使用するボタンを指定します。</p>
CloseUnresponsiveApplications	OPT_KILL_HANGING_APPS	ブール値	<p>無応答のアプリケーションを閉じるかどうかを指定します。タイムアウトなどの理由で、エージェントとアプリケーション間の通信に失敗した場合に、アプリケーションが無応答になります。複数のインスタンスを実行できないアプリケーションをテストする場合は、このオプションを TRUE に設定します。デフォルトで、これは FALSE です。</p>
CloseWindowTimeout	OPT_CLOSE_WINDOW_TIMEOUT	数値	<p>ウィンドウ閉じるための次の方式を試行する前に待機する時間をミリ秒で指定します。最終的に失敗と判断する前に、4 種類の方式が Agent によって実行されます。つまり、閉じるのに失敗するまでにかかる合計時間は、指定した値の 4 倍の時間になります。</p>
Compatibility	OPT_COMPATIBILITY	文字列	<p>Silk Test Workbench の最新バージョンで特定の機能の動作が変更されている場合は、これらの機能について、指</p>

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>定した Silk Test Workbench バージョンの動作を使用することができます。</p> <p>文字列の例：</p> <ul style="list-style-type: none"> • 12 • 11.1 • 13.0.1 <p>デフォルトでは、このオプションは設定されていません。</p>
EnsureObjectIsActive	OPT_ENSURE_ACTIVE_OBJDEF	ブール値	<p>ターゲット オブジェクトがアクティブであることを保証します。デフォルトで、これは FALSE です。</p>
EnableAccessibility	OPT_ENABLE_ACCESSIBILITY	ブール値	<p>Win32 アプリケーションをテストしているときに、Silk Test Workbench がオブジェクトを認識できない場合、ユーザー補助を有効にするには TRUE に設定します。ユーザー補助は、オブジェクトの認識機能をクラス レベルで強化するためのものです。</p> <p>ユーザー補助を無効にするには FALSE に設定します。</p> <p>デフォルトで、これは FALSE です。</p> <p> 注: Mozilla Firefox と Google Chrome の場合、ユーザー補助は常に有効で、無効にすることはできません。</p>
EnableMobileWebviewFallbackSupport	OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT	ブール値	<p>デフォルトのブラウザ サポートではテストできないハイブリッド モバイル アプリケーションに対して、モバイル ネイティブ フォールバックのサポートを有効化します。</p> <p>デフォルトで、これは FALSE です。</p>
EnableUiAutomationSupport	OPT_ENABLE_UI_AUTOMATION_SUPPORT	数値	<p>通常の Win32 コントロールの認識の代わりに Microsoft UI オートメーション サポートを有効化する場合は TRUE を設定します。Win32 アプリケーションをテストしているときに、Silk Test Workbench がオブジェクトを認識できない場合、このオプションは有用です。AUTODETECT を指定すると、JavaFX などの既存のテクノロジーに対して Microsoft UI オートメーションが自動的に有効になります。</p> <p>デフォルトで、これは FALSE です。</p>
HangAppTimeOut	OPT_HANG_APP_TIME_OUT	数値	<p>応答のない再生操作をキャンセルするまでのタイムアウト値を、無応答のア</p>

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>アプリケーションのタイムアウト で指定します。</p> <p>デフォルト値は、5000 ミリ秒です。</p>
HighlightObjectDuringPlayback	OPT_REPLAY_HIGHLIGHT	ブール値	<p>現在のオブジェクトが再生中にハイライトされるかどうかを指定します。</p> <p>デフォルトで、これは FALSE です。つまり、デフォルトではオブジェクトはハイライトされません。</p>
KeyboardEventDelay	OPT_KEYBOARD_INPUT_DELAY	数値	<p>再生時のキー入力間の遅延をミリ秒で指定します。</p> <p>テストするアプリケーションに応じて、選択する最適な値が異なることに注意してください。たとえば、Web アプリケーションをテストする場合、1 ミリ秒に設定すると、ブラウザが極端に遅くなります。ただし、この値を 0 (ゼロ) に設定すると、基本的なアプリケーション テストに失敗する可能性があります。</p>
KeysToCloseDialogs	OPT_CLOSE_DIALOG_KEYS	文字列のリスト	<p>CloseSynchron メソッドでウィンドウを閉じる際に表示されたダイアログ ボックスを閉じるキー シーケンスを指定します。例：<ESC>、<Alt+F4>。</p>
LocatorAttributesCaseSensitive	OPT_LOCATOR_ATTRIBUTES_CASE_SENSITIVE	ブール値	<p>はいに設定すると、ロケータ属性名の大文字と小文字が区別されるようになり、いいえに設定すると、ロケータ名の大文字と小文字が区別されなくなります。モバイル Web アプリケーションのロケータ属性の名前は、常に大文字と小文字の区別はされません。つまり、モバイル Web アプリケーションの記録や再生時に、このオプションは無視されます。</p>
MenuItemsToCloseWindows	OPT_CLOSE_WINDOW_MENUS	文字列のリスト	<p>CloseSynchron メソッドでウィンドウを閉じるために使用するメニュー項目を指定します。例：「ファイル/*終了」、「ファイル/*中止」。</p>
MouseEventDelay	OPT_MOUSE_INPUT_DELAY	数値	<p>各マウス イベントの前に使用される遅延をミリ秒で指定します。</p>
ObjectResolveRetryInterval	OPT_WAIT_RESOLVE_OBJDEF_RETRY	数値	<p>再生中にオブジェクトが解決できなかった場合に、再試行する前に待機する時間をミリ秒で指定します。</p> <p>ObjectResolveTimeout の値を超過した場合は、再試行は行われません。</p>
ObjectResolveTimeout	OPT_WAIT_RESOLVE_OBJDEF	数値	<p>再生中にオブジェクトの解決を待機する時間をミリ秒で指定します。オブジェクトが解決されるとすぐに、Silk</p>

.NET オプション名	エージェント オプション名	定数の型	説明
PlaybackMode	OPT_REPLAY_MODE	数値	<p>Test Workbench はこのオブジェクトを認識することができます。</p> <p>コントロールの再生方法を定義します。「低レベル」を使用すると、マウスとキーボードを使用して各コントロールが再生されます。「高レベル」を使用すると、API を使用して各コントロールが再生されます。コントロールごとにデフォルトの再生モードが割り当てられています。デフォルトの再生モードが選択されると、各コントロールごとのデフォルトの再生モードが使用されます。デフォルトモードを使用すると、最も信頼できる結果が得られます。「低レベル」または「高レベル」の再生を選択すると、すべてのコントロールの再生モードが選択した再生モードで上書きされます。</p> <p>有効な値は、0、1、2 です。0 はデフォルト、1 は高レベル、2 は低レベルです。デフォルトで、これは 0 です。</p>
PostReplayDelay	OPT_POST_REPLAY_DELAY	数値	<p>関数の呼び出しまたはプロパティの設定後に待機する時間をミリ秒で指定します。</p>
RemoveFocusOnCaptureText	OPT_REMOVE_FOCUS_ON_CAPTURE_TEXT	ブール値	<p>はいに設定すると、テキストのキャプチャ中にテスト対象アプリケーションからフォーカスが外れます。デフォルトではいいえに設定されており、テスト対象アプリケーションにフォーカスが残ります。テキストのキャプチャは、次のメソッドによる記録および再生中に実行されます。</p> <ul style="list-style-type: none"> • TextClick • TextCapture • TextExists • TextRect
SuspensionTimeout	OPT_NTF_SUSPENSION_TIMEOUT	数値	<p>NTF から回答を得るまで待機する最大時間をミリ秒で指定します。この時間を超えた場合は、NTF と Open Agent 間の通信が中断されたものとみなします。</p>
SyncTimeout	OPT_SYNC_TIMEOUT	数値	<p>オブジェクトが準備完了状態になる最大時間をミリ秒で指定します。</p> <p> 注: Silk Test 13.0 より前のバージョンの Silk Test からアップグレードする場合、OPT_XBROWSER_SYNC_TIMEOUT オプションが設定されていると、オブ</p>

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>シオン ダイアログ ボックスに OPT_SYNC_TIMEOUT のデフォルト値が表示されますが、タイムアウトは定義した値に設定されています。</p>
TransparentClasses	OPT_TRANSPARENT_CLASSES	文字列のリスト	<p>オブジェクト階層を単純化し、テストスクリプトや関数のコードの行の長さを短くするために、次のテクノロジーの確実に不要なクラスに対するコントロールを抑制できます。</p> <ul style="list-style-type: none"> • Win32 • Java AWT/Swing • Java SWT/Eclipse • Windows Presentation Foundation (WPF) <p>記録や再生中に無視したいクラスの名前を指定します。</p>
	OPT_WPF_CHECK_DISPATCHER_FOR_IDLE	ブール値	<p>WPF アプリケーションによっては、コントロールの実装方法が影響して、Silk Test 同期が機能しない場合があります。これにより、WPF アプリケーションがアイドルであることを Silk Test Workbench が認識できなくなります。このオプションを FALSE に設定すると、WPF 同期が無効になり、WPF アプリケーションを制御するスレッドである WPF ディスパッチャーを Silk Test Workbench が確認しないようになります。このオプションを FALSE に設定すると、一部の WPF アプリケーションで発生する同期問題を解決できます。デフォルトで、これは TRUE です。</p>
WPFCustomClasses	OPT_WPF_CUSTOM_CLASSES	文字列のリスト	<p>記録や再生の対象にしたい WPF クラスの名前を指定します。たとえば、MyGrid というカスタム クラスが WPF Grid クラスから継承された場合、MyGrid カスタム クラスのオブジェクトは記録や再生に使用できません。Grid クラスはレイアウト目的のためにのみ存在し、機能テストとは無関係であるため、Grid オブジェクトは記録や再生に使用できません。この結果、Grid オブジェクトはデフォルトでは公開されません。機能テストに無関係なクラスに基づいたカスタム クラスを使用するには、カスタム クラス (この場合は MyGrid) を OPT_WPF_CUSTOM_CLASSES オプションに追加します。これによ</p>

.NET オプション名	エージェント オプション名	定数の型	説明
WPFPrefillItems	OPT_WPF_PREFILL_ITEMS	ブール値	<p>て、記録、再生、検索、プロパティの検証など、すべてのサポートされる操作を指定したクラスに対して実行できるようにになります。</p> <p>記録および再生中に、WPFComboBox や WPFListBox のような WPFItemsControl 内の項目を事前に入力するかどうかを定義します。WPF 自体が特定のコントロールの項目を遅延読み込みするため、項目がビューにスクロールされない場合、それらの項目は Silk Test Workbench では使用できません。ビューにスクロールされないアクセスできない項目にアクセスするには、事前入力をオンにします。これはデフォルトの設定です。ただし、一部のアプリケーションでは Silk Test Workbench によってバックグラウンドで項目が事前入力されると問題が発生し、そのためアプリケーションがクラッシュすることがあります。この場合、事前入力をオフにします。</p>
XbrowserEnableIframeSupport	OPT_XBROWSER_ENABLE_IFRAME_SUPPORT	ブール値	<p>ブラウザの iframe および frame のサポートを有効にするかどうかを指定します。Web アプリケーションの iframe の内容を無視して良い場合、iframe のサポートを無効にすると再生速度が速くなる可能性があります。たとえば、多くの広告が表示される Web ページやモバイル ブラウザーでテストする際に iframe サポートを無効にすると、再生パフォーマンスが大幅に向上する可能性があります。このオプションは、Internet Explorer では無視されます。このオプションは、デフォルトで有効になっています。</p>
XBrowserExcludeIFrames	OPT_XBROWSER_EXCLUDE_IFRAMES	文字列	<p>リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象に含まれます。ワイルドカードを使用できます。たとえば、<code>"src:*advertising*"</code> という項目を定義すると、<code><IFRAME src=http://my.domain/advertising-banner.html></code> がテスト対象から除外されます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。</p>

.NET オプション名	エージェント オプション名	定数の型	説明
XBrowserFindHiddenInputFields	OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS	ブール値	非表示入力フィールドを表示するかどうかを指定します (タグに type="hidden" を指定した HTML フィールドです)。デフォルト値は、TRUE です。
XBrowserIncludeIFrames	OPT_XBROWSER_INCLUDE_IFRAMES	文字列	リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象から除外されます。ワイルドカードを使用できます。たとえば、"name:*form" という項目を定義すると、<IFRAME name="user-form" src=...> がテスト対象に含まれます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。
XBrowserSynchronizationMode	OPT_XBROWSER_SYNC_MODE	文字列	サポートする同期モード (HTML または AJAX) を設定します。HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれるより複雑なシナリオが使用される場合は、WaitForObject、WaitForProperty、WaitForDisappearance、または WaitForChildDisappearance などの同期関数を使用して手動でスクリプトを記述することが必要になる可能性があります。AJAX モードを使用すると、同期関数を手動で記述する必要はなくなります。デフォルト値は、 AJAX に設定されています。
XBrowserSynchronizationTimeout	OPT_XBROWSER_SYNC_TIMEOUT	数値	オブジェクトが準備完了状態になる最大時間をミリ秒で指定します。  注: 推奨されません。代わりに OPT_SYNC_TIMEOUT オプションを使用してください。
XBrowserSynchronizationURLExcludes	OPT_XBROWSER_SYNC_EXCLUDE_URLS	文字列	ページ同期中に除外するサービスまたは Web ページの URL を指定します。AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を 同期除外リスト 設定で指定します。</p> <p><code>http://test.com/timeService</code> のように URL 全体を入力したり、<code>timeService</code> のように URL の一部を入力します。</p> <p>次のようにカンマで項目を区切ります。</p> <pre>'VB .NET code Agent.SetOption(OPT_XBROWS ER_SYNC_EXCLUDE_URLS, { "fpdownload.macromedia.co m", "fpdownload.adobe.com", "download.microsoft.com" })</pre>

Silk Test Workbench コンポーネント間の接続設定

ファイアウォールを介してリモートマシンに接続したり、HTTPS を使って安全にリモートマシンに接続するために、Silk Test Workbench が Information Service や Open Agent との通信に使用するポートを設定できます。

Open Agent が起動すると、Silk Test Workbench およびテストするアプリケーションに対して、使用可能なポートがランダムに割り当てられます。ポート番号は *Silk Test Information Service* に登録されます。

Information Service は次の情報を Silk Test Workbench に提供します。

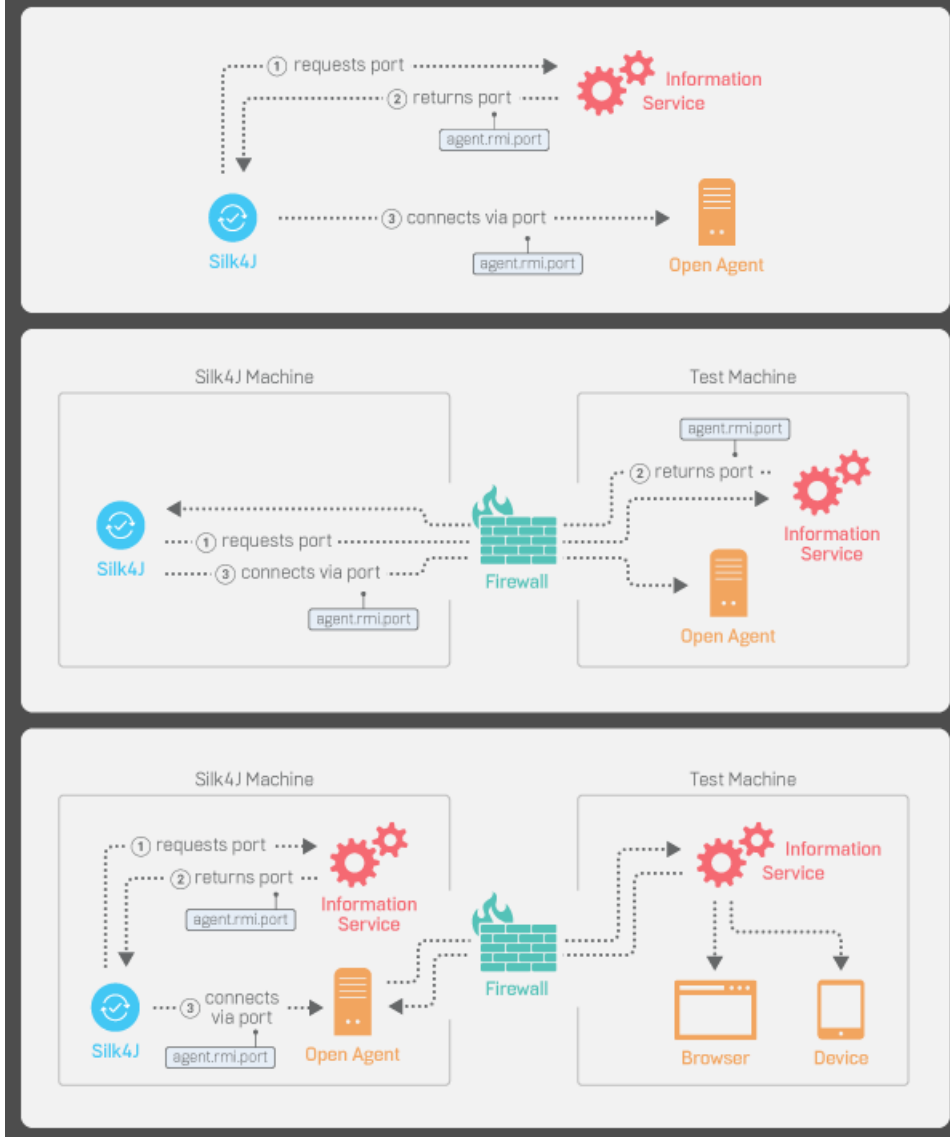
- Silk Test Workbench が Open Agent との接続に使用できるポート番号。通信は、Silk Test Workbench とエージェントとの間で直接行われます。リモートエージェントを使用するシナリオでは、たとえばファイアウォールによって遮断されないためなど、このポートを設定する必要がある場合があります。
- Information Service がインストールされているマシン上で利用可能なブラウザ。
- Information Service がインストールされているマシンに接続しているモバイルデバイス。
- Information Service がインストールされているマシン上で利用可能なエミュレータ。
- 上記のモバイルデバイスやエミュレータ上で利用可能なモバイルブラウザ。

デフォルトでは、ポート 22901 を使用して Open Agent は Information Service と通信します。デフォルトポートが利用可能でない場合に機能する代替ポートとして、Information Service の追加のポートを構成できます。デフォルトでは、Information Service は、代替ポートとして 2966、11998、および 11999 を使用します。

大抵の場合、手動でポート番号を設定する必要はありません。ただし、リモートマシン上でテストを実行する場合、ポート番号が競合したり、Silk Test Workbench をインストールしたマシンとテストマシン間でファイアウォールによって遮断される場合など、Silk Test Workbench とリモートマシン上の Information Service 間の通信に使用するポート番号を設定できます。テストするリモートマシンが複数ある場合は、リモートマシンごとに異なるポート番号を使用することも、すべてのリモートマシンで同じポート番号を使用することもできます。

次の図に、Silk Test Workbench、Information Service、Open Agent 間での通信を示します。

Configuring the Silk4J/Open Agent Connection



Open Agent に接続するポートの構成

このタスクを開始する前に、Silk Test Open Agent を停止します。

大抵の場合、手動でポート番号を設定する必要はありません。Information Service はポート構成を自動的に処理します。Open Agent との接続には、Information Service のデフォルトのポートを使用します。これにより、Information Service によって、Open Agent が使用するポートに通信が転送されます。ただし、ファイアウォールによって遮断されないようにする場合は、特定のポートに固定して通信するように Information Service ポートを設定できます。


必要に応じて、Silk Test クライアントまたはテストするアプリケーションが Open Agent に接続するために使用するポート番号を変更します。

1. agent.properties.sample ファイルに移動し、開きます。

デフォルトでは、このファイルは次の場所にあります：`%APPDATA%\Silk\SilkTest\conf`。大抵の場合、`C:\Users<ユーザー名>\AppData\Silk\SilkTest\conf` になります。ここで、<ユーザー名> は、現在のユーザー名に一致します。

2. 代替ポートの値を変更します。


大抵の場合、ポートの競合を解決するためにポートの設定を構成します。

 **注:** 各ポート番号は一意でなければなりません。エージェントのポート番号が Information Service のポート設定とは異なることを確認してください。

ポート番号は、1 から 65535 の間の任意の数値を指定できます。

ポートの設定には次のものがあります：

- agent.vtadapter.port：テストの実行時に、Silk Test Workbench と Open Agent 間の通信を制御します。
- agent.xpmodule.port：テストの実行時に、Silk Test Classic とエージェント間の通信を制御します。
- agent.autcommunication.port：Open Agent とテストするアプリケーション間の通信を制御します。
- agent.rmi.port：Open Agent と Silk4J 間の通信を制御します。
- agent.ntfadapter.port：Open Agent と Silk4NET 間の通信を制御します。
- agent.heartbeat.port：リモートマシンにインストールした Open Agent を使ってテストする場合に必要です。

 **注:** Apache Flex のテスト時に使用されるポートは、この構成ファイルでは制御できません。Flex アプリケーションのテストで割り当てられるポート番号は、6000 から始まり、各 Flex アプリケーションがテストされる度に 1 ずつ増加していきます。Flex テスト用に開始ポートを構成することはできません。

3. ファイルを agent.properties という名前で保存します。

4. Open Agent、Silk Test クライアント、およびテストするアプリケーションを再起動します。

Information Service に接続するポートの構成

このタスクを開始する前に、Silk Test Open Agent を停止します。

大抵の場合、手動でポート番号を設定する必要はありません。Information Service はポート構成を自動的に処理します。Open Agent との接続には、Information Service のデフォルトのポートを使用します。これにより、Information Service によって、Open Agent が使用するポートに通信が転送されます。ただし、ファイアウォールによって遮断されないようにする場合は、特定のポートに固定して通信するように Information Service ポートを設定できます。

デフォルトでは、Silk Test Workbench と Information Service との接続に使用されるポートは、セキュアな HTTPS 接続のポートで 48561 です。デフォルトのポートが使用可能であれば、ポート番号を指定せずに単純に hostname だけを入力できます。ポート番号を指定する場合には、Information Service のデフォルトのポートまたは追加したポートの 1 つと一致していることを確認してください。間違ったポートが指定されていると、通信に失敗します。

必要に応じて、Information Service に接続するためにすべてのクライアントが使用するポート番号を変更できます。

1. infoservice.properties.sample ファイルに移動し、開きます。

- Microsoft Windows システムでは、このファイルは C:¥ProgramData¥Silk¥Silk Test¥conf にあります。ここで、「C:¥ProgramData」は、Windows システムでデフォルトで設定されている環境変数 ALLUSERSPROFILE の値です。
- macOS では、このファイルは /Users/<ユーザー>/.silk/silktest/conf にあります。

このファイルには、コメントとサンプルの代替ポート設定が含まれています。

2. Silk Test Workbench と Information Service との通信に、HTTPS によるセキュアな接続を使用するかどうかを指定します。

- HTTPS によるセキュアな通信を使用する場合は、infoservice.https.enabled を true に設定します。これは、デフォルトの設定です。

- HTTPS によるセキュアな通信を使用しない場合は、`infoservice.https.enabled` を `false` に設定します。
3. 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.https.port` を設定します。
デフォルトの HTTPS ポートは 48561 です。ポート番号は、1 から 65535 の間の任意の数値を指定できます。
 4. 省略可能 : HTTPS によるセキュアな通信を使用しないように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.port` を設定します。
デフォルトのポートは 22901 です。
 5. ファイルを `infoservice.properties` という名前で保存します。
 6. Open Agent、Silk Test クライアント、およびテストするアプリケーションを再起動します。

Silk Test Information Service プロパティの編集

`infoservice.properties` ファイルを使用して、Silk Test Information Service のポートや、HTTPS によるセキュアな接続を使用するかどうかを指定したり、Silk Test Information Service を実行しているマシン上で Silk Test がテストを実行するたびに毎回適用されるケイパビリティ (Capabilities) を指定できます。

1. `infoservice.properties.sample` ファイルがあるディレクトリに移動します。
 - Windows マシン上で、`%PROGRAMDATA%\Silk\SilkTest\conf` に移動します (C:\ProgramData\Silk\SilkTest\conf など)。
 - macOS 上で、`~/silk/silktest/conf/` に移動します。
2. `infoservice.properties.sample` ファイルの名前を `infoservice.properties` に変更します。
3. Silk Test Workbench と Information Service との通信に、HTTPS によるセキュアな接続を使用するかどうかを指定します。
 - HTTPS によるセキュアな通信を使用する場合は、`infoservice.https.enabled` を `true` に設定します。これは、デフォルトの設定です。
 - HTTPS によるセキュアな通信を使用しない場合は、`infoservice.https.enabled` を `false` に設定します。
4. 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.https.port` を設定します。
デフォルトの HTTPS ポートは 48561 です。ポート番号は、1 から 65535 の間の任意の数値を指定できます。
5. 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合に、すべての HTTP リクエストを HTTPS ポートにリダイレクトするには、`infoservice.http-to-https.enabled` を `true` に設定します
デフォルト値は、`false` です。
6. 省略可能 : HTTPS によるセキュアな通信を使用しないように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.port` を設定します。
デフォルトのポートは 22901 です。
7. 省略可能 : Silk Test が HTTPS 接続で使用する証明書をユーザーの証明書で置き換える場合は、[「Information Service との HTTPS 接続に使用する証明書の置換」](#)を参照してください。
8. ケイパビリティを指定するには、次の行を `infoservice.properties` ファイルに追加します。
`customCapabilities=<custom_capability_1>;<custom_capability_2>;...`

例：指定した言語での iOS シミュレータの実行

Mac 上で iOS シミュレータを常に同じ言語で実行するには、カスタム ケイパビリティ *language* および *locale* を指定します。たとえば日本語の場合、次の行を `infoservice.properties` ファイルに追加します。

```
customCapabilities=language=ja;locale=ja_JP
```

Information Service との HTTPS 接続に使用する証明書の置換

Silk Test Workbench と Information Service 間でセキュアな HTTPS 接続を使用する場合、次の自己署名証明書ファイルが使用されます。

- keystore 証明書ファイルは、Information Service HTTPS サーバーで使用されます。
- 次の証明書ファイルは、Silk Test Workbench クライアントを実行するマシンで使用されます。
 - cacerts
 - cacerts.p12
 - cacerts.pem

OpenSSL と Java keytool 実行可能ファイルを使って、これらのファイルをユーザー独自の証明書ファイルで置き換えることができます。

1. 使用するマシン上に OpenSSL と JDK がインストールされていることを確認してください。
2. JDK インストール フォルダの bin フォルダにある Java keytool 実行可能ファイルを実行します。
3. Information Service HTTPS サーバー上で、秘密キーと公開キーのペアを秘密 keystore ファイルに作成します。

```
keytool -genkey -alias jetty -keyalg RSA -keypass Borland -storepass Borland -keystore keystore -validity 1095
```

4. プロンプトの指示に従い、氏名を入力し、ホストにはワイルドカード * を入力します。
5. 秘密 keystore ファイルから、server.cer という名前のテンポラリ証明書ファイルに情報をエクスポートします。

```
keytool -export -alias jetty -storepass Borland -file server.cer -keystore keystore
```

このテンポラリ証明書ファイルは、Silk Test Workbench クライアントを実行するマシン用の証明書ファイルを生成するために必要になります。

6. server.cer ファイルから、cacerts という名前の証明書ファイルを作成します。

```
keytool -import -v -trustcacerts -alias jetty -file server.cer -keystore cacerts -keypass Borland -storepass Borland
```

7. cacerts ファイルから、cacerts.p12 という名前のテンポラリ証明書ファイルに情報をインポートします。

```
keytool -importkeystore -srckeystore cacerts -destkeystore cacerts.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass Borland -deststorepass Borland
```

8. テンポラリ証明書ファイル cacerts.p12 から、PKCS12 キーストア形式の公開キーストア ファイル cacerts.pem を作成します。


```
openssl pkcs12 -in cacerts.p12 -out cacerts.pem -clcerts -nokeys
```

9. ファイル (keystore、cacerts、cacerts.p12、cacerts.pem) を設定フォルダにコピーします。

- Windows マシンの場合、`%PROGRAMDATA%\Silk\SilkTest\conf` にファイルをコピーします (C: \ProgramData\Silk\SilkTest\conf など)。
- macOS の場合、`~/silk/silktest/conf/` にファイルをコピーします。

Open Agent を使用したリモート テスト

リモート マシン上に Silk Test Workbench をインストールして、ローカル マシンにインストールされている Silk Test Workbench からリモート ロケーション上のアプリケーションをテストすることができます。

 **注:** リモート マシンに接続されているモバイル デバイスや、リモート マシン上のエミュレータまたはシミュレータ上のモバイル アプリケーションをテストする場合や、Apple Safari またはリモート Microsoft Edge 上の Web アプリケーションをテストする場合には、リモート Open Agent ではなく、リモート Silk Test Information Service を使用する必要があります。

リモート Open Agent を使用したテスト

Silk Test Workbench を使用してリモート マシン上のアプリケーションに対してテストを再生するには、次のタスクを実行します。

1. ローカル マシン上のアプリケーションに対してテストを作成します。
2. リモート マシン上に Open Agent をインストールします。
詳細については、『[Silk Test インストール ガイド](#)』を参照してください。
3. リモート マシン上で Open Agent を開始します。

オブジェクト マップ

オブジェクト マップはテスト資産の一種であり、コントロールまたはウィンドウのロケーターではなく、コントロールまたはウィンドウに論理名 (エイリアス) を関連付ける項目が含まれています。コントロールがオブジェクト マップ資産に登録されると、ビジュアルテストおよびスクリプトでのそのコントロールに対する参照はすべて、実際のロケーター名ではなく、そのエイリアスによって行われます。

複数のスクリプトで頻繁に使用するオブジェクトを格納するために、オブジェクト マップを使用できます。複数のテストで 1 つのオブジェクト マップ項目の定義を参照できるため、ユーザーがそのオブジェクト マップ定義を 1 回更新すると、オブジェクト マップ定義を参照するすべてのテストでそのオブジェクト マップ定義が Silk Test Workbench によって更新されます。

スクリプトで、オブジェクト マップ識別子とロケーターを混在させることができます。この機能により、オブジェクト マップを比較的小さいまま保ち、管理しやすくすることが可能です。共通で使用されるオブジェクトをオブジェクト マップに格納し、まれにしかしようないオブジェクトを参照するにはロケーターを使用します。



ヒント: オブジェクト マップが提供する機能を最適に使用するには、テストしたいアプリケーションごとに個々のプロジェクトを Silk Test Workbench に作成します。

オブジェクト マップの例

以下の構成では、ロケーターが使用されている `BrowserWindow` の定義が示されています。

```
_desktop.BrowserApplication("cnn_com").BrowserWindow("//  
BrowserWindow[1]")
```

オブジェクト マップ資産の名前は `cnn_com` です。オブジェクト マップのエイリアスによって置き換えることができるロケーターは、以下のとおりです。

```
"//BrowserWindow[1]"
```

`BrowserWindow` のオブジェクト マップ エントリは `BrowserWindow` です。

結果的に、スクリプト内の `BrowserWindow` の定義は以下のようになります。

```
_desktop.BrowserApplication("cnn_com").BrowserWindow("BrowserWindow")
```

ロケーターのインデックスが変更された場合、テスト スクリプトのロケーターのすべての外観を変更する必要はなく、オブジェクト マップのエイリアスを変更するだけで済みます。Silk Test Workbench によって、オブジェクト マップ定義を参照するすべてのテストが更新されます。

オブジェクト マップ識別子とロケーターを混在させる例

つぎのサンプルコードは、オブジェクト マップ識別子と、オブジェクト マップに格納されたオブジェクトのまれに使用される子オブジェクトを指定するロケーターを混在させる方法を示します：

```
// VB  
Window window = _desktop.Window("MyApplication") // object map id -  
the application window is used often  
MenuItem aboutMenuItem = _desktop.MenuItem("@caption='About'") //  
locator - the About dialog is only used once  
aboutMenuItem.Select()
```

オブジェクト マップを使用する利点

オブジェクト マップには、以下の利点があります。


- オブジェクト マップ項目のロケーターに加えられた変更を、対応するオブジェクト マップ項目を含むすべてのテストに適用することによって、テストのメンテナンスが簡単になる。
- 大規模な機能テスト環境において、ロケーターの扱いが容易になる。
- 個々のスクリプトから独立して管理することができるようになる。
- 複雑なロケーター名がわかりやすい名前でも置き換えられるため、ビジュアル テストとスクリプトが読みやすくなる。
- テスト アプリケーションが変更された場合に変わる可能性のあるロケーターに依存しなくなる。


オブジェクト マップのオン/オフの切り替え

記録時に Silk Test Workbench でロケーター名またはオブジェクト マップのエイリアスのいずれを使用するかを設定できます。

記録中にオブジェクト マップからエイリアスを使用するには、以下を実行します。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。記録 オプションが右側のパネルに表示されます。
3. ロケーター をクリックします。
4. オブジェクト マップ エントリを記録するか、XPath ロケーターを記録するかを定義するには、**オブジェクト マップ エントリを記録する** リストから適切な記録モードを選択します。
 - **オブジェクト マップ エントリ (新しいオブジェクトと既存のオブジェクト)**。これはデフォルトのモードです。
 - **XPath ロケーター (新しいオブジェクトと既存のオブジェクト)**。
 - **XPath ロケーター (新しいオブジェクトのみ)**。オブジェクト マップに既に存在するオブジェクトに対しては、オブジェクト マップ エントリが再利用されます。この設定を選択すると、AUT のメイン コントロールに対するオブジェクト マップを作成し、AUT に対して追加のテストを作成する間、これらのオブジェクト マップを保持することができます。

 **注:** XPath 属性のほかに、ロケーターの記録中にオブジェクト マップをマージする際に、Silk Test Workbench は要素の追加の属性を使用します。ただし、記録したスクリプトでオブジェクト マップ ID の用法を曖昧にする可能性のある属性は既存のオブジェクト マップ エントリにロケーターをマップするために使用されません。

 **注:** **オブジェクト マップを記録する** 設定を有効にすると、Silk Test Workbench 全体にわたって、ロケーター名の代わりにオブジェクト マップの項目名が表示されます。たとえば、**プロパティ** ペインで **アプリケーション構成** カテゴリを表示する場合、ロケーター名ではなくオブジェクト マップ項目名が **ロケーター** ボックスに表示されます。

オブジェクト マップのスコープ

追加のビジュアル テストまたはスクリプトを記録する場合、Silk Test Workbench は現在のプロジェクトを検索して、記録時に使用する一致したオブジェクト マップ項目が決まります。一致する項目が見つからない場合、Silk Test Workbench はすべての参照プロジェクトを検索します。一致する項目が参照プロジェクトで見つかり、その項目が使用されます。一致する項目が参照プロジェクトで見つからない場合、Silk Test Workbench によって、現在のプロジェクトに新しい項目が作成されます。



注: 一致するアイテムが参照プロジェクトに存在する場合、このルールの例外処理が実行されます。新しいオブジェクトが識別され、そのオブジェクトが参照プロジェクトのオブジェクト マップの子オブジェクトである場合は、参照プロジェクトのオブジェクト マップに追加されます。

オブジェクト マップで現在のプロジェクトのみを使用するには、**オプション** ダイアログ ボックスの **オブジェクト マップの場所** オプションを **現在のプロジェクト** に設定します。このように設定すると、一致するオブジェクト マップ項目の検索および新しい項目の記録で、現在のプロジェクトのみが使用されます。

操作の記録中でのオブジェクト マップのマージ

Silk Test Workbench を使用して操作を記録するときに、Silk Test Workbench は、既存のオブジェクト マップ エントリが再利用できるかどうか確認します。Silk Test Workbench は、新しいロケーターが生成されるときに、記録中に直接確認します。Silk Test Workbench は、テスト対象アプリケーションで現在記録されているオブジェクトが既存のオブジェクト マップ エントリと完全に一致するかどうか確認し、一致する場合に Silk Test Workbench はオブジェクト マップからそのオブジェクト マップ識別子を再利用します。

この動作には以下のような利点があります。

- オブジェクト マップのロケーターが変更された場合でも、Silk Test Workbench は記録中にオブジェクト マップ識別子を正しく再利用します。
- 記録したスクリプトに間違ったオブジェクト マップ識別子を含むはずがないため、間違ったオブジェクト マップ識別子によって再生に失敗することは決してありません。
- 階層のレベルをさらに追加した場合など、オブジェクト マップを再構成した場合でも、オブジェクト マップ識別子を再利用することができます。

例

Micro Focus Web サイト (<http://www.borland.com>) の **Products** リンクをクリックしたとき、Silk Test Workbench は、次のスクリプトを記録します。

```
With _desktop.BrowserApplication( "borland_com" )
  With .BrowserWindow( "BrowserWindow" )
    .DomLink( "Products" ).Click( MouseButton .Left, New Point (47, 18))
  End With
End With
```

記録したオブジェクト マップは次のようになります。

```
borland_com //BrowserApplication
  BrowserWindow //BrowserWindow
  Products //
A[@textContents='Products']
```

ここで、Micro Focus Web サイトのヘッダー部分を含むようにオブジェクト マップを手動で再構成した場合を考えます。

```
borland_com //BrowserApplication
  BrowserWindow //BrowserWindow
  header //
  HEADER[@role='banner']
  Products //
A[@textContents='Products']
```

Products リンクのクリックを記録すると、オブジェクト マップが正しく再利用され、次のスクリプトが記録されます。

```
With _desktop.BrowserApplication( "borland_com" )
  With .BrowserWindow( "BrowserWindow" )
    .DomElement("header").DomLink( "Products" ).Click( MouseButton .Left
, New Point (47, 18))
```

End With
End With



注: About リンクなどの Micro Focus Web サイトのヘッダー部分にあるほかのオブジェクトを記録すると、Silk Test Workbench は **header** ではなく、**BrowserWindow** の子として **About** オブジェクト マップ エントリを追加します。

Web アプリケーションでのオブジェクト マップの使用

デフォルトで、Web アプリケーションに対する操作を記録すると、Silk Test Workbench は、*Common* プロジェクトで、ネイティブ ブラウザのコントロール用に *WebBrowser* という名前のオブジェクト マップを作成し、各 Web ドメイン用にオブジェクト マップ資産を作成します。

記録 オプションの **全般** セクションで、**オブジェクト マップの場所** を現在のプロジェクトに変更できます。その後、Silk Test Workbench は記録中に Web ドメインごとにオブジェクト マップ資産を生成します。

印刷または設定用のメイン ウィンドウやダイアログ ボックスなど、Web ドメインに特有ではない共通のブラウザ コントロールの場合、*WebBrowser* という名前を使用して、現在のプロジェクトに追加のオブジェクト マップが生成されます。生成されたオブジェクト マップ資産は、**アセット ブラウザ**で確認できません。

オブジェクト マップで、オブジェクト マップのエントリのグループ化に使用される URL パターンを編集できます。パターンを編集すると、Silk Test Workbench はそのパターンの構文検証を行います。パターンには、ワイルドカード * および ? を使用できます。

例

<http://www.borland.com> および <http://www.microfocus.co.jp> で何らかの操作を記録した後、プリンタ ダイアログを開くと、次の 3 つの新しいオブジェクト マップ資産が**アセット ブラウザ**に追加されます。

- WebBrowser
- borland_com
- microfocus_com



注: Silk Test Workbench では、オブジェクト マップのないプロジェクトに対してのみ、新しいオブジェクト マップ資産が生成されます。バージョン 14.0 よりも前のバージョンの Silk Test Workbench を使用して生成されたオブジェクト マップをすでに含む Silk Test Workbench の Web アプリケーションに対する操作を記録すると、追加で記録されたエントリは既存のオブジェクト マップに保存されます。Web ドメインに対して追加のオブジェクト マップ資産が生成されることはありません。

オブジェクト マップ項目名の変更

オブジェクト マップでは、項目とロケーターの名前を手動で変更できます。



警告: オブジェクト マップ項目の名前を変更すると、その項目を使用するすべてのビジュアル テストまたはスクリプトが影響を受けます。たとえば、**キャンセル** ボタンのオブジェクト マップ項目の名前を **CancelMe** から **Cancel** に変更すると、**CancelMe** を使用するすべてのスクリプトを、**Cancel** を使用するよう手動で変更する必要があります。

オブジェクト マップ項目は一意である必要があります。重複するオブジェクト マップ項目を追加しようとすると、オブジェクト マップ項目は一意である必要があることが Silk Test Workbench から通知されます。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、..、@、[,]です。無効なロケータ パスは、空または不完全なロケータ パスです。

オブジェクト マップのルート ノードの名前を変更すると、オブジェクト マップ資産の名前も変更されます。ルート ノードの名前を既存のオブジェクト マップ資産の名前に変更した場合、オブジェクト マップ資産を保存することはできません。オブジェクト マップ資産の名前を変更すると、オブジェクト マップのルート ノードの名前も変更されます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 名前を変更するオブジェクト マップ項目を含むオブジェクト マップをダブルクリックします。
- 名前を変更するオブジェクト マップ項目を含むオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 名前を変更するオブジェクト マップ項目に移動します。
たとえば、名前を変更する項目を検索するには、ノードの展開が必要な場合があります。
6. 名前を変更するオブジェクトをクリックしてから、オブジェクトを再度クリックするか、**F2** を押しします。
インライン編集モードにアクセスすると、オブジェクトを 1 回クリックして編集できます。**Tab** を押して、オブジェクト マップ項目の名前と説明を切り替えます。
項目フィールドの周囲に黒のボックスが表示され、テキストが強調表示されます。

7. 使用する項目名を入力し、Enter を押しします。

無効な文字を使用すると、項目名が赤で表示されます。

新しい名前が **項目名** リストに表示されます。

8. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。


- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。



注: オブジェクト マップ ツリーに含まれるすべてのノードのすべての子ノードは、オブジェクト マップを保存するときにアルファベット順にソートされます。

変更した項目名を既存のビジュアルテストまたはスクリプトで使用する場合は、新しい項目名を使用するようにビジュアルテストまたはスクリプトを手動で変更する必要があります。

 **注:** Web アプリケーションまたはモバイル Web アプリの記録中に、**操作の選択** ダイアログでオブジェクト マップ エントリの名前を直接変更できます。オブジェクトを右クリックして、**操作の選択** ダイアログの **オブジェクトの識別** 領域を展開します。そして、**オブジェクト マップ ID** フィールドでオブジェクト マップ エントリを編集できます。この機能は、次のブラウザーに対するテストで利用できます。

- Microsoft Edge
- Apple Safari
- Mozilla Firefox 41 以降
- Google Chrome 50 以降
- モバイル ブラウザー

オブジェクト マップの変更

デフォルトでは、オブジェクト マップは読み取り専用です。オブジェクト マップを使用するビジュアルテストまたはスクリプトを記録すると、オブジェクト マップ項目が資産としてデータベースに追加されます。

オブジェクト マップを使用するビジュアルテストまたはスクリプトの記録が完了すると、別のユーザーがそのオブジェクト マップを使用しているかどうかは Silk Test Workbench で判断されます。別のユーザーがオブジェクト マップを変更していると、その旨を通知するメッセージ ボックスが表示され、変更しているユーザーの名前が示されます。ロックされたオブジェクト マップによって記録できなくなるのを防ぐには、**オプション** ダイアログ ボックスの **ロック済みのオブジェクト マップに対してロケータを記録する オプション** を **はい** に設定します。このように設定すると、別のユーザーがオブジェクト マップを変更している場合に、以前に識別されたオブジェクト マップ項目が適用可能な場合に使用され、新しいオブジェクトを記録する場合にはロケータが使用されます。

既存のオブジェクト マップは、オブジェクト マップに構造化要素をさらに追加したとしても、記録中に既存のオブジェクト マップ識別子を再利用することができます。

例：既存のオブジェクト マップへの DIV の追加

次の単純なオブジェクト マップの email フィールドと login フィールドをまとめる DIV 要素を追加することを考えます。

```
demo_borland_com //BrowserApplication
  BrowserWindow //
BrowserWindow
  login-form email //
INPUT[@id='login-form:email']
  login-form login //
INPUT[@id='login-form:login']
```

新たに DIV `loginArea` を追加することにより、オブジェクト マップの構造を変更できますが、オブジェクト マップは、記録中にオブジェクト マップ識別子は正しく再利用することができます。

```
demo_borland_com //BrowserApplication
  BrowserWindow //
BrowserWindow
loginArea //DIV[@id='loginArea']
  login-form email //
INPUT[@id='login-form:email']
```

```
login-form login //
INPUT[@id='login-form:login']
```

オブジェクト マップのロケーターの変更

ビジュアルテストまたはスクリプトを記録するときに、ロケーターは自動的にオブジェクト マップ項目に関連付けられます。ただし、より汎用的にするために、ロケーター パスを変更できます。たとえば、テストアプリケーションで特定のコントロールに自動的に日付または時刻が割り当てられる場合、ワイルドカードを使用するようにそのコントロールのロケーターを変更できます。ワイルドカードを使用すると、それぞれのテストで異なる日付または時刻が挿入される場合でも、各テストで同じロケーターを使用できます。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 変更するロケーターを含むオブジェクト マップをダブルクリックします。
- 変更するロケーターを含むオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 変更するロケーターに移動します。
たとえば、変更するロケーターを検索するには、ノードの展開が必要な場合があります。
6. 変更するロケーター パスをクリックしてから、ロケーター パスを再度クリックするか、**F2** を押します。

インライン編集モードにアクセスすると、ロケーター パスを 1 回クリックして編集できます。**Tab** を押して、オブジェクト マップ項目の名前と説明を切り替えます。

ロケーター パス フィールドの周囲に黒のボックスが表示され、テキストが強調表示されます。

7. 有効なロケーター パスがある場合は、使用する項目名とロケーター パスを入力して **Enter** を押すことができます。有効なロケーター パスを判別するには、以下のステップで説明するように、**オブジェクトの識別** ダイアログ ボックスを使用します。

- a) **ツール** > **オブジェクトの識別** を選択します。

テストアプリケーションがオブジェクト マップに関連付けられていないと、**アプリケーションの選択** ダイアログ ボックスが表示されることがあります。この場合は、テストするアプリケーションを選択し、**OK** をクリックします。

オブジェクトの識別 ダイアログ ボックスが開きます。

- b) **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケータを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャします。

c) **識別の開始** をクリックします。

d) 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホット キー** モードでオブジェクトをキャプチャします。
デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキスト ボックスに表示されます。

e) **ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。

f) **貼り付け** をクリックします。Silk Test Workbench によって、項目名および関連付けられたロケータがオブジェクト マップに追加されます。

8. 必要に応じて、ニーズに合わせて項目名またはロケータ テキストを変更します。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。

オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、.、@、[,] です。

無効なロケータ パスは、空または不完全なロケータ パスです。

9. 次のいずれか 1 つのステップを行います :

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

変更したロケータ パスが既存のビジュアル テストまたはスクリプトによって使用されている場合は、新しいロケータ パスを使用するように、そのビジュアル テストまたはスクリプトを手動で変更する必要があります。

オブジェクト マップの比較

2 つの異なるオブジェクト マップや、同じオブジェクト マップの 2 つのバージョンを比較して、これらの間の相違点を調べることができます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 調査したいオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2 つの異なるオブジェクト マップを比較する場合は、**Ctrl + クリック** を使って比較する 2 つのオブジェクト マップを選択します。
4. 同じオブジェクト マップの 2 つのバージョンを比較するには :

- a) 調査したいオブジェクト マップを右クリックして、**すべてのバージョンを表示** を選択します。
- b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。

5. **相違点の表示** をクリックします。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

テスト アプリケーションからのオブジェクト マップの更新

テスト アプリケーションの項目が変化した場合は、**オブジェクト マップ** UI を使用してそれらの項目のロケータを更新できます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 使用するオブジェクト マップをダブルクリックします。
- 使用するオブジェクト マップを右クリックし、**開く** をクリックします。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押し、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. **ロケータの更新** をクリックします。**オブジェクトの識別** ダイアログ ボックス が表示され、Silk Test Workbench によってテスト アプリケーションが開かれます。
6. 記録するオブジェクトの上にカーソルを合わせて、**Ctrl+Alt** を押します。Silk Test Workbench の **ロケータ** テキスト フィールドにロケータ文字列が表示されます。
7. **ロケータの詳細** テーブルで、使用するロケータを選択します。
8. **ロケータ** テキスト フィールドに表示されているロケータから、使用しない属性を削除します。
9. **ロケータの検証** をクリックして、ロケータが機能することを検証します。
10. **ロケータをエディターに貼り付け** をクリックして、オブジェクト マップのロケータを更新します。
11. 変更されたオブジェクト マップを保存します。

AUT からオブジェクト マップ項目を更新するときに、オブジェクト マップ ツリーのリーフ ノードの XPath 表現のみを変更できます。親ノードの XPath 表現を変更することはできません。オブジェクト マップ ツリー内のより高いレベルのノードにある XPath 表現が更新後に整合しなくなると、エラー メッセージが表示されます。

例

たとえば、次の3つの階層レベルを持つオブジェクト マップ ID を含むオブジェクト マップ項目があるとします：

```
WebBrowser.Dialog.Cancel
```

これらの階層レベルに対応する XPath 表現は次のようになります：

```
/BrowserApplication//Dialog//PushButton[@caption='Cancel']
```

- 最初の階層レベル： /BrowserApplication
- 2 番目の階層レベル： //Dialog
- 3 番目の階層レベル： //PushButton[@caption='Cancel']

次のロケーターを使用して、オブジェクト マップ項目を更新できます：

```
/BrowserApplication//Dialog//PushButton[@id='123']
```

- 最初の階層レベル： /BrowserApplication
- 2 番目の階層レベル： //Dialog
- 3 番目の階層レベル： //PushButton[@id='123']

2 番目のレベルの階層が一致しないため、次のロケーターを使用してオブジェクト マップ項目を更新することはできません：

```
/BrowserApplication//BrowserWindow//PushButton[@id='9999999']
```

- 最初の階層レベル： /BrowserApplication
- 2 番目の階層レベル： //BrowserWindow
- 3 番目の階層レベル： //PushButton[@id='9999999']

オブジェクト マップ項目のコピー

オブジェクト マップ内、またはオブジェクト マップ間で、オブジェクト マップ エントリをコピーおよび貼り付けできます。たとえば、2つの異なるテスト アプリケーションに同じ機能が存在する場合は、一方のオブジェクト マップの一部分をコピーして、他方のオブジェクト マップに貼り付けることができます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- コピーするオブジェクト マップ項目を含むオブジェクト マップをダブルクリックします。
- コピーするオブジェクト マップ項目を含むオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押し、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. コピーするオブジェクト マップ項目に移動します。

たとえば、コピーするオブジェクト マップ項目を検索するには、ノードの展開が必要な場合があります。

6. 次のいずれか 1 つを選んでください：

- コピーするオブジェクト マップ項目を右クリックし、**ツリーのコピー** を選択します。
- コピーするオブジェクト マップ項目をクリックし、**編集 > コピー** を選択します。
- コピーするオブジェクト マップ項目をクリックし、Ctrl+C を押します。

7. オブジェクト マップ階層で、コピーした項目を貼り付ける位置に移動します。

たとえば、階層の第 1 レベルに項目を組み込むには、項目リストの最初の項目の名前をクリックします。特定の項目の 1 レベル下にコピーする項目の位置を設定するには、コピーする項目の上にある項目をクリックします。

オブジェクト マップ間でコピーして貼り付けるには、オブジェクト マップ項目をコピーしたマップを終了し、オブジェクト マップ項目を貼り付けるオブジェクト マップを開いて編集する必要があります。

8. 次のいずれか 1 つを選んでください：

- コピーしたオブジェクト マップ項目を貼り付けるオブジェクト マップ内の位置を右クリックし、**貼り付け** を選択します。
- コピーしたオブジェクト マップ項目を貼り付けるオブジェクト マップ内の位置をクリックし、**編集 > 貼り付け** を選択します。
- コピーしたオブジェクト マップ項目を貼り付けるオブジェクト マップ内の位置をクリックし、Ctrl+V を押します。

オブジェクト マップ間でコピーして貼り付けるには、オブジェクト マップ項目を貼り付ける前に、**編集** を必ずクリックします。

オブジェクト マップ項目が、階層内の新しい位置に表示されます。

9. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

移動したオブジェクト マップ項目を既存のビジュアル テストまたはスクリプトで使用する場合は、階層内の新しい位置を使用するようにビジュアル テストまたはスクリプトを手動で変更する必要があります。

オブジェクト マップ項目の追加

ビジュアル テストまたはスクリプトを記録すると、オブジェクト マップ項目が自動的に作成されます。場合によっては、手動でオブジェクト マップ項目を追加することもできます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

- 新しい項目を追加したい場所で、オブジェクト マップをダブルクリックします。オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

- 編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押し、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

- オブジェクト マップ階層で、新しいオブジェクト マップ項目を追加したい位置の下の項目を右クリックします。

たとえば、階層の第 1 レベルに項目を組み込むには、項目リストの最初の項目の名前を右クリックします。特定の項目の 1 レベル下に新しい項目の位置を設定するには、新しい項目を配置したい位置の下の項目をクリックします。

- 新規挿入** をクリックします。新しい項目が、現在のノードの最初の子として階層に追加されます。

- 有効なロケータ パスがある場合は、使用する項目名とロケータ パスを入力して Enter を押すことができます。有効なロケータ パスを判別するには、以下のステップで説明するように、**オブジェクトの識別** ダイアログ ボックスを使用します。

- ツール > オブジェクトの識別** を選択します。

テスト アプリケーションがオブジェクト マップに関連付けられていないと、**アプリケーションの選択** ダイアログ ボックスが表示されることがあります。この場合は、テストするアプリケーションを選択し、**OK** をクリックします。

オブジェクトの識別 ダイアログ ボックスが開きます。

- 選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケータを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャします。

- 識別の開始** をクリックします。

- 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホット キー** モードでオブジェクトをキャプチャします。

デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキスト ボックスに表示されます。

- ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。

- 貼り付け** をクリックします。Silk Test Workbench によって、項目名および関連付けられたロケータがオブジェクト マップに追加されます。

- 必要に応じて、ニーズに合わせて項目名またはロケータ テキストを変更します。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。

オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、.、@、[,] です。

無効なロケータ パスは、空または不完全なロケータ パスです。

9. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。



注: オブジェクト マップ ツリーに含まれるすべてのノードのすべての子ノードは、オブジェクト マップを保存するときにアルファベット順にソートされます。

スクリプトからオブジェクト マップを開く

スクリプトを編集している際に、スクリプトのオブジェクト マップ エントリを右クリックし、**資産を開く** を選択してオブジェクト マップを開くことができます。オブジェクト マップは GUI 上で開かれます。

例

```
// VB .NET code
<TestMethod()> Public Sub TestMethod1()
    With _desktop.Window("Untitled -
Notepad").TextField("TextField").TypeKeys("hello")
    End With
End Sub
```

上記のコード例で、Untitled - Notepad エントリをオブジェクト マップで開く場合は、Untitled - Notepad を右クリックします。オブジェクト マップで Untitled - Notepad.TextField エントリをオブジェクト マップで開く場合は、TextField を右クリックします。

テスト アプリケーションでのオブジェクト マップ項目の検索

オブジェクト マップ項目を追加または記録したあと、**検索** をクリックして、テスト アプリケーションで項目をハイライトできます。変更する項目であることをオブジェクト マップ内で確認する場合などに項目をハイライトできます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。




ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 使用するオブジェクト マップをダブルクリックします。
- 使用するオブジェクト マップを右クリックし、**開く** をクリックします。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。


 **注:** 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. オブジェクト マップ階層で、テスト アプリケーションでハイライトするオブジェクト マップ項目を選択します。

 **注:** テスト アプリケーションの 1 つのインスタンスのみが実行中であることを確認します。テスト アプリケーションの複数のインスタンスを実行すると、複数のオブジェクトがロケータと一致するためエラーになります。

6. **検索** をクリックします。

テスト アプリケーションがオブジェクト マップに関連付けられていないと、**アプリケーションの選択** ダイアログ ボックスが表示されることがあります。この場合は、テストするアプリケーションを選択し、**OK** をクリックします。

アプリケーションの選択 ダイアログ ボックスで誤ったアプリケーションを選択した場合は、**検索** サブメニューで **アプリケーションを指定して検索** をクリックします。**アプリケーションの選択** ダイアログ ボックスが開き、正しいテスト アプリケーションを選択できます。


Silk Test Workbench によってテスト アプリケーションが開かれ、オブジェクト マップ項目を示すコントロールの周囲に緑のボックスが表示されます。

オブジェクト マップのエラーの検出

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。**オブジェクト マップ** ウィンドウのツール バーを使用して、エラーに移動します。

1. **表示** > **アセット ブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセット ブラウザ** を選択します。


2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。

 **ヒント:** 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- トラブルシューティングするオブジェクト マップをダブルクリックします。
- トラブルシューティングするオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。

 **注:** 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 赤色で表示された項目名またはロケータ テキストを探します。
6. 必要に応じて、ニーズに合わせて項目名またはロケータ テキストを変更します。
無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツールヒントにエラーの説明が表示されます。
オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、.、@、[,] です。
無効なロケータ パスは、空または不完全なロケータ パスです。
7. 次のいずれか 1 つのステップを行います：
 - **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。
Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。
 - **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。
Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

オブジェクト マップ項目の削除

テスト アプリケーションに存在しなくなったなどの理由により、オブジェクト マップから項目を削除できます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 削除するオブジェクト マップ項目を含むオブジェクト マップをダブルクリックします。オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。
デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。
編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。
5. 削除するオブジェクト マップ項目に移動します。
たとえば、削除するオブジェクト マップ項目を検索するには、ノードの展開が必要な場合があります。
6. 次のいずれか 1 つを選んでください：
 - 削除するオブジェクト マップ項目を右クリックし、**削除** を選択するか、そのオブジェクト マップ項目のすべての子項目も削除する場合は **ツリーの削除** を選択します。
 - 削除するオブジェクト マップ項目をクリックし、**Del** を押すか、そのオブジェクト マップ項目のすべての子項目も削除する場合は **Ctrl+Del** を押します。

オブジェクト マップ項目を削除した後、フォーカスはオブジェクト マップの次の項目に移動します。

7. 次のいずれか 1 つのステップを行います：
 - **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。


Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

削除したオブジェクト マップ項目または子オブジェクトを既存のビジュアル テストまたはスクリプトで使用する場合は、そのオブジェクト マップ項目への参照をビジュアル テストまたはスクリプトで、手動で変更する必要があります。

オブジェクト マップを最初に書き出す


ベストプラクティスとして、テストを記録する前に、すべてのオブジェクト マップ項目を書き出し、確認することをお勧めします。

AUT のすべての利用可能な項目をもつオブジェクト マップを最初に書き出すためには、テスト対象アプリケーションのすべてのオブジェクトをクリックし、すべてのウィンドウとダイアログ ボックスを開くビジュアル テストを作成する必要があります。その後、各オブジェクトに対するオブジェクト マップ項目を確認し、機能テストを記録する前に、必要な変更を加えることができます。オブジェクト マップ項目を確認し、修正した後に、オブジェクト マップを書き出すために作成したビジュアル テストを削除できます。


 **ヒント:** オブジェクト マップ中の項目間を移動するには、矢印キーを使用できます。

オブジェクト マップの要素のグループ化

オブジェクト マップの項目が一貫した親オブジェクトを持たない場合、XPath の現在の要素のロケーターを意味するロケーター "." を持つ新しいツリー項目を追加して、これらの要素をグループ化できます。

 **警告:** オブジェクト マップ項目をグループ化すると、それらの項目を使用するすべてのビジュアル テストまたはスクリプトが影響を受けます。これらの項目を使用するすべてのビジュアル テストまたはスクリプトは、新しいロケーターを使用するように手動で変更する必要があります。


1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。

 **ヒント:** 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください:

- 編集するオブジェクト マップをダブルクリックします。
- 編集するオブジェクト マップを右クリックし、**開く** をクリックします。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。

 **注:** 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押し、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 新しく追加する構成項目のすぐ上のツリー項目を右クリックして、**新規挿入** を選択します。
6. 新しいオブジェクト マップ項目の **項目名** フィールドをダブルクリックします。
7. 使用する項目名を入力し、Enter を押します。
無効な文字を使用すると、項目名が赤で表示されます。
新しい名前が **項目名** リストに表示されます。
8. 新しいオブジェクト マップ項目の **ロケーター パス** フィールドをクリックして、フィールドに「.」を入力します。
9. Enter を押します。
- 10 新しい項目の下に新たに配置し直したいすべてのオブジェクト マップ項目に対して、次の操作を行います。
 - a) 配置し直したい項目を右クリックして、**ツリーの切り取り** を選択します。
 - b) 新しい構成項目を右クリックして、**貼り付け** を選択します。
- 11 次のいずれか 1 つのステップを行います：
 - **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。
Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。
 - **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。
Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

オブジェクト マップ：よくある質問

このセクションでは、Silk Test Workbench でオブジェクト マップを使用する時に直面する可能性のある質問を一覧します。

複数のオブジェクト マップを単一のオブジェクト マップのマージする方法

既存のオブジェクト マップをマージせずに、同じオブジェクト マップに記録することを Micro Focus は推奨しますが、テキスト エディタを使用して複数のオブジェクト マップを単一のマップにマージすることができます。

テスト スクリプトを削除したときにオブジェクト マップで起こること

オブジェクト マップ エントリを含んだテスト スクリプトまたはビジュアル テストを削除しても、関連するオブジェクト マップは変更されません。すべてのオブジェクト マップ エントリはそのまま残ります。

テスト対象アプリケーションのオブジェクト マップを手動で作成する方法

Micro Focus では、テストの記録時にオブジェクト マップを作成することを推奨します。しかし、空のオブジェクト マップを作成し、このマップにオブジェクト マップの項目を手動で追加することもできます。

新しいオブジェクト マップを作成するには、**アセットブラウザ** の **オブジェクト マップ** を右クリックし、**オブジェクト マップの新規作成** を選択します。


イメージ解決のサポート

イメージ解決は、次の場合に使用できます。

- オブジェクト解決で識別できない、高度にカスタマイズされたコントロールを含むテスト アプリケーションを簡単に操作する場合。座標ベースのクリックの代わりにイメージ クリックを使用し、指定されたイメージをクリックできます。
- テスト対象アプリケーションのグラフィカル オブジェクト (グラフなど) をテストする場合。
- テスト対象アプリケーションの視覚的な UI のチェックを実行する場合。

認識されないコントロールをクリックするには、ImageClick メソッドとイメージ資産を使用します。テスト対象アプリケーションに、認識されないコントロールがあるかどうかを確認するには、VerifyAsset メソッドとイメージ検証を使用します。


イメージ解決メソッドは、Silk Test Workbench でサポートされるすべてのテクノロジー ドメインでサポートされます。

 **注:** イメージ解決メソッドは、画面に表示されていないコントロールに対しては機能しません。たとえば、スクロールして画面外にあるイメージに対しては、イメージ解決を使用できません。

イメージ クリックの記録

イメージ クリックの記録を行うと、大量のイメージが生成されてわかりにくくなるため、デフォルトではイメージ クリックの記録が無効となり、座標ベースのクリックの記録が優先されます。


イメージ クリックの記録を有効にするには、**ツール > オプション > 記録 > 全般** をクリックし、**'ImageClick' を記録する** の値を **はい** に設定します。

 **注:** モバイルブラウザを記録する場合、イメージ クリックの記録を有効にする必要はありません。

イメージ クリックの記録を有効にすると、Silk Test Workbench は、オブジェクト解決またはテキスト解決ができない場合に、ImageClick メソッドを記録します。イメージ クリックが記録されない場合でも、任意のコントロールに対してイメージ クリックをスクリプトに挿入できます。

ImageClick 操作を記録しない場合は、イメージ クリックの記録をオフに切り替えて通常のクリックまたはテキスト クリックを記録できます。

 **注:** 記録したイメージは再利用されません。Silk Test Workbench は、記録するイメージ クリックごとに新しいイメージ資産を作成します。


 **注:** イメージ クリックの記録は、Java AWT/Swing コントロールを使用するアプリケーションまたはアプレットではサポートされません。

イメージ解決メソッド

Silk Test Workbench では、イメージ解決用に次のメソッドが用意されています。

メソッド	説明
ImageClick	資産で指定されたイメージの中央をクリックします。イメージが見つかるか、オブジェクト解決タイムアウト (同期オプションで定義可能) が経過するまで待機します。
ImageExists	資産で指定されたイメージが存在するかどうかを返します。

メソッド	説明
ImageRectangle	資産で指定されたイメージのオブジェクト相対矩形を返します。
ImageClickFile	ファイルで指定されたイメージをクリックします。
ImageExistsFile	ファイルで指定されたイメージが存在するかどうかを返します。
ImageRectangleFile	ファイルで指定されたイメージのオブジェクト相対矩形を返します。
VerifyAsset	検証資産を実行します。検証に合格しなかった場合、VerificationFailedException がスローされます。

 **注:** イメージ解決メソッドは、画面に表示されていないコントロールに対しては機能しません。たとえば、スクロールして画面外にあるイメージに対しては、イメージ解決を使用できません。

イメージ資産

イメージ資産は、次の場合に使用できます。

- オブジェクト解決で識別できない、高度にカスタマイズされたコントロールを含むテスト アプリケーションを簡単に操作する場合。座標ベースのクリックの代わりにイメージ クリックを使用し、指定されたイメージをクリックできます。
- テスト対象アプリケーションのグラフィカル オブジェクト (グラフなど) をテストする場合。

イメージ資産は、イメージと、Silk Test Workbench で資産を操作するために必要な追加情報で構成されます。

Silk Test Workbench では、イメージ資産用に次のメソッドが用意されています。

メソッド	説明
ImageClick	指定されたイメージ資産の中央をクリックします。イメージが見つかるか、オブジェクト解決タイムアウト (同期オプションで定義可能) が経過するまで待機します。
ImageExists	指定されたイメージ資産があるかどうかを返します。
ImageRectangle	指定されたイメージ資産のオブジェクト相対矩形を返します。

Creating an Image Asset

You can create image assets in one of the following ways:

- By inserting a new image asset into an existing visual test or script.
- During recording.
- By using the **Asset Browser**.

To create a new image asset in the **Asset Browser**, perform the following steps:

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. In the menu, click **New Image Asset**.
3. In the **Asset Types** list, right-click **Image Asset** and select **New Image Asset**. The image asset UI opens.
4. Select how you want to add an image to the asset.
 - If you want to use an existing image, click **Browse** and select the image file.
 - If you want to capture a new image from the UI of the application under test, select **Capture**. If you are testing a Web application, you can select the browser on which you want to capture the image from the **Select Application** window.

- If you want to capture a new image after a delay of three seconds, for example to expand a menu in the application under test before the image is captured, select **Capture with Pause**.

5. If you have selected to capture a new image, select the area of the screen that you want to capture and click **Capture Selection**.

6. In the **Name** text field, type a meaningful name for the new image asset.



注: If you have selected to use an existing image, the name of the image asset is by default the name of the existing image.

7. *Optional:* Click **Verify** to check if Silk Test Workbench can find the image asset in the UI of the AUT.

Web アプリケーションをテストする場合、**アプリケーションの選択** ウィンドウからイメージをキャプチャするブラウザを選択できます。

8. *Optional:* Check the **Click position** check box to select the location on which any clicks on the image asset are performed.

The default location is the center of the image. Type the location into the **x** and **y** fields or select the location on the image.

9. **精度レベル** を指定します。

精度レベルは、検証されるイメージがテスト対象アプリケーションのイメージと異なってもよい度合いを定義し、これを超えて異なっている場合、Silk Test Workbench はイメージが異なっていると判断します。これは、画面解像度が異なる複数のシステムまたはブラウザをテストする場合に役立ちます。誤検出を防ぐため、できるだけ精度レベルを高くすることを推奨します。デフォルトの精度レベル値は、オプションで変更できます。



注: **精度レベル** を 5 未満に設定した場合、イメージの実際の色が比較で考慮されなくなります。イメージのグレースケール表現だけが比較されます。

10 Click **Save** to save the image asset into the Silk Test Workbench repository, or click **Save and close** to additionally close the image asset UI.

The new image asset is listed in the **Asset Browser**, and you can use it to perform image clicks.

You can add multiple images to the same image asset.



注: To add an image click while recording against a mobile browser, you can right-click in the **Recording** window and select **ImageClick** from the action list.

同じイメージ資産に複数のイメージを追加する

テスト時に、異なるテスト構成を使用して、複数の環境の機能をテストする必要性が生じることがよくあります。環境が異なると、イメージ資産にキャプチャしたイメージと実際のイメージが若干異なることがあり、イメージが存在するにもかかわらずイメージ クリックが失敗することがあります。このような場合、同じイメージ資産に複数のイメージを追加できます。

イメージ資産に別のイメージを追加するには、以下を実行します。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **イメージ資産** を選択します。
3. イメージ資産に追加したいイメージをダブルクリックします。イメージ資産の UI が開きます。
4. UI の下部に表示されるプラス記号をクリックして、新しいイメージをイメージ資産に追加します。
5. Click **Save** to save the image asset into the Silk Test Workbench repository, or click **Save and close** to additionally close the image asset UI.

新しいイメージが資産に追加されます。イメージ クリックが呼び出されるたびに、一致するものに到達するまで、Silk Test Workbench は資産のイメージとテスト対象アプリケーションの UI のイメージを比較します。デフォルトで、Silk Test Workbench は資産に追加された順にイメージを比較します。



注: Silk Test Workbench で比較するイメージの順番を変更するには、イメージ資産の UI の下部でイメージをクリックし、目的の場所にドラッグします。左から右の順に比較されます。最初に比較されるのは、最も左にあるイメージです。

スクリプトから資産を開く

スクリプトを編集しているときに、資産を右クリックして **資産を開く** を選択し、資産を開くことができます。これにより、GUI で資産が開きます。

資産がシステム上のファイルへの参照である場合 (ImageClickFile によって参照される場合など)、ファイルはシステムのデフォルト エディターで開かれます。

イメージクリックの検証資産名での変数の使用

ビジュアルテストからイメージクリックを実行するとき、ビジュアルテスト変数、アクティブデータ変数、式を使用して、イメージ資産の名前を指定できます。これにより、変数や式の値に応じて、さまざまなイメージ資産で同じイメージクリックを実行できます。

VB .NET スクリプトでは、ImageClick や ImageExists などのイメージ解決メソッドを介してこの機能をサポートします。これらのメソッドに対して、イメージ資産名を文字列で指定します。この文字列を、変数や式の名前とすることができます。

ビジュアルスクリプトの場合は、イメージクリックテストステップのプロパティでイメージ資産の名前を指定する必要があります。

1. イメージクリックが含まれているビジュアルテストを開きます。
2. イメージクリックテストステップを選択します。
3. **プロパティ** ペインを開きます。
4. **パラメータ** ノードを展開します。
5. **image** フィールドの ... をクリックし、イメージ資産名として使用する変数の名前を選択します。

イメージ検証

イメージ検証を使用して、テスト対象アプリケーション (AUT) の UI にイメージがあるかどうかをチェックできます。

イメージ検証は、イメージと、Silk Test Workbench で資産を操作するために必要な追加情報で構成されます。

イメージ検証を実行するには、VerifyAsset メソッドを使用します。

Silk Test Workbench が AUT のイメージを見つけることができなかった場合、イメージ検証は失敗します。この場合、スクリプトは続行します。

AUT 内でイメージ検証のロケータが見つからなかった場合、Silk Test Workbench は ObjectNotFoundException をスローします。

イメージ検証の作成

イメージ検証は、次のいずれかの方法で作成できます。

- **アセットブラウザ** を使用。
- 記録時。
- 新しいイメージ検証を既存のビジュアルテストに挿入。

新しいイメージ検証を **アセットブラウザ** で作成するには、以下のステップを実行します。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストで **イメージ検証** を右クリックし、**新規イメージ検証** を選択します。イメージ検証の UI が開きます。
3. **識別** をクリックして、テスト対象アプリケーションの、検証するイメージを識別します。
4. **名前** テキスト フィールドに、新しいイメージ検証のわかりやすい名前を入力します。



注: 既存のイメージを使用する場合は、イメージ検証の名前はデフォルトで既存のイメージの名前になります。

5. 省略可能：最初にキャプチャしたイメージから変更されたために、テスト対象アプリケーションから同じイメージを再キャプチャする必要がある場合は、**再キャプチャ** をクリックします。

Web アプリケーションをテストする場合、**アプリケーションの選択** ウィンドウからイメージをキャプチャするブラウザを選択できます。

6. 省略可能：**検証** をクリックすると、イメージ検証が機能するかどうかをテストできます。Silk Test Workbench は、AUT の UI でイメージを、上から下へ、左から右へと検索し、最初に一致したイメージをハイライトします。
7. 省略可能：Silk Test Workbench がイメージ検証とテスト対象アプリケーション (AUT) の UI を比較するときに考慮しない除外領域をイメージ検証に追加できます。
8. 省略可能：オプション **クライアント領域のみ** を設定して、Silk Test Workbench がイメージ検証と AUT の UI を比較するときに、実際に AUT の一部であるイメージの部分だけを考慮するように定義できます。
9. **精度レベル** を指定します。

精度レベルは、検証されるイメージがテスト対象アプリケーションのイメージと異なってもよい度合いを定義し、これを超えて異なっている場合、Silk Test Workbench はイメージが異なっていると判断します。これは、画面解像度が異なる複数のシステムまたはブラウザをテストする場合に役立ちます。誤検出を防ぐため、できるだけ精度レベルを高くすることを推奨します。デフォルトの精度レベル値は、オプションで変更できます。



注: 精度レベル を 5 未満に設定した場合、イメージの実際の色が比較で考慮されなくなります。イメージのグレースケール表現だけが比較されます。

- 10 **保存** をクリックすると新しいイメージ検証が Silk Test Workbench リポジトリに保存されます。**保存して閉じる** をクリックすると、さらにイメージ検証の UI が閉じます。

新しいイメージ検証が**アセットブラウザ** に表示され、これを使用して、テスト対象アプリケーションの UI にイメージが存在するかどうかをチェックできます。

ビジュアル テストにイメージ検証を追加する

イメージ検証をビジュアル テストに追加して、テスト対象アプリケーションの UI に認識されないコントロールがあるかどうかをチェックできます。ビジュアル テストにイメージ検証を追加するには、以下を実行します。

1. ビジュアル テストを開きます。
2. イメージ検証を追加する場所の次のステップを右クリックします。
3. **挿入** > **テスト ロジック** > **検証** をクリックします。**テスト ロジック デザイナ** が開きます。
4. **次へ** をクリックします。**ロジックの種類**の**選択** ページが開きます。
5. **検証資産** をクリックします。**検証資産の構築** ページが開きます。
6. 次のいずれか 1 つのステップを行います：
 - イメージ検証の UI で新しいイメージ検証を作成するには、**新しい検証の作成** をクリックし、。
 - 既存のイメージ検証資産を挿入するには、**既存の検証の挿入** をクリックし、。
7. **次へ** をクリックします。**要約** ページが開き、新しい検証のプレビューが表示されます。
8. **終了** をクリックします。新しいイメージ検証が追加されます。

記録中にイメージ検証を追加する

イメージ検証をビジュアルテストまたは VB .NET スクリプトに追加して、テスト対象アプリケーションの UI に認識されないコントロールがあるかどうかをチェックできます。ビジュアルテストまたは VB .NET スクリプトの記録中にイメージ検証を追加するには、以下のステップを実行します。

1. 記録を開始します。
2. 検証するオブジェクトの上にマウスカーソルを移動して、**Ctrl+Alt** を押します。
モバイル Web アプリケーションを記録する場合は、オブジェクトをクリックして **検証の追加** をクリックすることもできます。
このオプションを実行すると、記録が一時的に停止され、**テスト ロジック デザイナ** ウィザードが表示されます。
3. **次へ** をクリックします。 **ロジックの種類を選択** ページが開きます。
4. **検証資産** をクリックします。 **検証資産の構築** ページが開きます。
5. 次のいずれか 1 つのステップを行います：
 - イメージ検証の UI で新しいイメージ検証を作成するには、**新しい検証の作成** をクリックし、。
 - 既存のイメージ検証資産を挿入するには、**既存の検証の挿入** をクリックし、。
6. **次へ** をクリックします。 **要約** ページが開き、新しい検証のプレビューが表示されます。
7. **終了** をクリックします。新しいイメージ検証が追加されます。
8. 記録を続けます。

イメージ検証が失敗した理由の確認

ビジュアルテストまたはスクリプトでイメージ検証の実行が失敗した場合、**イメージ検証の相違点ビュー** を開き、テスト対象アプリケーションで見つかった実際のイメージと予想されるイメージとの相違点を確認できます。

イメージ検証の相違点ビュー を開くには、以下のステップを実行します。

1. イメージ検証が失敗した実行の結果を開きます。
2. ビジュアルテストまたはスクリプトで失敗したイメージ検証のステップを右クリックします。
TrueLog Explorer **イメージ検証の相違点ビュー** が開きます。
3. 予想されるイメージと実際のイメージを比較し、イメージ検証が失敗した理由を確認します。
4. 省略可能：AUT の実際のイメージが変更されたためにイメージ検証が失敗している場合は、**予想として保存** をクリックして、実際のイメージを予想されるイメージに設定します。

イメージ検証の検証資産名での変数の使用

ビジュアルテストからイメージ検証を実行するとき、ビジュアルテスト変数、アクティブデータ変数、式を使用して、検証資産の名前を指定できます。これにより、変数や式の値に応じて、さまざまな検証資産で同じイメージ検証を実行できます。

VB .NET スクリプトでは、ImageClick や ImageExists などのイメージ解決メソッドを介してこの機能をサポートします。これらのメソッドに対して、イメージ資産名を文字列で指定します。この文字列を、変数や式の名前とすることができます。

ビジュアルスクリプトの場合は、イメージ検証テストステップのプロパティで検証資産の名前を指定する必要があります。

1. イメージ検証が含まれているビジュアルテストを開きます。
2. イメージ検証テストステップを選択します。
3. **プロパティ** ペインを開きます。

4. **名前** フィールドの ... をクリックし、検証資産名として使用する変数の名前を選択します。

イメージ検証の相違点ビュー

ビジュアルテストまたはスクリプトでイメージ検証の実行が失敗すると、**イメージ検証の相違点ビュー**に、テスト対象アプリケーションで見つかった実際のイメージと予想されるイメージとの相違点が表示されます。

イメージ検証の相違点ビューには、実際のイメージと予想されるイメージ、イメージ名、比較されたイメージのバージョンが表示されます。

特定の環境のテスト

Silk Test Workbench では、複数の種類の環境でのテストがサポートされています。


Apache Flex のサポート

Silk Test Workbench は、Internet Explorer やスタンドアロンの Flash Player を使用した Apache Flex アプリケーション、および Apache Flex 4 以降でビルドした Adobe AIR アプリケーションのテストを組み込みでサポートしています。

Silk Test Workbench では、Apache Flex 3.x および 4.x アプリケーションにおいて複数のアプリケーション ドメインもサポートされているため、サブアプリケーションをテストできます。Silk Test Workbench では、ロケーター階層ツリーの各サブアプリケーションが、関連するアプリケーション ドメイン コンテキストを持つアプリケーション ツリーとして認識されます。Apache Flex 4.x サブアプリケーションでは、ロケーター属性テーブルのルート レベルで SparkApplication クラスが使用されます。Apache Flex 3.x サブアプリケーションでは、FlexApplication クラスが使用されます。

サポートするコントロール

Apache Flex のテストで記録および再生できるコントロールの完全なリストについては、「Flex クラス リファレンス」を参照してください。

 **注:** Silk Test Flex オートメーション SDK は、Apache Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Apache Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、Flex オートメーション API の typekey ステートメントでは、すべてのキーはサポートされません。テキスト入力ステートメントを使用してこの問題を解決できます。Flex オートメーション API の詳細については、『Apache Flex リリース ノート』を参照してください。

Adobe Flash Player で実行するための Flex アプリケーションの構成

Apache Flex アプリケーションを Flash Player で実行するには、以下のいずれか、または両方の条件が満たされている必要があります。

- Flex アプリケーションを作成する開発者は、アプリケーションを EXE ファイルとしてコンパイルする必要があります。アプリケーションは、ユーザーが起動すると、Flash Player で開きます。Windows Flash Player は、<http://www.adobe.com/support/flashplayer/downloads.html> からインストールします。
 - ユーザーが、Windows Flash Player Projector をインストールしている必要があります。ユーザーは、Flex の .SWF ファイルを開いた場合に Flash Player で開くように構成できます。Apache Flex 開発者スイートをインストールしないと、Flash Player をインストールしても Windows Flash Projector はインストールされません。Windows Flash Projector は、<http://www.adobe.com/support/flashplayer/downloads.html> からインストールします。
1. Microsoft Windows 7 および Microsoft Windows Server 2008 R2 では、管理者として実行されるように Flash Player を構成します。以下の手順を実行します。
 - a) Adobe Flash Player プログラム ショートカットまたは FlashPlayer.exe ファイルを右クリックして、**プロパティ** をクリックします。
 - b) **プロパティ** ダイアログ ボックスで、**互換性** タブをクリックします。

- c) 管理者としてこのプログラムを実行する チェック ボックスをオンにして、OK をクリックします。
2. コマンド プロンプト (cmd.exe) で以下のコマンドを入力して、Flash Player で .SWF ファイルを起動します。

```
"<Application_Install_Directory>%ApplicationName.swf"
```

デフォルトで、<SilkTest_Install_Directory> は Program Files¥Silk¥Silk Test にあります。

Component Explorer の起動

Silk Test には、Component Explorer というサンプルの Apache Flex アプリケーションが含まれています。Component Explorer は、Adobe オートメーション SDK および Silk Test 固有のオートメーション実装を使用してコンパイルされており、テスト用に事前に構成されています。

Internet Explorer で、<http://demo.borland.com/flex/SilkTest19.5/index.html> を開きます。デフォルトブラウザでアプリケーションが起動します。

Apache Flex アプリケーションのテスト

Silk Test は、Apache Flex アプリケーションのテストを組み込みでサポートしています。Silk Test では、いくつかのサンプル Apache Flex アプリケーションを提供しています。サンプル アプリケーションには、<http://demo.borland.com/flex/SilkTest19.5/index.html> からアクセスできます。


新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『リリースノート』を参照してください。

独自の Apache Flex アプリケーションをテストする前に、Apache Flex 開発者は以下のステップを実行する必要があります。

- Apache Flex アプリケーションのテストの有効化
- テスト可能な Apache Flex アプリケーションの作成
- Apache Flex コンテナのコーディング
- カスタム コントロールのオートメーション サポートの実装

独自の Apache Flex アプリケーションをテストするには、以下のステップを実行します。

- ローカルの Flash Player のセキュリティ設定の構成
- ビジュアル テストの記録
- ビジュアル テストの再生
- Apache Flex スクリプトのカスタマイズ
- カスタム Apache Flex コントロールのテスト

 **注:** Apache Flex アプリケーションを読み込み、Flex オートメーション フレームワークを初期化する
とき、テストを実行するマシンおよび Apache Flex アプリケーションの複雑度に応じて、多少の時間
がかかる場合があります。アプリケーションが完全に読み込まれるように、ウィンドウのタイムアウト
値を高い値に設定します。

Apache Flex カスタム コントロールのテスト

Silk Test Workbench では、Apache Flex カスタム コントロールのテストがサポートされています。ただし、デフォルトでは、Silk Test Workbench は、カスタム コントロールの個別のサブコントロールを記録および再生することはできません。

カスタム コントロールをテストする場合、以下のオプションが存在します。

- 基本サポート

基本サポートでは、動的呼び出しを使用して、再生中にカスタム コントロールと対話します。作業量が少なく済むこのアプローチは、テスト アプリケーションにおいて、Silk Test Workbench が公開しないカスタム コントロールのプロパティおよびメソッドにアクセスする場合に使用します。カスタム コ

トロールの開発者は、コントロールのテストを容易にすることのみを目的としたメソッドおよびプロパティをカスタム コントロールに追加することもできます。ユーザーは、動的呼び出し機能を使用してこれらのメソッドやプロパティを呼び出すことができます。

基本サポートには以下のような利点があります。

- 動的呼び出しでは、テスト アプリケーションのコードを変更する必要がありません。
- 動的呼び出しを使用することによって、ほとんどのテストのニーズを満たすことができます。

基本サポートには以下のような短所があります。

- ロケーターには、具体的なクラス名が組み込まれません (たとえば、Silk Test Workbench では「//FlexSpinner」ではなく「//FlexBox」と記録されます)。
- 記録のサポートが限定されます。
- Silk Test Workbench では、イベントを再生できません。

例を含む動的呼び出しの詳細については、「*Apache Flex* メソッドの動的呼び出し」を参照してください。

- 高度なサポート

高度なサポートでは、カスタム コントロールに対して、特定のオートメーション サポートを作成できます。この追加のオートメーション サポートによって、記録のサポートおよびより強力な再生のサポートが提供されます。高度なサポートには以下のような利点があります。


- イベントの記録と再生を含む、高レベルの記録および再生のサポートが提供されます。
- Silk Test Workbench では、カスタム コントロールが他のすべての組み込み Apache Flex コントロールと同様に処理されます。
- Silk Test Workbench API とシームレスに統合できます。
- Silk Test Workbench では、ロケーターで具体的なクラス名が使用されます (たとえば、Silk Test Workbench では「//FlexSpinner」と記録されます)。

高度なサポートには以下のような短所があります。

- 実装作業が必要です。テスト アプリケーションを変更し、Open Agent を拡張する必要があります。


Flex メソッドの動的呼び出し

動的呼び出し機能を使用して Silk Test Workbench が対象としないコントロールのメソッドを呼び出したリ、プロパティを取得/設定することができます。この機能は、カスタム コントロールを使用したり、カスタマイズせずに Silk Test Workbench がサポートするコントロールを使用する場合に有効です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- Flex API で定義されているすべてのパブリック メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point など) が含まれます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

例

たとえば、アプリケーション開発者が次のメソッドとプロパティを持つ Calculator カスタム コントロールを作成したとします。

```
public function reset() : void
public function add(number1 : int, number2 : int) : int
public function get description : String
```

テスト担当者は、テスト内からメソッドを直接呼び出すことができます。例：

```
customControl.Invoke("reset")
Dim sum as Integer = customControl.Invoke("add", 1, 2)
Dim description As String = customControl.GetProperty("description")
```

テスト アプリケーションでのカスタム コントロールの定義

通常、テスト アプリケーションには、アプリケーションの開発中に追加されたカスタム コントロールがすでに含まれています。テスト アプリケーションにすでにカスタム コントロールが含まれている場合は、「動的呼び出しを使用して Flex カスタム コントロールをテストする」または「オートメーション サポートを使用してカスタム コントロールをテストする」に進んでください。

この手順では、Flex アプリケーション開発者が Flex で Spinner カスタム コントロールを作成する方法を示します。このトピックで作成する Spinner カスタム コントロールは、カスタム コントロールの実装およびテストのプロセスを説明するために、いくつかのトピックで使用されています。

Spinner カスタム コントロールは、以下のグラフィックに示すように、2 つのボタンと 1 つのテキスト フィールドを含んでいます。



ユーザーは、**Down** をクリックしてテキスト フィールドに表示されている値を 1 減分させ、**Up** をクリックしてテキスト フィールドの値を 1 増分させることができます。

カスタム コントロールには、設定および取得が可能なパブリックの Value プロパティが用意されています。

1. テスト アプリケーションで、コントロールのレイアウトを定義します。
たとえば、Spinner コントロール タイプでは、以下のように記述します。

```
<?xml version="1.0" encoding="utf-8"?>
<customcontrols:SpinnerClass xmlns:mx="http://www.adobe.com/2006/mxml"
xmlns:controls="mx.controls.*" xmlns:customcontrols="customcontrols.*">
  <controls:Button id="downButton" label="Down" />
  <controls:TextInput id="text" enabled="false" />
```

```
<controls:Button id="upButton" label="Up"/>
</customcontrols:SpinnerClass>
```

2. カスタム コントロールの実装を定義します。
たとえば、Spinner コントロール タイプでは、以下のように記述します。

```
package customcontrols
{
    import flash.events.MouseEvent;

    import mx.containers.HBox;
    import mx.controls.Button;
    import mx.controls.TextInput;
    import mx.core.UIComponent;
    import mx.events.FlexEvent;

    [Event(name="increment",      type="customcontrols.SpinnerEvent")]
    [Event(name="decrement",     type="customcontrols.SpinnerEvent")]

    public class SpinnerClass extends HBox
    {
        public var downButton : Button;
        public var upButton : Button;
        public var text : TextInput;
        public var ssss: SpinnerAutomationDelegate;
        private var _lowerBound : int = 0;
        private var _upperBound : int = 5;

        private var _value : int = 0;
        private var _stepSize : int = 1;

        public function SpinnerClass() {
            addEventListener(FlexEvent.CREATION_COMPLETE,
creationCompleteHandler);
        }

        private function creationCompleteHandler(event:FlexEvent) : void {
            downButton.addEventListener(MouseEvent.CLICK, downButtonClickHandler);
            upButton.addEventListener(MouseEvent.CLICK, upButtonClickHandler);
            updateText();
        }

        private function downButtonClickHandler(event : MouseEvent) : void {
            if(Value - stepSize >= lowerBound) {
                Value = Value - stepSize;
            }
            else {
                Value = upperBound - stepSize + Value - lowerBound + 1;
            }

            var spinnerEvent : SpinnerEvent = new
SpinnerEvent(SpinnerEvent.DECREMENT);
            spinnerEvent.steps = _stepSize;
            dispatchEvent(spinnerEvent);
        }

        private function upButtonClickHandler(event : MouseEvent) : void {
            if(cValue <= upperBound - stepSize) {
                Value = Value + stepSize;
            }
        }
    }
}
```



```

    }
    else {
        Value = lowerBound + Value + stepSize - upperBound - 1;
    }

    var spinnerEvent : SpinnerEvent = new
SpinnerEvent(SpinnerEvent.INCREMENT);
    spinnerEvent.steps = _stepSize;
    dispatchEvent(spinnerEvent);
}

private function updateText() : void {
    if(text != null) {
        text.text = _value.toString();
    }
}

public function get Value() : int {
    return _value;
}

public function set Value(v : int) : void {
    _value = v;
    if(v < lowerBound) {
        _value = lowerBound;
    }
    else if(v > upperBound) {
        _value = upperBound;
    }
    updateText();
}

public function get stepSize() : int {
    return _stepSize;
}

public function set stepSize(v : int) : void {
    _stepSize = v;
}

public function get lowerBound() : int {
    return _lowerBound;
}

public function set lowerBound(v : int) : void {
    _lowerBound = v;
    if(Value < lowerBound) {
        Value = lowerBound;
    }
}

public function get upperBound() : int {
    return _upperBound;
}

public function set upperBound(v : int) : void {
    _upperBound = v;
    if(Value > upperBound) {

```

```

        Value = upperBound;
    }
}
}
}
}

```

3. コントロールが使用するイベントを定義します。
たとえば、Spinner コントロール タイプでは、以下のように記述します。

```

package customcontrols
{
    import flash.events.Event;

    public class SpinnerEvent extends Event
    {
        public static const INCREMENT : String = "increment";
        public static const DECREMENT : String = "decrement";

        private var _steps : int;

        public function SpinnerEvent(eventName : String) {
            super(eventName);
        }

        public function set steps(value:int) : void {
            _steps = value;
        }

        public function get steps() : int {
            return _steps;
        }
    }
}

```

次のステップでは、テスト アプリケーションのオートメーション サポートを実装します。

動的呼び出しを使用した Flex カスタム コントロールのテスト

Silk Test Workbench では、動的呼び出しを使用したカスタム コントロールの記録と再生のサポートが提供されており、これにより再生中にカスタム コントロールを操作できます。作業量が少なく済むこのアプローチは、テスト アプリケーションにおいて、Silk Test Workbench が公開しないカスタム コントロールのプロパティおよびメソッドにアクセスする場合に使用します。カスタム コントロールの開発者は、コントロールのテストを容易にすることのみを目的としたメソッドおよびプロパティをカスタム コントロールに追加することもできます。



注: 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

1. コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。
2. オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。
3. オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。
4. オブジェクトの複数の動的メソッドは invokeMethods メソッドを使用して呼び出します。
5. コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。
6. 動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。

例

この例では、以下の図に示すように、2つのボタンと1つのテキストフィールドを含む Spinner カスタム コントロールをテストします。



ユーザーは、**Down** をクリックしてテキスト フィールドに表示されている値を 1 減分させ、**Up** をクリックしてテキスト フィールドの値を 1 増分させることができます。

カスタム コントロールには、設定および取得が可能なパブリックの Value プロパティが用意されています。

Spinner の値を 4 に設定するには、以下のように入力します。

```
Dim spinner = Desktop.Find("//  
FlexBox[@className=customcontrols.Spinner]")  
spinner.SetProperty("Value", 4)
```

オートメーション サポートを使用したカスタム コントロールのテスト

カスタム コントロールに対して、特定のオートメーション サポートを作成できます。この追加のオートメーション サポートによって、記録のサポートおよびより強力な再生のサポートが提供されます。オートメーション サポートを作成するには、テスト アプリケーションを変更し、Open Agent を拡張する必要があります。

Silk Test Workbench でカスタム コントロールをテストする前に、以下のステップを実行します。

- テスト アプリケーションでのカスタム コントロールの定義
- オートメーション サポートの実装

テスト アプリケーションを変更してオートメーション サポートを組み込んだあと、以下のステップを実行します。

1. ビジュアル テストでは、追加の構成は必要ありません。必要に応じて、カスタム コントロールに対してビジュアル テストを記録および再生できます。
2. スクリプトの場合は、スクリプトを記録して、カスタム コントロールに合わせて手動で変更を加えます。
たとえば、以下のコードでは、オートメーションの委譲に実装されている「Increment」メソッドを使用して、Spinner の値を 3 増分する方法を示しています。

```
_desktop.TestObject("//FlexSpinner[@caption='index:1']").Invoke("Increment", 3)
```

以下の例は、Spinner の値を 3 に設定する方法を示しています。

```
_desktop.TestObject("//FlexSpinner[@caption='index:1']").SetProperty("Value", 3)
```

カスタム コントロールのオートメーション サポートの実装

カスタム コントロールをテストする前に、カスタム コントロールの ActionScript でオートメーション サポート（オートメーションの委譲）を実装し、テスト アプリケーションにコンパイルします。

以下の手順では、Flex のカスタム Spinner コントロールを使用して、カスタム コントロールのオートメーション サポートの実装方法を示します。Spinner カスタム コントロールは、以下のグラフィックに示すように、2つのボタンと1つのテキストフィールドを含んでいます。



ユーザーは、**Down** をクリックしてテキスト フィールドに表示されている値を 1 減分させ、**Up** をクリックしてテキスト フィールドの値を 1 増分させることができます。

カスタム コントロールには、設定および取得が可能なパブリックの Value プロパティが用意されています。

1. カスタム コントロールの ActionScript でオートメーション サポート（オートメーションの委譲）を実装します。

オートメーションの委譲の実装の詳細については、Adobe Live ドキュメント (http://livedocs.adobe.com/flex/3/html/help.html?content=functest_components2_14.html) を参照してください。

この例では、オートメーションの委譲によって、「increment」および「decrement」メソッドに対してサポートが追加されます。オートメーションの委譲のコード例は以下のとおりです。

```
package customcontrols
{
    import flash.display.DisplayObject;
    import mx.automation.Automation;
    import customcontrols.SpinnerEvent;
    import mx.automation.delegates.containers.BoxAutomationImpl;
    import flash.events.Event;
    import mx.automation.IAutomationObjectHelper;
    import mx.events.FlexEvent;
    import flash.events.IEventDispatcher;
    import mx.preloaders.DownloadProgressbar;
    import flash.events.MouseEvent;
    import mx.core.EventPriority;

    [Mixin]
    public class SpinnerAutomationDelegate extends BoxAutomationImpl
    {

        public static function init(root:DisplayObject) : void {
            // register delegate for the automation
            Automation.registerDelegateClass(Spinner, SpinnerAutomationDelegate);
        }

        public function SpinnerAutomationDelegate(obj:Spinner) {
            super(obj);
            // listen to the events of interest (for recording)
            obj.addEventListener(SpinnerEvent.DECREMENT, decrementHandler);
            obj.addEventListener(SpinnerEvent.INCREMENT, incrementHandler);
        }

        protected function decrementHandler(event : SpinnerEvent) : void {
            recordAutomatableEvent(event);
        }

        protected function incrementHandler(event : SpinnerEvent) : void {
            recordAutomatableEvent(event);
        }

        protected function get spinner() : Spinner {
            return uiComponent as Spinner;
        }

        //-----
        // override functions
    }
}
```

```

//-----
override public function get automationValue():Array {
    return [ spinner.Value.toString() ];
}

private function replayClicks(button : IEventDispatcher, steps : int) : Boolean
{
    var helper : IAutomationObjectHelper =
Automation.automationObjectHelper;
    var result : Boolean;
    for(var i:int; i < steps; i++) {
        helper.replayClick(button);
    }
    return result;
}

override public function replayAutomatableEvent(event:Event):Boolean {

    if(event is SpinnerEvent) {
        var spinnerEvent : SpinnerEvent = event as SpinnerEvent;
        if(event.type == SpinnerEvent.INCREMENT) {
            return replayClicks(spinner.upButton, spinnerEvent.steps);
        }
        else if(event.type == SpinnerEvent.DECREMENT) {
            return replayClicks(spinner.downButton, spinnerEvent.steps);
        }
        else {
            return false;
        }
    }
    else {
        return super.replayAutomatableEvent(event);
    }
}

// do not expose the child controls (i.e the buttons and the textfield) as
individual controls
override public function get numAutomationChildren():int {
    return 0;
}
}
}

```

2. Open Agent にオートメーションの委譲を導入するために、カスタム コントロールを記述する XML ファイルを作成します。

クラス定義ファイルには、インストルメント化されたすべての Flex コンポーネントについての情報が含まれています。このファイルでは、記録中にイベントを送信でき、再生中にイベントを受け取ることができるコンポーネントについての情報が提供されます。クラス定義ファイルには、サポートされているプロパティの定義も含まれています。

Spinner カスタム コントロールの XML ファイルは以下のようになります。

```

<?xml version="1.0" encoding="UTF-8"?>
<TypeInfoInformation>
    <ClassInfo Name="FlexSpinner" Extends="FlexBox">
        <Implementation
            Class="customcontrols.Spinner" />
        <Events>

```

```

    <Event Name="Decrement">
      <Implementation
        Class="customcontrols.SpinnerEvent"
        Type="decrement" />
      <Property Name="steps">
        <PropertyType Type="integer" />
      </Property>
    </Event>
    <Event Name="Increment">
      <Implementation
        Class="customcontrols.SpinnerEvent"
        Type="increment" />
      <Property Name="steps">
        <PropertyType Type="integer" />
      </Property>
    </Event>
  </Events>
  <Properties>
    <Property Name="lowerBound" accessType="read">
      <PropertyType Type="integer" />
    </Property>
    <Property Name="upperBound" accessType="read">
      <PropertyType Type="integer" />
    </Property>
    <!-- expose read and write access for the Value property -->
    <Property Name="Value" accessType="both">
      <PropertyType Type="integer" />
    </Property>
    <Property Name="stepSize" accessType="read">
      <PropertyType Type="integer" />
    </Property>
  </Properties>
</ClassInfo>
</TypeInfo>

```

3. サポートされている Flex コントロールのすべてのクラス、およびそのメソッドとプロパティを記述するすべての XML ファイルが格納されるフォルダに、カスタムコントロールの XML ファイルを配置します。

Silk Test には、サポートされている Flex コントロールのすべてのクラス、およびそのメソッドとプロパティを記述するいくつかの XML ファイルが含まれています。これらの XML ファイルは、<Silk Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment フォルダにあります。

独自の XML ファイルを提供する場合は、XML ファイルをこのフォルダにコピーする必要があります。Open Agent が起動して、Apache Flex のサポートを初期化する場合、このディレクトリの内容が読み込まれます。

Flex の Spinner サンプルコントロールをテストするには、CustomControls.xml ファイルをこのフォルダにコピーする必要があります。Open Agent が現在実行されている場合は、ファイルをフォルダにコピーしたあと、Open Agent を再起動します。

これで、カスタムコントロールを含むビジュアル テストまたはスクリプトを記録および再生できるようになりました。

Flex クラス定義ファイル

クラス定義ファイルには、インストール化されたすべての Flex コンポーネントについての情報が含まれています。このファイルでは、記録中にイベントを送信でき、再生中にイベントを受け取ることができるコンポーネントについての情報が提供されます。クラス定義ファイルには、サポートされているプロパティの定義も含まれています。

Silk Test には、Flex の共通コントロールおよび特殊化されたコントロールのすべてのクラス、イベント、およびプロパティを記述するいくつかの XML ファイルが含まれています。これらの XML ファイルは、`<Silk_Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment` フォルダにあります。

独自の XML ファイルを提供する場合は、XML ファイルをこのフォルダにコピーする必要があります。Silk Test のエージェントが起動して Apache Flex のサポートを初期化するとき、このディレクトリの内容が読み込まれます。

XML ファイルの基本的な構造は以下のとおりです。

```
<TypeInfoInformation>
<ClassInfo>
<Implementation />
<Events>
<Event />
...
</Events>
<Properties>
<Property />
...
</Properties>
</ClassInfo>
</TypeInfoInformation>
```

Apache Flex スクリプトのカスタマイズ

手動で Flex スクリプトをカスタマイズできます。Flex オブジェクトのプロパティに対して Verify 関数を使用して、手動で検証を挿入できます。各 Flex オブジェクトには、検証可能な一連のプロパティがあります。検証に使用できるプロパティのリストについては、「Flex クラス リファレンス」を参照してください。


1. Flex アプリケーションのテストを記録します。
2. カスタマイズするスクリプト ファイルを開きます。
3. 追加するコードを手動で入力します。

同一 Web ページ上の複数の Flex アプリケーションのテスト


同じ Web ページに複数の Flex アプリケーションが存在する場合、Silk Test Workbench は、Flex アプリケーションの ID またはアプリケーションの size プロパティを使用して、テスト対象アプリケーションを特定します。同じページに複数のアプリケーションが存在し、それらのサイズが異なる場合、Silk Test Workbench は、size プロパティを使用して操作実行対象のアプリケーションを特定します。追加の操作は必要ありません。

以下の場合、Silk Test Workbench は、JavaScript を使用して Flex アプリケーションの ID を検索し、操作実行対象のアプリケーションを特定します。

- 単一の Web ページ上に複数の Flex アプリケーションが存在する場合。
- これらのアプリケーションのサイズが同じである場合。

 **注:** この場合、ブラウザ マシンで JavaScript が有効になっていないと、スクリプト実行時にエラーが発生します。

1. JavaScript を有効にします。
2. Internet Explorer で、以下の手順を実行します。
 - a) ツール > インターネット オプション を選択します。
 - b) セキュリティ タブをクリックします。
 - c) レベルのカスタマイズ をクリックします。
 - d) スクリプト作成 セクションの アクティブ スクリプト で、有効にする をクリックして OK をクリックします。
3. 「Apache Flex アプリケーションのテスト」の手順に従います。

 **注:** Web ページにフレームが存在し、アプリケーションが同じサイズである場合、この方法は動作しません。

Adobe AIR のサポート

Silk Test Workbench がサポートする Adobe AIR でのテストは、Flex 4 コンパイラを使用してコンパイルされたアプリケーションのみです。サポートされているバージョンの詳細については、リリース ノートで最新の情報を確認してください。

Silk Test には、サンプルの Adobe AIR アプリケーションが含まれています。 <http://demo.borland.com/flex/SilkTest19.5/index.html> にあるサンプル アプリケーションにアクセスして、使用する Adobe AIR アプリケーションをクリックしてください。オートメーションあり、またはオートメーションなしのアプリケーションを選択できます。AIR アプリケーションを実行するには、Adobe AIR ランタイムをインストールする必要があります。

名前またはインデックスを使用する Flex の Select メソッドの概要

Flex の Select メソッドは、選択するコントロールの Name または Index を使用して記録できます。デフォルトで、Silk Test Workbench では、コントロールの名前を使用して Select メソッドが記録されます。ただし、コントロールのインデックスを使用して Select イベントを記録するように環境を変更したり、名前を使用した記録とインデックスを使用した記録を切り替えたりすることができます。

以下のコントロールでは、インデックスを使用して Select イベントを記録できます。

- FlexList
- FlexTree
- FlexDataGrid
- FlexAdvancedDataGrid
- FlexOLAPDataGrid
- FlexComboBox


デフォルト設定は、コントロールの名前を使用する ItemBasedSelection (Select イベント) です。インデックスを使用するには、IndexBasedSelection (SelectIndex イベント) を使用するように AutomationEnvironment を変更する必要があります。これらのクラスのいずれかの動作を変更するには、以下のコードを使用して FlexCommonControls.xml、AdvancedDataGrid.xml、または OLAPDataGrid.xml ファイルを変更する必要があります。これらの XML ファイルは、<Silk Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment フォルダにあります。対応する xml ファイルで、以下の変更を行います。

```
<ClassInfo Extends="FlexList" Name="FlexControlName"
EnableIndexBasedSelection="true" >
```



```
...  
</ClassInfo>
```

この変更では、FlexList::SelectIndex イベントの記録に IndexBasedSelection が使用されています。コードの EnableIndexBasedSelection= を false に設定するか、またはこのブール値を削除すると、記録で名前が使用される設定に戻ります (FlexList::Select イベント)。

 **注:** これらの変更内容を有効にするには、アプリケーションを再起動する必要があります。アプリケーションを再起動すると、Silk Test Agent も自動的に再起動されます。

FlexDataGrid コントロールでの項目の選択

FlexDataGrid コントロールの項目は、インデックス値または内容値を使用して選択します。

1. インデックス値を使用して FlexDataGrid コントロールの項目を選択するには、SelectIndex メソッドを使用します。
たとえば、FlexDataGrid.SelectIndex(1) のように入力します。
2. 内容値を使用して FlexDataGrid コントロールの項目を選択するには、Select メソッドを使用します。
必要な形式の文字列を使用して、選択する行を識別します。項目と項目の間は、縦線文字 (|) で区切る必要があります。少なくとも 1 つの項目を 2 つのアスタリスク (*) で囲む必要があります。これにより、クリックが実行される項目が識別されます。
構文は FlexDataGrid.Select("*Item1* | Item2 | Item3") です。

Flex アプリケーションのテストの有効化


Flex アプリケーションをテストに対して有効化するには、Apache Flex 開発者は Flex アプリケーションに以下のコンポーネントを組み込む必要があります。

- Apache Flex オートメーション パッケージ
- Silk Test オートメーション パッケージ

Apache Flex オートメーション パッケージ

開発者は、Flex オートメーション パッケージを使用して、オートメーション API を使用する Flex アプリケーションを作成できます。Flex オートメーション パッケージは、Adobe の Web サイト (<http://www.adobe.com>) からダウンロードできます。パッケージには、以下の内容が含まれています。

- オートメーション ライブラリ : automation.swc ライブラリおよび automation_agent.swc ライブラリは、Flex フレームワーク コンポーネントの委譲の実装です。automation_agent.swc ファイルおよび関連するリソースバンドルは、汎用的なエージェント メカニズムです。Silk Test Agent などのエージェントは、これらのライブラリの上に構築されます。
- サンプル

 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、Flex オートメーション API の typekey ステートメントでは、すべてのキーはサポートされません。テキスト入カステートメントを使用してこの問題を解決できます。Flex オートメーション API の詳細については、『*Apache Flex リリース ノート*』を参照してください。

Silk Test オートメーション パッケージ

Silk Test の Open Agent は、Apache Flex オートメーション エージェント ライブラリを使用しています。FlexTechDomain.swc ファイルに、Silk Test 固有の実装が含まれています。

以下のいずれかの方法を使用して、アプリケーションをテストに対して有効化できます。

- Flex アプリケーションへのオートメーション パッケージのリンク
- 実行時の読み込み

Flex アプリケーションへのオートメーション パッケージのリンク

テストする予定の Flex アプリケーションを事前にコンパイルする必要があります。機能テスト クラスは、コンパイル時にアプリケーションに埋め込まれ、アプリケーションは実行時に自動テストに関する外部依存関係を持ちません。

コンパイル時にアプリケーションの SWF ファイルに機能テスト クラスを埋め込むと、SWF ファイルのサイズが大きくなります。SWF ファイルのサイズが重要でない場合は、機能テストと展開に同じ SWF ファイルを使用します。SWF ファイルのサイズが重要である場合は、2 つの SWF ファイルを生成します。1 つは機能テスト クラスが埋め込まれたファイル、もう 1 つは機能テスト クラスが埋め込まれていないファイルです。展開には、テスト クラスが埋め込まれていない SWF ファイルを使用します。

include-libraries コンパイラ オプションを指定してテストのために Flex アプリケーションを事前にコンパイルする場合は、以下のファイルを参照します。

- automation.swc
- automation_agent.swc
- FlexTechDomain.swc
- automation_charts.swc (アプリケーションでグラフおよび Flex 2.0 を使用する場合のみインクルード)
- automation_dmv.swc (アプリケーションでグラフおよび Flex 3.x 以降を使用する場合にインクルード)
- automation_flasflexkit.swc (アプリケーションで埋め込みの Flash コンテンツを使用する場合にインクルード)
- automation_spark.swc (アプリケーションで新しい Flex 4.x コントロールを使用する場合にインクルード)
- automation_air.swc (アプリケーションが AIR アプリケーションである場合にインクルード)
- automation_airspace.swc (アプリケーションが AIR アプリケーションであり、新しい Flex 4.x コントロールを使用する場合にインクルード)

Flex アプリケーションの最終リリース バージョンを作成する場合は、これらの SWC ファイルへの参照なしでアプリケーションを再コンパイルします。オートメーション SWC ファイルの使用の詳細については、『*Apache Flex リリース ノート*』を参照してください。

アプリケーションをサーバーに展開しないで、ファイル プロトコルを使用して要求したり、Apache Flex Builder 内で実行したりする場合は、各 SWF ファイルをローカルの信頼済みサンドボックスに組み込む必要があります。このためには、追加の構成情報が必要です。コンパイラの構成ファイルを変更するか、またはコマンド ライン オプションを使用して、構成情報を追加します。




注: Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、オートメーション コードを使用してアプリケーションがコンパイルされ、連続的に SWF ファイルが読み込まれる場合、メモリ リークが発生して、最終的にアプリケーションでメモリが不足します。Flex Control Explorer サンプル アプリケーションは、この問題の影響を受けます。回避策として、Explorer が読み込むアプリケーションの SWF ファイルをオートメーション ライブラリを使用してコンパイルしない方法があります。たとえば、Explorer のメイン アプリケーションのみをオートメーション ライブラリを使用してコンパイルします。SWFLoader の代わりにモジュール ローダーを使用する方法もあります。Flex オートメーション API の詳細については、『*Apache Flex リリース ノート*』を参照してください。

テストのための Flex アプリケーションの事前コンパイル

アプリケーションをテスト用に事前コンパイルするか、または実行時の読み込みを使用することによって、アプリケーションをテストに対して有効化できます。

1. 以下のコードを構成ファイルに追加することによって、コンパイラの構成ファイルに automation.swc、automation_agent.swc、および FlexTechDomain.swc ライブラリをインクルードします。

```
<include-libraries>
...
<library>/libs/automation.swc</library>
<library>/libs/automation_agent.swc</library>
<library>pathinfo/FlexTechDomain.swc</library>
</include-libraries>
```

 **注:** アプリケーションでグラフを使用する場合は、automation_charts.swc ファイルも追加する必要があります。


2. コマンドラインコンパイラで include-libraries コンパイラ オプションを使用して、automation.swc、automation_agent.swc、および FlexTechDomain.swc ライブラリの場所を指定します。構成ファイルは以下の場所にあります。


Apache Flex 2 SDK – <flex_installation_directory>/frameworks/flex-config.xml

Apache Flex データ サービス – <flex_installation_directory>/flex/WEB-INF/flex/flex-config.xml

以下の例では、automation.swc ファイルと automation_agent.swc ファイルがアプリケーションに追加されています。

```
mxmmlc -include-libraries+=../frameworks/libs/automation.swc;../frameworks/libs/automation_agent.swc;pathinfo/FlexTechDomain.swc MyApp.mxml
```

 **注:** コマンドラインで include-libraries オプションを明示的に設定すると、既存のライブラリに対して追加されるのではなく、既存のライブラリが上書きされます。コマンドラインで include-libraries オプションを使用して automation.swc ファイルと automation_agent.swc ファイルを追加する場合は、+= 演算子を使用します。これにより、インクルードされる既存のライブラリが上書きされるのではなく、インクルードされる既存のライブラリに対して追加されます。

 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、オートメーションコードを使用してアプリケーションがコンパイルされ、連続的に SWF ファイルが読み込まれる場合、メモリリークが発生して、最終的にアプリケーションでメモリが不足します。Flex Control Explorer サンプルアプリケーションは、この問題の影響を受けます。回避策として、Explorer が読み込むアプリケーションの SWF ファイルをオートメーションライブラリを使用してコンパイルしない方法があります。たとえば、Explorer のメインアプリケーションのみをオートメーションライブラリを使用してコンパイルします。SWFLoader の代わりにモジュールローダーを使用する方法もあります。Flex オートメーション API の詳細については、『Apache Flex リリースノート』を参照してください。

実行時の読み込み

Silk Test Flex オートメーションランチャを使用して、実行時に Flex オートメーションサポートを読み込むことができます。このアプリケーションは、オートメーションライブラリを使用してコンパイルされており、SWFLoader クラスを使用してユーザーのアプリケーションを読み込みます。これにより、SWF ファイルにオートメーションライブラリをコンパイルしなくても、アプリケーションが自動的にテストに対して有効化されます。Silk Test Flex オートメーションランチャは、HTML および SWF のファイル形式で利用できます。

制限事項

- Flex オートメーションランチャアプリケーションは、自動的にルートアプリケーションとなります。ユーザーのアプリケーションをルートアプリケーションにする必要がある場合は、Silk Test Flex オートメーションランチャを使用してオートメーションサポートを読み込むことができません。
- 外部ライブラリを読み込むアプリケーション（他の SWF ファイルライブラリを読み込むアプリケーション）をテストするには、自動テストに特別な設定が必要です。実行時に読み込まれるライブラリ（ランタイム共有ライブラリ（RSL）を含む）は、読み込むアプリケーションの ApplicationDomain に読み込まれる必要があります。アプリケーションで使用される SWF ファイルが異なるアプリケーションドメインに読み込まれた場合、自動テストの記録と再生が正しく動作しません。以下に、同じ ApplicationDomain に読み込まれるライブラリの例を示します。

```
import flash.display.*;

import flash.net.URLRequest;

import flash.system.ApplicationDomain;

import flash.system.LoaderContext;

var ldr:Loader = new Loader();

var urlReq:URLRequest = new URLRequest("RuntimeClasses.swf");

var context:LoaderContext = new LoaderContext();

context.applicationDomain = ApplicationDomain.currentDomain;

loader.load(request, context);
```

実行時読み込み

1. Silk Test Automation SDK Flex <バージョン> FlexAutomationLauncher ディレクトリの内容を、テストする Flex アプリケーションのディレクトリにコピーします。
2. Windows Explorer で FlexAutomationLauncher.html を開き、ファイルパスへの接尾辞として以下のパラメータを追加します。

```
?automationurl=YourApplication.swf
```

YourApplication.swf は Flex アプリケーションの SWF ファイルの名前です。

3. ファイルパスへの接頭辞として file:/// を追加します。
たとえば、ファイルの URL に ?automationurl=explorer.swf などのパラメータが含まれている場合は、以下のように入力します。を設定して、インストールするコンポーネントを指定できます。

```
file:///C:/Program%20Files/Silk/Silk Test/ng/sampleapplications/Flex/3.2/
FlexControlExplorer32/FlexAutomationLauncher.html?automationurl=explorer.swf
```

コマンドラインを使用した構成情報の追加

コマンドラインコンパイラを使用して automation.swc、automation_agent.swc、および FlexTechDomain.swc ライブラリの場所を指定するには、include-libraries コンパイラオプションを使用します。

次の例では、automation.swc ファイルと automation_agent.swc ファイルがアプリケーションに追加されます。

```
mxmmlc -include-libraries+=../frameworks/libs/automation.swc;../frameworks/libs/automation_agent.swc;pathinfo/FlexTechDomain.swc MyApp.mxml
```



注: アプリケーションでグラフを使用する場合は、include-libraries コンパイラ オプションに automation_charts.swc ファイルも追加する必要があります。

コマンドラインで include-libraries オプションを明示的に設定すると、既存のライブラリに対して追加されるのではなく、既存のライブラリが上書きされます。コマンドラインで include-libraries オプションを使用して automation.swc ファイルと automation_agent.swc ファイルを追加する場合は、+= 演算子を使用します。これにより、インクルードされる既存のライブラリが上書きされるのではなく、インクルードされる既存のライブラリに対して追加されます。

Flex Builder プロジェクトに自動テストサポートを追加するには、include-libraries コンパイラ オプションに automation.swc および automation_agent.swc ライブラリも追加する必要があります。

Flex アプリケーションにパラメータを渡す

以下の手順に従って、Flex アプリケーションにパラメータを渡すことができます。

実行する前に Flex アプリケーションにパラメータを渡す

オートメーション ライブラリを使用して、実行する前に Flex アプリケーションにパラメータを渡すことができます。

1. 適切なオートメーション ライブラリを使用して、アプリケーションをコンパイルします。
2. パラメータの指定には、通常どおり標準的な Flex のメカニズムを使用します。

Flex オートメーションランチャを使用して、実行時に Flex アプリケーションにパラメータを渡す

このタスクを開始する前に、実行時の読み込みに対応するようにアプリケーションを準備します。

1. FlexAutomationLauncher.html ファイルを開くか、または例として FlexAutomationLauncher.html を使用してファイルを作成します。
2. 以下のセクションに移動します。

```
<script language="JavaScript" type="text/javascript">
    AC_FL_RunContent(eef
        "src", "FlexAutomationLauncher",
        "width", "100%",
        "height", "100%",
        "align", "middle",
        "id", "FlexAutomationLauncher",
        "quality", "high",
        "bgcolor", "white",
        "name", "FlexAutomationLauncher",
        "allowScriptAccess","sameDomain",
        "type", "application/x-shockwave-flash",
```

```
        "pluginspage", "http://www.adobe.com/go/getflashplayer",
        "flashvars", "yourParameter=yourParameterValue"+
"&automationurl=YourApplication.swf"

    );
</script>
```



注: 「src」、 「id」、 および 「name」 の 「FlexAutomationLauncher」 の値は変更しないでください。

3. 「yourParameter=yourParameterValue」 に、独自のパラメータを追加します。
4. 「& automationurl=YourApplication.swf」 の値として、テストする Flex アプリケーションの名前を渡します。
5. ファイルを保存します。

テスト可能な Flex アプリケーションの作成

Flex 開発者は、Flex アプリケーションを可能なかぎりテストしやすくするためのテクニックを利用できます。以下のテクニックがあります。

- オブジェクトに対するわかりやすい ID の指定
- オブジェクトの重複の回避

オブジェクトに対するわかりやすい ID の指定

テストしやすいアプリケーションを作成するには、スクリプト内でオブジェクトを識別しやすくする必要があります。テストするすべてのコントロールに対して、わかりやすい文字列を使用した ID プロパティの値を設定できます。

オブジェクトに対してわかりやすい ID を指定するには：

- テスト可能なすべての MXML コンポーネントに対して ID を指定して、その Flex コントロールの参照時にテスト スクリプトで一意的 ID が使用できるようにします。
- これらの ID は、ユーザーがテスト スクリプト内でそのオブジェクトを容易に識別できるように、可能なかぎり人間が理解しやすい文字列にします。たとえば、TabNavigator 内の Panel コンテナの id プロパティは、panel1 や p1 ではなく submit_panel とします。

Silk Test Workbench を使用する場合、id や childIndex などの特定のタグに基づいて、オブジェクトに対して自動的に名前が設定されます。id プロパティに値がない場合、Silk Test Workbench では、childIndex プロパティなどの他のプロパティが使用されます。id プロパティに値を割り当てると、テスト スクリプトを読みやすくすることができます。

オブジェクトの重複の回避

自動エージェントの処理は、実行中にオブジェクト インスタンスの一部のプロパティが変更されないことを前提としています。実行時に Silk Test Workbench によってオブジェクト名として使用されている Flex コンポーネント プロパティを変更すると、予期しない結果が発生する可能性があります。たとえば、automationName プロパティのない Button コントロールを作成し、最初は label プロパティに値を設定しないで、その後、label プロパティに値を設定した場合、問題が発生することがあります。この場合、Silk Test Workbench では、automationName プロパティが設定されていない場合は Button コントロールを識別するためにコントロールの label プロパティの値が使用されます。あとから label プロパティの値を設定したり、既存の label の値を変更すると、Silk Test Workbench ではこのオブジェクトを新しいオブジェクトとして識別し、既存のオブジェクトを参照しなくなります。

重複オブジェクトを回避するには：

- エージェントにおいて、オブジェクトの識別にどのプロパティが使用されているかを理解し、実行時にそれらのプロパティを変更しないようにします。
- 記録されたスクリプトに含まれているすべてのオブジェクトに対して、人間が理解しやすい一意の id プロパティまたは automationName プロパティを設定します。


Apache Flex アプリケーションのカスタム属性

Apache Flex アプリケーションは、あらかじめ定義されたプロパティ automationName を使用して、次のように Apache Flex コントロールに対して安定した識別子を指定します。

```
<?xml version="1.0" encoding="utf-8"?>
  <s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" width="400" height="300">
    <fx:Script>
      ...
    </fx:Script>
    <s:Button x="247" y="81" label="Button" id="button1" enabled="true"
click="button1_clickHandler(event)"
      automationName="AID_buttonRepeat"/>
    <s:Label x="128" y="123" width="315" height="18" id="label1" verticalAlign="middle"
      text="awaiting your click" textAlign="center"/>
  </s:Group>
```

Apache Flex アプリケーションのロケータは次のようになります。


```
...//SparkApplication//SparkButton[@caption='AID_buttonRepeat'
```

 **注目:** Apache Flex アプリケーションの場合、Silk Test Workbench では automationName はロケータ属性 caption に常にマップされます。automationName 属性が指定されていない場合、Silk Test Workbench は属性 ID をロケータ属性 caption にマップします。

Flex の AutomationName プロパティと AutomationIndex プロパティ

Flex オートメーション API には、automationName プロパティと automationIndex プロパティが用意されています。automationName を指定すると、Silk Test Workbench では、記録されたウィンドウ宣言の名前としてこの値が使用されます。わかりやすい名前を指定すると、そのオブジェクトを Silk Test Workbench で識別しやすくなります。ベストプラクティスとして、アプリケーションのテストに含まれているすべてのオブジェクトの automationName プロパティに値を設定することをお勧めします。

automationIndex プロパティを使用して、オブジェクトに対して一意のインデックス値を割り当てます。たとえば、2つのオブジェクトが同じ名前を共有している場合は、インデックス値を割り当てて、2つのオブジェクトを識別します。

 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、オートメーションコードを使用してアプリケーションがコンパイルされ、連続的に SWF ファイルが読み込まれる場合、メモリリークが発生して、最終的にアプリケーションでメモリが不足します。Flex Control Explorer サンプルアプリケーションは、この問題の影響を受けます。回避策として、Explorer が読み込むアプリケーションの SWF ファイルをオートメーションライブラリを使用してコンパイルしない方法があります。たとえば、Explorer のメインアプリケーションのみをオートメーションライブラリを使用してコンパイルします。SWFLoader の代わりにモジュールローダーを使用する方法もあります。Flex オートメーション API の詳細については、『Apache Flex リリースノート』を参照してください。

Flex クラス定義ファイル

クラス定義ファイルには、インストルメント化されたすべての Flex コンポーネントについての情報が含まれています。このファイルでは、記録中にイベントを送信でき、再生中にイベントを受け取ることができるコンポーネントについての情報が提供されます。クラス定義ファイルには、サポートされているプロパティの定義も含まれています。

Silk Test には、Flex の共通コントロールおよび特殊化されたコントロールのすべてのクラス、イベント、およびプロパティを記述するいくつかの XML ファイルが含まれています。これらの XML ファイルは、`<Silk_Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment` フォルダにあります。

独自の XML ファイルを提供する場合は、XML ファイルをこのフォルダにコピーする必要があります。Silk Test のエージェントが起動して Apache Flex のサポートを初期化するとき、このディレクトリの内容が読み込まれます。

XML ファイルの基本的な構造は以下のとおりです。

```
<TypeInfo>
<ClassInfo>
<Implementation />
<Events>
<Event />
...
</Events>
<Properties>
<Property />
...
</Properties>
</ClassInfo>
</TypeInfo>
```

Flex の automationName プロパティの設定

automationName プロパティは、テストに表示されるコンポーネント名を定義します。このプロパティのデフォルト値は、コンポーネントの種類に応じて異なります。たとえば、Button コントロールの automationName は、Button コントロールのラベルです。automationName がコントロールの id プロパティと同じ場合もありますが、常に同じであるわけではありません。

一部のコンポーネントでは、automationName プロパティの値は、Flex によってそのコンポーネントを認識しやすい属性に設定されています。これにより、テスト担当者は、テストでコンポーネントを認識しやすくなります。通常、テスト担当者は、アプリケーションの基になるソースコードにアクセスできないため、コントロールの表示されるプロパティによってそのコントロールを認識できるようにすることは有用です。たとえば、「Process Form Now」というラベルが設定された Button は、テストで FlexButton("Process Form Now") と表示されます。

新しいコンポーネントを実装する場合や、既存のコンポーネントから派生する場合は、automationName プロパティのデフォルト値をオーバーライドできます。たとえば、UIComponent では、automationName の値は、デフォルトでコンポーネントの id プロパティに設定されます。ただし、一部のコンポーネントでは、独自の方法を使用して値が設定されます。たとえば、Flex Store サンプルアプリケーションでは、コンテナを使用して製品のサムネイルが作成されています。コンテナのデフォルトの automationName はコンテナの id プロパティと同じ値となるため、あまり役立ちません。そのため、Flex Store では、製品のサムネイルを生成するカスタムコンポーネントで明示的に automationName を製品名に設定して、アプリケーションをテストしやすくしています。

例

以下の CatalogPanel.mxml カスタム コンポーネントの例では、automationName プロパティの値をカタログに表示される項目名に設定しています。これにより、デフォルトのオートメーション名を使用するよりもサムネイルを認識しやすくなります。

```
thumbs[i].automationName = catalog[i].name;
```

例

以下の例では、ComboBox コントロールの automationName プロパティを「Credit Card List」に設定しています。このように設定すると、通常、テストツールでは、スクリプトにおいて id プロパティではなく「Credit Card List」を使用して ComboBox が識別されます。

```
<?xml version="1.0"?>
<!-- at/SimpleComboBox.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:Script>
    <![CDATA[
      [Bindable]
      public var cards: Array = [
        {label:"Visa", data:1},
        {label:"MasterCard", data:2},
        {label:"American Express", data:3}
      ];

      [Bindable]
      public var selectedItem:Object;
    ]>
  </mx:Script>
  <mx:Panel title="ComboBox Control Example">
    <mx:ComboBox id="cb1" dataProvider="{cards}"
      width="150"
      close="selectedItem=ComboBox(event.target).selectedItem"
      automationName="Credit Card List"
    />
    <mx:VBox width="250">
      <mx:Text width="200" color="blue" text="Select a type of credit
card." />
      <mx:Label text="You selected: {selectedItem.label}"/>
      <mx:Label text="Data: {selectedItem.data}"/>
    </mx:VBox>
  </mx:Panel>
</mx:Application>
```

automationName プロパティの値を設定すると、オブジェクト名が実行時に変更されないことが保証されます。このことは、予期しない結果の回避に役立ちます。

automationName プロパティの値を設定すると、テストでは、デフォルト値ではなく、その値が使用されます。たとえば、Silk Test Workbench では、デフォルトで、スクリプトにおいて Button コントロールの label プロパティがボタンの名前として使用されます。この場合、ラベルが変更されると、スクリプトが動作しなくなります。automationName プロパティの値を明示的に設定することによって、このような事態を回避できます。

ラベルがなく、アイコンがあるボタンは、インデックス番号によって記録されます。この場合は、automationName プロパティをわかりやすい文字列に設定して、テスト担

当者がスクリプトでボタンを認識できるようにします。automationName プロパティの値を設定したあとは、コンポーネントのライフ サイクル全体を通して値を変更しないでください。項目レンダラでは、automationName プロパティではなく automationValue プロパティを使用します。automationValue プロパティを使用するには、createAutomationIDPart() メソッドをオーバーライドして、automationName プロパティに割り当てる新しい値を返します。以下に例を示します。

```
<mx:List xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:Script>

    import mx.automation.IAutomationObject;
    override public function
    createAutomationIDPart(item:IAutomationObject):Object {
      var id:Object = super.createAutomationIDPart(item);
      id["automationName"] = id["automationIndex"];
      return id;
    }

  </mx:Script>
</mx:List>
```

このテクニックを使用して、任意のコンテナまたはリスト形式コントロールの子にインデックス値を追加します。子が自分自身のインデックスを指定する方法はありません。

名前またはインデックスを使用するように Flex の Select メソッドを設定

Flex の Select メソッドは、選択するコントロールの Name または Index を使用して記録できます。デフォルトで、Silk Test では、コントロールの名前を使用して Select メソッドが記録されます。ただし、コントロールのインデックスを使用して Select イベントを記録するように環境を変更したり、名前を使用した記録とインデックスを使用した記録を切り替えたりすることができます。

1. インデックスを使用するように変更するクラスを特定します。

以下のコントロールでは、インデックスを使用して Select イベントを記録できます。

- FlexList
- FlexTree
- FlexDataGrid
- FlexOLAPDataGrid
- FlexComboBox
- FlexAdvancedDataGrid

2. 変更するクラスに関連する XML ファイルを特定します。

上記のコントロールに関連する XML ファイルには、FlexCommonControls.xml、AdvancedDataGrid.xml、または OLAPDataGrid.xml があります。

3. 変更するクラスに関連する XML ファイルに移動します。


XML ファイルは、<Silk Test_install_directory>%ng%agent%plugins
%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment フォルダにあります。

4. 対応する XML ファイルで、以下の変更を行います。

```
<ClassInfo Extends="FlexList" Name="FlexControlName"
EnableIndexedSelection="true" >
...
</ClassInfo>
```

たとえば、「FlexControlName」として「FlexList」を使用し、FlexCommonControls.xml ファイルを変更できます。

この変更では、FlexList::SelectIndex イベントの記録に IndexBasedSelection が使用されています。

 **注:** コードの EnableIndexBasedSelection= を false に設定するか、またはこのブール値を削除すると、記録で名前が使用される設定に戻ります (FlexList::Select イベント)。

5. これらの変更内容を有効にするには、Flex アプリケーションおよび Open Agent を再起動します。

Flex コンテナのコーディング

コンテナは、ユーザー対話（ユーザーが Accordion コンテナの次のページに移動したなど）の記録、およびテスト スクリプト内でのコントロールに対する一意の場所の提供の両方の目的で使用されるため、他の種類のコントロールとは異なります。

オートメーション階層におけるコンテナの追加と削除

通常、自動テスト機能のスクリプトでは、ネストされたコンテナについての詳細情報は少なく抑えられます。テストの結果やコントロールの識別に影響がないコンテナは、スクリプトから削除されます。削除対象となるコンテナは、HBox、VBox、Canvas などの、レイアウトの目的でのみ使用されるコンテナです。ただし、ViewStack、TabNavigator、Accordion などの複数ビュー ナビゲータ コンテナで使用されている場合は削除されません。このような場合、コンテナはオートメーション階層に追加されて、ナビゲーションに使用されます。

多くの複合コンポーネントでは、Canvas や VBox などのコンテナを使用して、子が整理されます。これらのコンテナは、アプリケーション上では視覚的な効果を持ちません。この結果、これらのコンテナでは、ユーザー操作は実行されず、操作を視覚的に記録する必要もないため、通常、これらのコンテナはテストから除外されます。テストからコンテナを除外することによって、関連するテスト スクリプトが簡潔になり、読みやすくなります。

コンテナを記録から除外するには、コンテナの showInAutomationHierarchy プロパティを false に設定します（子は除外されません）。このプロパティは、UIComponent クラスによって定義されているため、UIComponent のサブクラスであるすべてのコンテナにこのプロパティが存在します。階層で表示されないコンテナの子は、階層内でそのコンテナの次に上位の親の子として表示されます。

showInAutomationHierarchy プロパティのデフォルト値は、コンテナの種類に応じて異なります。Panel、Accordion、Application、DividedBox、Form などのコンテナではデフォルト値は true であり、Canvas、HBox、VBox、FormItem などのコンテナではデフォルト値は false です。

以下の例では、VBox コンテナがテスト スクリプトの階層に組み込まれています。

```
<?xml version="1.0"?>
<!-- at/NestedButton.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:Panel title="ComboBox Control Example">
<mx:HBox id="hb">
<mx:VBox id="vb1" showInAutomationHierarchy="true">
<mx:Canvas id="c1">
<mx:Button id="b1" automationName="Nested Button 1" label="Click Me" />
</mx:Canvas>
</mx:VBox>
<mx:VBox id="vb2" showInAutomationHierarchy="true">
<mx:Canvas id="c2">
<mx:Button id="b2" automationName="Nested Button 2" label="Click Me 2" />
</mx:Canvas>
</mx:VBox>
</mx:HBox>
</mx:Panel>
</mx:Application>
```

複数ビュー コンテナ

TabNavigator や Accordion コンテナなどの複数ビュー コンテナ内の複数のタブに同じラベルを使用しないでください。同じラベルを使用することもできますが、このことは、通常、推奨 UI 設計プラクティスとして推奨されていません。このようにすると、テスト環境においてコントロールの識別に問題が発生することがあります。

Flex 自動テスト ワークフロー

Flex アプリケーションのテストの Silk Test Workbench ワークフローは、以下のとおりです。

- 自動テストの初期化
- 自動テストの記録
- 自動テストの再生

Flex 自動テストの初期化

ユーザーが Flex アプリケーションを起動すると、以下の初期化イベントが発生します。

1. オートメーション初期化コードによって、コンポーネントの委譲クラスがコンポーネントのクラスに関連付けられます。
2. コンポーネントの委譲クラスは、IAutomationObject インターフェイスを実装します。
3. AutomationManager のインスタンスがミックスインの init() メソッドで作成されます。(AutomationManager はミックスインです。)
4. SystemManager によってアプリケーションが初期化されます。コンポーネント インスタンスおよび対応する委譲インスタンスが作成されます。委譲インスタンスによって、目的のイベントに対するイベント リスナーが追加されます。
5. Silk Test Workbench FlexTechDomain はミックスインです。FlexTechDomain の init() メソッドで、FlexTechDomain が SystemManager.APPLICATION_COMPLETE イベントに登録されます。イベントを受信すると、FlexTechDomain インスタンスが作成されます。
6. FlexTechDomain インスタンスが、同じマシン上の記録および再生機能に登録する Silk Test Agent に TCP/IP ソケット経由で接続します。
7. FlexTechDomain は、自動環境についての情報を要求します。この情報は XML ファイルに格納され、Silk Test Agent から FlexTechDomain に転送されます。

Flex 自動テストの記録

ユーザーが Silk Test Workbench で Flex アプリケーションの新しいテストを記録すると、以下のイベントが発生します。

1. Silk Test Workbench によって Silk Test Agent が呼び出されて、記録が開始されます。Agent は、このコマンドを FlexTechDomain インスタンスに転送します。
2. FlexTechDomain は、beginRecording() を呼び出すことによって、AutomationManager に対して記録の開始を通知します。AutomationManager は、SystemManager からの AutomationRecordEvent.RECORD イベントに対するリスナーを追加します。
3. ユーザーがアプリケーションを操作します。たとえば、ユーザーが Button コントロールをクリックしたとします。
4. ButtonDelegate.clickEventHandler() メソッドによって、プロパティとしてクリック イベントと Button のインスタンスが指定された AutomationRecordEvent イベントがディスパッチされます。
5. AutomationManager は、XML 環境情報に基づいて、クリック イベントのどのプロパティを格納するかを決定します。値が適切な型または書式に変換されます。記録イベントがディスパッチされます。
6. FlexTechDomain イベントハンドラがイベントを受信します。AutomationManager.createID() メソッドが呼び出されて、ボタンの AutomationID オブジェクトが作成されます。このオブジェクトは、オブジェクト識別用の構造体を提供します。AutomationID 構造体は、AutomationIDParts の配列になっています。AutomationIDParts は、IAutomationObject を使用して作成されます。(Button コントロールの UIComponent.id、automationName、automationValue、childIndex、および label プロ

パティが読み込まれて、オブジェクトに格納されます。XML 情報に、label プロパティを Button の識別に使用できることが指定されているため、label プロパティが使用されます。)

7. FlexTechDomain は、AutomationManager.getParent() メソッドを使用して、Button の論理的な親を取得します。アプリケーション レベルまでの各レベルで、親コントロールの AutomationIDParts オブジェクトが収集されます。
8. すべての AutomationIDParts が AutomationID オブジェクトの一部として組み込まれます。
9. FlexTechDomain は、Silk Test Workbench への呼び出しでこの情報を送信します。
10. ユーザーが記録を停止すると、FlexTechDomain.endRecording() メソッドが呼び出されます。

Flex 自動テストの再生

ユーザーが Silk Test Workbench で **再生** ボタンをクリックすると、以下のイベントが発生します。

1. 各スクリプト呼び出しにおいて、Silk Test Workbench は Silk Test Agent に接続し、実行されるスクリプト呼び出しの情報を送信します。この情報には、完全なウィンドウ宣言、イベント名、およびパラメータが含まれています。
2. Silk Test Agent は、その情報を FlexTechDomain に転送します。
3. FlexTechDomain は、ウィンドウ宣言情報と共に AutomationManager.resolveIDToSingleObject を使用します。AutomationManager は、説明情報 (automationName、automationIndex、id など) に基づいて、解決したオブジェクトを返します。
4. Flex コントロールが解決されると、FlexTechDomain は AutomationManager.replayAutomatableEvent() を呼び出して、イベントを再生します。
5. AutomationManager.replayAutomatableEvent() メソッドによって、委譲クラスの IAutomationObject.replayAutomatableEvent() メソッドが呼び出されます。委譲では、IAutomationObjectHelper.replayMouseEvent() メソッド (または replayKeyboardEvent() などの他のいずれかの再生メソッド) を使用してイベントが再生されます。
6. スクリプトに検証がある場合、FlexTechDomain は AutomationManager.getProperties() を呼び出して、検証する必要がある値にアクセスします。

Apache Flex アプリケーションのスタイル

Apache Flex 3.x で開発されたアプリケーションについて、Silk Test Workbench ではスタイルとプロパティを区別しません。この結果、スタイルはプロパティとして公開されます。ただし、Apache Flex 4.x の Spark という接頭辞が付いているすべての新しい Flex コントロール (SparkButton など) では、スタイルがプロパティとして公開されません。この結果、Flex 4.x コントロールの GetProperty() メソッドおよび GetPropertyList() メソッドでは color や fontSize などのスタイルが返されず、text や name などのプロパティのみが返されます。

GetStyle(string styleName) メソッドは、スタイルの値を文字列として返します。どのようなスタイルが存在するかを確認するには、次の Adobe ヘルプを参照してください: http://help.adobe.com/ja_JP/FlashPlatform/reference/actionsript/3/package-detail.html。

スタイルが設定されていない場合は、再生中に StyleNotSetException が発生します。

FlexTree などの Flex 3.x コントロールでは、GetProperty() を使用してスタイルを取得できます。GetStyle() を使用することもできます。Flex 3.x コントロールでは、GetProperty() メソッドと GetStyle() メソッドの両方が動作します。

色スタイルの計算

Flex では、色は数値として表されます。色は、以下の式を使用して計算できます。

```
red*65536 + green*256 + blue
```

例

以下のスクリプト例では、Spark アプリケーションの ButtonBar がフォント サイズ 12 を使用しているかどうかを検証しています。

```
Imports SilkTest.Ntf.Flex

Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        Dim Application As SparkApplication
        Dim ButtonBar As SparkButtonBar
        Application = _desktop.Find( "/BrowserApplication//
        BrowserWindow//
        SparkApplication" )
        ButtonBar = Application.SparkButtonBar()

        Workbench.Verify(ButtonBar.GetStyle( "fontSize" ), "12" )
    End Sub
End Module
```

Adobe Flash Player のセキュリティ制約に対応するための Flex アプリケーションの構成

Adobe Flash Player 10 では、セキュリティ モデルが以前のバージョンから変更されています。Flash Player を使用するテストを記録する場合、記録は想定どおりに動作します。ただし、テストを再生する場合は、特定の状況で高レベルのクリックが行われると、予期しない結果が発生します。たとえば、**ファイル参照** ダイアログ ボックスをプログラムから開くことができません。このシナリオの再生を試みると、セキュリティ制約が原因でテストに失敗します。

このセキュリティ制約を回避するには、ダイアログ ボックスを開くボタンに対して低レベルのクリックを実行します。低レベルのクリックを作成するには、Click メソッドにパラメータを追加します。

たとえば、SparkButton::Click() の代わりに SparkButton::Click(MouseButton.Left) を使用します。パラメータを指定しない Click() は高レベルのクリックとして再生され、パラメータを指定したクリック (ボタンなど) は低レベルのクリックとして再生されます。

1. Flash Player を使用するテストを記録します。
2. Click メソッドに移動して、パラメータを追加します。
たとえば、**ファイルを開く** ダイアログ ボックスを開くには、以下のように指定します：

```
SparkButton("@caption='Open File Dialog...']").Click(MouseButton.Left)
```

。テストを再生すると、想定どおりに動作します。

Apache Flex アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Flex アプリケーションがサポートする属性は次のとおりです。

- automationName
- caption (automationName と同様)

- automationClassName (FlexButton など)
- className (実装クラスの完全修飾名。 mx.controls.Button など)
- automationIndex (FlexAutomation のビューでのコントロールのインデックス。 index:1 など)
- index (automationIndex と同様。ただし、接頭辞はなし。 1 など)
- id (コントロールの ID)
- windowId (id と同様)
- label (コントロールのラベル)
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

Silk Test Workbench が Apache Flex コントロールを認識できない理由

Web サーバーを通してアクセスしている Apache Flex アプリケーションのコントロールを Silk Test Workbench が認識できない場合、以下のことを行ってみてください。

- Adobe オートメーション ライブラリと Apache Flex バージョン用に適した FlexTechDomain.swc を使用して、Apache Flex アプリケーションをコンパイルします。
- 実行時ローディングを使用します。
- Apache Flex アプリケーションを空の id 属性を使用して埋め込んでいると、Apache Flex コントロールは認識されません。

Java AWT/Swing のサポート

Silk Test Workbench は、Java AWT/Swing コントロールを使用するアプリケーションまたはアプレットのテストを組み込みでサポートしています。Java AWT/Swing を使用するアプリケーションまたはアプレットを設定すると、Silk Test Workbench は標準の AWT/Swing コントロールのテストのサポートを組み込みで提供します。



注: Java AWT/Swing アプリケーションまたはアプレットに埋め込まれた Java SWT コントロールや、Java SWT アプリケーションに埋め込まれた Java AWT/Swing コントロールもテストできます。



注: イメージ クリックの記録は、Java AWT/Swing コントロールを使用するアプリケーションまたはアプレットではサポートされません。

サンプル アプリケーション

Silk Test は、サンプルの Swing テスト アプリケーションを提供しています。サンプル アプリケーションを <http://supportline.microfocus.com/websync/SilkTest.aspx> からダウンロードしてインストールします。サンプル アプリケーションをインストールしたあと、(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > Sample Applications > Java Swing > Swing Test Application**、または (Microsoft Windows 10) **スタート > Silk** をクリックします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリース ノート](#)』を参照してください。

サポートするコントロール

Java AWT/Swing のテストで使用可能なコントロールの完全な一覧については、「[Java Swing クラス リファレンス](#)」を参照してください。

Java AWT/Swing アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Java AWT/Swing でサポートされる属性には以下のものがあります。

- caption
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケータに使用されます。コントロールの **priorlabel** 値 (テキスト入力フィールドなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。
- name
- accessibleName
- *Swing* のみ : すべてのカスタム オブジェクトの定義属性は、ウィジェットに `putClientProperty("propertyName", "propertyValue")` で設定されます。



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Java メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。



注: 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの動的メソッドは `Invoke` メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、`GetDynamicMethodList` メソッドを使用します。

オブジェクトの複数の動的メソッドは `InvokeMethods` メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、`GetDynamicMethodList` メソッドを使用します。

動的プロパティの取得には `GetProperty` メソッドを、動的プロパティの設定には `SetProperty` メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、`GetPropertyList` メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを `String` 型の入力パラメータとして設定する必要がある `SetTitle` というメソッドを呼び出すには、次のように入力します :

```
control.Invoke("SetTitle", "my new title")
```



注: 通常、ほとんどのプロパティは読み取り専用で、設定できません。



注: ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、`Reflection` を使用します。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- SWT、AWT、または Swing ウィジェットのすべてのパブリック メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- プリミティブ型 (boolean、integer、long、double、string)
プリミティブ型 (int など) とオブジェクト タイプ (java.lang.Integer など) の両方がサポートされます。プリミティブ型は必要に応じて拡大変換されます。たとえば、long が必要な場所で int を渡すことができます。
- 列挙型
列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが列挙型 java.sql.ClientInfoStatus のパラメータを必要とする場合、次の文字列値を使用できます： REASON_UNKNOWN、REASON_UNKNOWN_PROPERTY、REASON_VALUE_INVALID、REASON_VALUE_TRUNCATED
- リスト
リスト、配列、または可変長引数のパラメータを持つメソッドを呼び出すことができます。リストの要素がターゲットの配列型に代入可能の場合、配列型への変換は自動的に行われます。
- その他のコントロール
コントロール パラメーターは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

Java AWT/Swing テクノロジ ドメインでの priorLabel の判別


Java AWT/Swing テクノロジ ドメインで priorLabel を判別するには、ターゲット コントロールと同じウィンドウ内のすべてのラベルおよびグループを考慮する必要があります。判別の基準は、次のとおりです。

- priorLabel の候補とみなされるのは、コントロールの上または左にあるラベル、およびコントロールが属しているグループのみです。
- コントロールの親が JViewport または ScrollPane の場合、アルゴリズムはこのコントロールを含むウィンドウが親であるかのように機能し、外側の要素はどれも関連しないとみなされます。
- 最も単純なケースでは、コントロールに最も近いラベルが priorLabel として使用されます。
- 2 つのラベルがコントロールから等距離にあり、1 つがコントロールの左、もう 1 つが上にある場合は、左側のラベルが優先します。
- 適したラベルがない場合は、最も近いグループのキャプションが使用されます。

Oracle Forms のサポート

この機能がサポートされるのは、Open Agent を使用している場合のみです。

Silk Test Workbench は、Oracle Forms ベースのアプリケーションのテストを組み込みでサポートしません。

 **注:** コントロールによっては、Silk Test Workbench は低レベルの記録のみをサポートします。

サポート対象バージョン、既知の問題、回避策の詳細については、「[リリースノート](#)」を参照してください。

Oracle Forms テストの前提条件

Oracle Forms を使ってビルドしたアプリケーションをテストするには、次の前提条件を満たす必要があります。


- 次世代 Java プラグインを有効化する必要があります。この設定は、デフォルトで有効になっています。**Java コントロール パネル** で設定を変更できます。次世代 Java プラグインの詳細については、Java のドキュメントを参照してください。
- Java セキュリティ ダイアログをテストの実行中に表示されないようにするには、アプレットにサインする必要があります。
- Micro Focus は、Names プロパティを有効にすることを推奨します。このプロパティを有効にすると、Oracle Forms ランタイムは内部の名前を公開します。この名前は、コントロールの開発者が、コントロールの Name プロパティとしてコントロールに対して指定したものです。そうでない場合、Name プロパティには、コントロールのクラス名とインデックスから構成された値が設定されます。これを使用すると、Silk Test Workbench はコントロールに対して安定したロケータを生成できます。

Oracle Forms アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Oracle Forms がサポートする属性は次のとおりです。

- **priorlabel** : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケータに使用されます。コントロールの **priorlabel** 値 (テキスト入力フィールドなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。
- **name**
- **accessibleName**

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Java SWT と Eclipse RCP のサポート

Silk Test は、SWT (Standard Widget Toolkit) コントロールのウィジェットを使用したアプリケーションのテストの組み込みサポートを提供します。Java SWT/RCP アプリケーションを設定すると、Silk Test は標準の Java SWT/RCP コントロールのテストのサポートを組み込みで提供します。

Silk Test では、以下をサポートしています。

- Java AWT/Swing アプリケーションに埋め込まれた Java SWT コントロール、および Java SWT アプリケーションに埋め込まれた Java AWT/Swing コントロールのテスト。
- Java SWT アプリケーションのテスト。
- レンダリングに SWT ウィジェットを使用した Eclipse ベースのアプリケーション。Silk Test は、Eclipse IDE ベース、および RCP ベースの両方のアプリケーションをサポートします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

SWT テストで使用できるウィジェットの完全なリストについては、「[Java SWT クラス リファレンス](#)」を参照してください。

Java SWT クラス リファレンス

Java SWT アプリケーションを設定すると、Silk Test Workbench は標準の Java SWT コントロールのテストのサポートを組み込みで提供します。

Java SWT カスタム属性

カスタム属性をテスト アプリケーションに追加して、テストをより安定させることができます。たとえば、Java SWT では、GUI を実装する開発者が属性 ('silkTestAutomationId' など) をウィジェットに対して定義することによって、アプリケーション内でそのウィジェットを一意に識別することができます。これにより、Silk Test Workbench を使用するテスト担当者は、その属性（この場合は 'silkTestAutomationId'）をカスタム属性のリストに追加すると、その一意の ID によってコントロールを識別できるようになります。カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のウィジェットが追加されると変更されるためです。

複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として 'loginName' を 2 つの異なるテキストフィールドに割り当てた場合は、'loginName' 属性を呼び出したときに、両方のフィールドが返されます。

Java SWT の例

以下のコードを使用して、テストするアプリケーションにボタンを作成する場合：

```
Button myButton = Button(parent, SWT.NONE);  
  
myButton.setData("SilkTestAutomationId", "myButtonId");
```

テストの XPath クエリ文字列に属性を追加するには、以下のクエリを使用します。

```
Dim button =  
desktop.PushButton("@SilkTestAutomationId='myButton'")
```

Java SWT アプリケーションをカスタム属性のテストに対して有効化にするには、開発者はカスタム属性をアプリケーションに含める必要があります。属性を含めるには `org.swt.widgets.Widget.setData(String key, Object value)` メソッドを使用します。

Java SWT アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Java SWT がサポートする属性は次のとおりです。

- caption
- すべてのカスタム オブジェクト定義属性




注：属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別され

ますが、他のオプションと同様にこのデフォルト設定は変更できます。ローケータ属性は、ワイルドカード? および * をサポートしています。

Java メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- SWT、AWT、または Swing ウィジェットのすべてのパブリック メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- プリミティブ型 (boolean、integer、long、double、string)

プリミティブ型 (int など) とオブジェクト タイプ (java.lang.Integer など) の両方がサポートされます。プリミティブ型は必要に応じて拡大変換されます。たとえば、long が必要な場所で int を渡すことができます。

- 列挙型

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが列挙型 java.sql.ClientInfoStatus のパラメータを必要とする場合、次の文字列値を使用できます： REASON_UNKNOWN、REASON_UNKNOWN_PROPERTY、REASON_VALUE_INVALID、REASON_VALUE_TRUNCATED

- リスト

リスト、配列、または可変長引数のパラメータを持つメソッドを呼び出すことができます。リストの要素がターゲットの配列型に代入可能な場合、配列型への変換は自動的に行われます。

- その他のコントロール

コントロールパラメーターは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

Java SWT と Eclipse アプリケーションのトラブルシューティング

一部の SWTTree メソッドが低レベル再生時に再生されない

低レベル再生を使用したとき、Expand や Collapse などの一部の SWTTree メソッドが再生されません。

この問題を解決するには、再生モードを **デフォルト** に設定します。詳細については、「全般再生オプションの設定」を参照してください。

SWTTree の非表示ノードの選択

低レベル再生を使用したとき、Silk Test Workbench は SWTTree の非表示ノードを操作できません。

この問題を解決するには、再生モードを **デフォルト** に設定します。詳細については、「全般再生オプションの設定」を参照してください。

モバイル アプリケーションのテスト

Silk Test Workbench では、ネイティブ モバイル アプリケーション (アプリ) およびモバイル Web アプリケーションを自動的にテストすることができます。Silk Test Workbench を使用してモバイル アプリケーションを自動的にテストすることには、次のメリットがあります。

- モバイル アプリケーションのテスト時間を大幅に減少させることができます。
- テストを一旦作成すれば、数多くの異なるデバイスやプラットフォーム上でモバイル アプリケーションをテストできます。
- エンタープライズ モバイル アプリケーションに要求される信頼性とパフォーマンスを確保できます。
- QA チームのメンバーおよびモバイル アプリケーションの開発者の効率を向上できます。
- モバイル アプリケーションは、多くのモバイル デバイスとプラットフォームで動作することを要求されるため、アジャイルにフォーカスした開発環境にとって手動テストは十分効率的とは言えない場合があります。



注: Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。



注: Silk Test Workbench は、Android および iOS デバイスの両方でのモバイル アプリのテストをサポートします。

モバイル アプリケーションのテストをサポートするオペレーティング システムとサポートするブラウザについての情報は、『[リリース ノート](#)』を参照してください。

Android

Silk Test Workbench では、Android デバイスまたは Android エミュレータ上のモバイル アプリケーションをテストすることができます。

Android 上のモバイル アプリケーションのテストにおける前提条件

Android デバイスや Android エミュレータ上のモバイル アプリケーションをテストする前に、次の前提条件を満たしていることを確認してください。

- ネイティブ モバイル アプリに Web ビューを追加したハイブリッド アプリを作成した場合は、アプリを Silk Test Workbench でテスト可能にするために、次のコードをアプリに追加してください。

```
WebView.setWebContentsDebuggingEnabled(true);  
webView.getSettings().setJavaScriptEnabled(true);
```

- Silk Test Workbench では、Android 4.4 のデバイス画面のライブ ビュー表示をサポートしていません。このため、Android 5 以降を使用することを Micro Focus は推奨しています。

Android 上のモバイル アプリケーションのテスト

物理 Android デバイスや Android エミュレータ上のモバイル アプリケーションをテストするには、次のタスクを実行します。

1. Android 上のモバイル アプリケーションのテストにおける前提条件を満たすことを確認します。
詳細については、「[Android 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。
2. Android エミュレータ上のモバイル アプリケーションをテストするには、Silk Test Workbench 用にエミュレータを設定します。
詳細については、「[Silk Test 用に Android エミュレータを設定する](#)」を参照してください。
3. Silk Test Workbench をインストールしたマシンで Android エミュレータを開始するか、デバイスを接続します。
4. 物理 Android デバイスを初めて使用するマシンでテストする場合、適切な Android USB ドライバをマシンにインストールします。
詳細については、「[USB ドライバのインストール](#)」を参照してください。
5. 物理 Android デバイス上でモバイル アプリケーションをテストする場合は、Android デバイスの USB デバッグを有効化します。
詳細については、「[USB デバッグの有効化](#)」を参照してください。
6. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
7. モバイル アプリケーション用のテストを作成します。
8. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
9. モバイル Web アプリケーションをテストするには：
 - a) **Web** タブを選択します。
 - b) 使用するモバイル ブラウザーを選択します。
 - c) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。
10. ネイティブ モバイル アプリケーションまたはハイブリッド アプリケーションをテストするには：



注: Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「[ライセンス情報](#)」を参照してください。

- a) **モバイル** タブを選択します。
- b) アプリをテストするモバイル デバイスをリストから選択します。

- c) **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。

このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。

Silk Test Workbench は、モバイル デバイスまたはエミュレータ上に指定したアプリをインストールします。

11OK をクリックします。

Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。

12記録 ウィンドウを使用して、モバイル アプリケーションに対するテストを記録します。

詳細については、「[モバイル アプリケーションの記録](#)」を参照してください。

13すべての操作の記録を終えたら、記録を停止します。

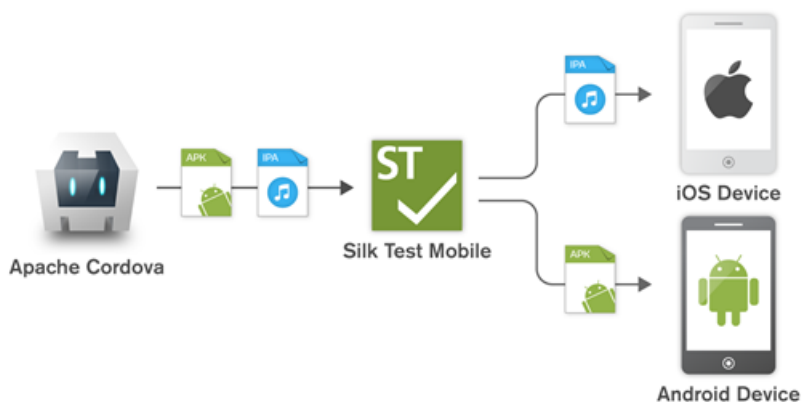
14テストを再生します。

15テスト結果を分析します。

Android 上のハイブリッド アプリケーションのテスト

ハイブリッド アプリケーション (アプリ) は、デバイス上で実行されるネイティブ アプリケーションのようなアプリですが、HTML5、CSS、JavaScript などの Web テクノロジーを使用して記述されたアプリです。

Silk Test Workbench は、ネイティブ コンテナに埋め込まれた単一の Web ビューで構成されたデバッグ ハイブリッド アプリのテストにする完全なブラウザー サポートを提供します。このようなハイブリッド アプリの一般的な例は、Apache Cordova アプリケーションです。



非デバッグ ハイブリッド アプリをテスト可能にするには、次のコードをアプリに追加してアプリをリモート デバッグできるようにします。

```
WebView.setWebContentsDebuggingEnabled(true);  
webView.getSettings().setJavaScriptEnabled(true);
```


リモート デバッグを有効化しない非デバッグ ハイブリッド アプリや、複数の Web ビューを含んだハイブリッド アプリをテストするには、Silk Test Workbench フォールバック サポートを有効化するために、オプション `OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT` を `TRUE` に設定します。詳細については、「[詳細オプションの設定](#)」を参照してください。フォールバック サポートを有効化すると、Silk Test Workbench は Web ビューのコントロールをブラウザー コントロールではなく、ネイティブ モバイル コントロールとして解決して処理します。たとえば、以下のコードは、ブラウザー サポートを使用したときのリンクのクリックです。

```
' VB .NET code  
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, False)  
_desktop.DomLink("//BrowserApplication//BrowserWindow//INPUT[@id='email']").Click()
```

フォールバック サポートを有効化すると、同じリンクをクリックするコードは次のようになります。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, True)
_desktop.Find("//BrowserApplication//BrowserWindow//MobileTextField[@resource-
id='email']").Click()
```

Silk Test Workbench は、Chrome リモート デバッグをサポートする Web ビューを検出できます。Silk Test Workbench は、*com.android.webview* パッケージまたは *com.google.android.webview* パッケージ（多くの Android デバイス上のデフォルト パッケージ）のいずれかを使用した Web ビューを検出できます。

 **注:** Silk Test Workbench は、Android 4.4 以降でのハイブリッド アプリのテストをサポートします。Android でハイブリッド アプリをテストするには、Android システムの WebView バージョン 51 以降が必要です。

Android 上のハイブリッド アプリをテストする手順は、モバイル ネイティブ アプリケーションをテストする手順と同じです。詳細については、「[Android 上のモバイル アプリケーションのテスト](#)」を参照してください。

USB ドライバのインストール

モバイル アプリケーションをテストするために、ローカル マシンに最初に Android デバイスに接続するには、適切な USB ドライバをインストールする必要があります。

デバイスの製造元は、そのデバイスに必要なすべてのドライバをもった実行可能ファイルを提供している可能性があります。この場合、ローカル マシンにその実行可能ファイルをインストールするだけです。製造元がこのような実行可能ファイルを提供していない場合、マシン上にデバイスに対する単一の USB ドライバをインストールできます。

Android USB ドライバをインストールするには：

1. デバイス用の適切なドライバをダウンロードします。
たとえば、Google Nexus デバイス用の USB ドライバを検索し、インストールする場合は、<http://developer.android.com/tools/extras/oem-usb.html> を参照します。
2. Android デバイスをローカル マシンの USB ポートに接続します。
3. デスクトップ、または **Windows Explorer** から、**コンピュータ** を右クリックし、**管理** を選択します。
4. 左側のペインで、**デバイス マネージャ** を選択します。
5. 右側のペインで、**その他のデバイス** を探して展開します。
6. デバイス名 (*Nexus 5x* など) を右クリックして、**ドライバ ソフトウェアの更新** を選択します。**ハードウェアの更新ウィザード** が開きます。
7. **コンピュータを参照してドライバ ソフトウェアを検索します** を選択して、**次へ** をクリックします。
8. **参照** をクリックし、USB ドライバをダウンロードしたフォルダに移動します。
9. USB ドライバを選択します。
- 10 **次へ** をクリックしてドライバをインストールします。

USB デバッグの有効化

Android Debug Bridge (adb) 上で Android デバイスと通信するために USB デバッグを有効化します。

1. Android デバイスで設定を開きます。
2. **開発者向けオプション** (Dev Settings) をタップします。
開発者向けオプションは、デフォルトでは表示されません。開発者向けオプションがデバイスの設定メニューに含まれていない場合：
 - a) 画面を下にスクロールさせて、デバイスが携帯電話の場合は **端末情報** を、タブレットの場合は **タブレット情報** をタップします。

b) 再度画面を下にスクロールさせて **ビルド番号** を 7 回タップします。

3. **開発者向けオプション** ウィンドウで、**USB デバッグ** をオンにします。


4. デバイスの USB モードをデフォルトの設定である **メディア デバイス (MTP)** に設定します。
詳細については、デバイスのドキュメントを参照してください。

Android デバイスの推奨設定

Silk Test Workbench を使用したテストを最適化するために、テストしたい Android デバイスで次の設定を行ってください。


- USB デバッグを Android デバイスで有効化します。詳細については、「[USB デバッグの有効化](#)」を参照してください。
- Android デバイスは、Open Agent を実行しているマシンに、メディア デバイスとして接続されている必要があります。Android デバイスの USB モードは、**メディア デバイス (MTP)** を設定します。
- Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。

Silk Test Workbench 用に Android エミュレータを設定する


 **注:** Android エミュレータを使用する場合、Silk Test Workbench が使用している adb サーバー以外の adb サーバーが実行されてます。実行中の adb サーバーのバージョンが異なる場合、Open Agent とデバイスとの接続が不安定になったり、接続できない場合があります。このようなバージョンの不一致によるエラーを避けるには、環境変数 `SILK_ANDROID_HOME` に Android SDK ディレクトリへのパス (C:\Users\%<ユーザー>\AppData\Local\Android\android-sdk など) を指定してください。Information Service が実行中の場合、Windows のサービス マネージャーを使用して、Silk Test Information Service を再起動して更新した環境変数を適用する必要があります。環境変数が設定されていない場合は、Silk Test Workbench は Silk Test Workbench に同梱されたバージョンの adb を使用します。

Silk Test Workbench を使用して Android エミュレータ上でモバイル アプリケーションをテストする場合、テスト用にエミュレータを設定する必要があります。

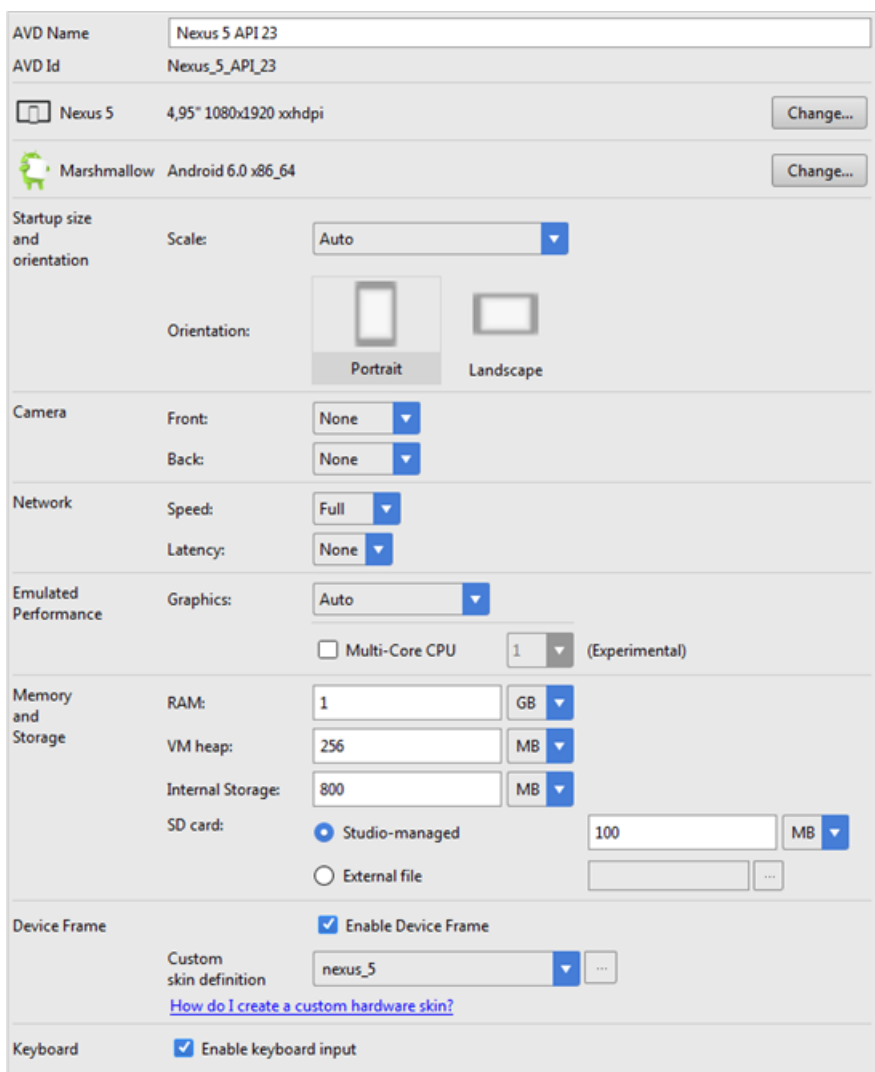
1. Android SDK の最新版をインストールします。
Android SDK のインストールと設定についての詳細は、「[Get the Android SDK](#)」を参照してください。
2. Android Studio 2 をインストールします。

 **ヒント:** Android Studio 2 のインストールをスキップして、Android SDK で提供されるエミュレータを使用することもできます。ただし、エミュレータのパフォーマンスを向上させるため、Android Studio 2 をインストールすることを、Micro Focus では推奨しています。このトピックの以下の手順では、Android Studio 2 がインストールされていることを前提としています。

3. Android Studio 2 から **AVD Manager** を開きます。
4. **Create Virtual Device** をクリックします。
5. 仮想デバイスを選択します。
6. **Next** をクリックします。
7. Google API を含む Android のシステム イメージをダウンロードして選択します。
8. **Next** をクリックします。
9. 要件に従って仮想デバイスを設定します。
10. **Show Advanced Settings** をクリックします。
11. エミュレータが使用する RAM サイズとヒープ領域を対象のマシンで管理可能な量に調整します。

 **ヒント:** 最低 1 GB の RAM と 256 MB のヒープ領域を使用することを、Micro Focus では推奨しています。

12. **Emulated Performance** 領域のリストで **Auto** を選択します。

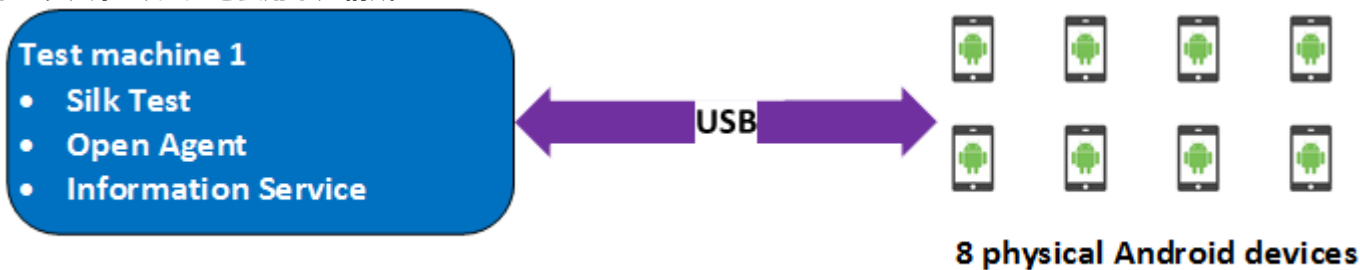


13 **Finish** をクリックします。

並列テスト実行のテスト済みの構成

Silk Test Workbench を使用して、複数の Android デバイス上で並列に自動テストを実行することができます。並列に実行できる Android デバイスの数は利用可能なハードウェアに依存します。Micro Focus では、次のハードウェア構成でテストの実行を確認しています。

単一テスト マシンを使用した構成

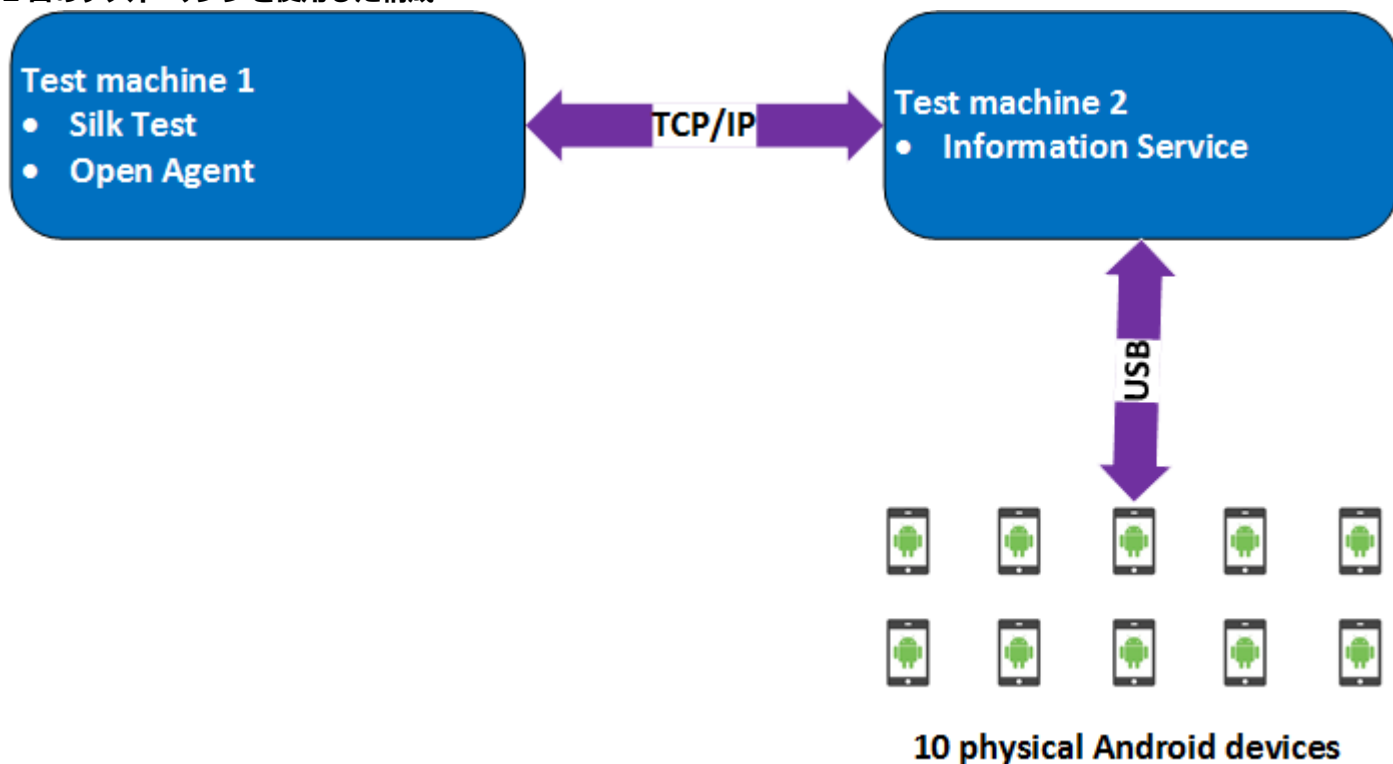


USB により Android デバイスを直接接続した単一のテスト マシンを使用して、最大 8 台の物理 Android デバイスまでの並列テストを行いました。

テスト マシンは、次のハードウェア仕様の Lenovo ThinkPad T450 です。

- Intel® Core™ i7 - 5600U CPU @ 2.60 GHz
- 2 コア (4 スレッド)
- 8 GB RAM

2 台のテスト マシンを使用した構成



また、2 台のテスト マシンを使用してテストを行いました。1 台に Silk Test Workbench をインストールし、もう 1 台には Silk Test Information Service をインストールして最初のマシンのリモート ロケーションとして設定しました。このような構成を使用して、最大 10 台の物理 Android デバイスまでの並列テストを行いました。

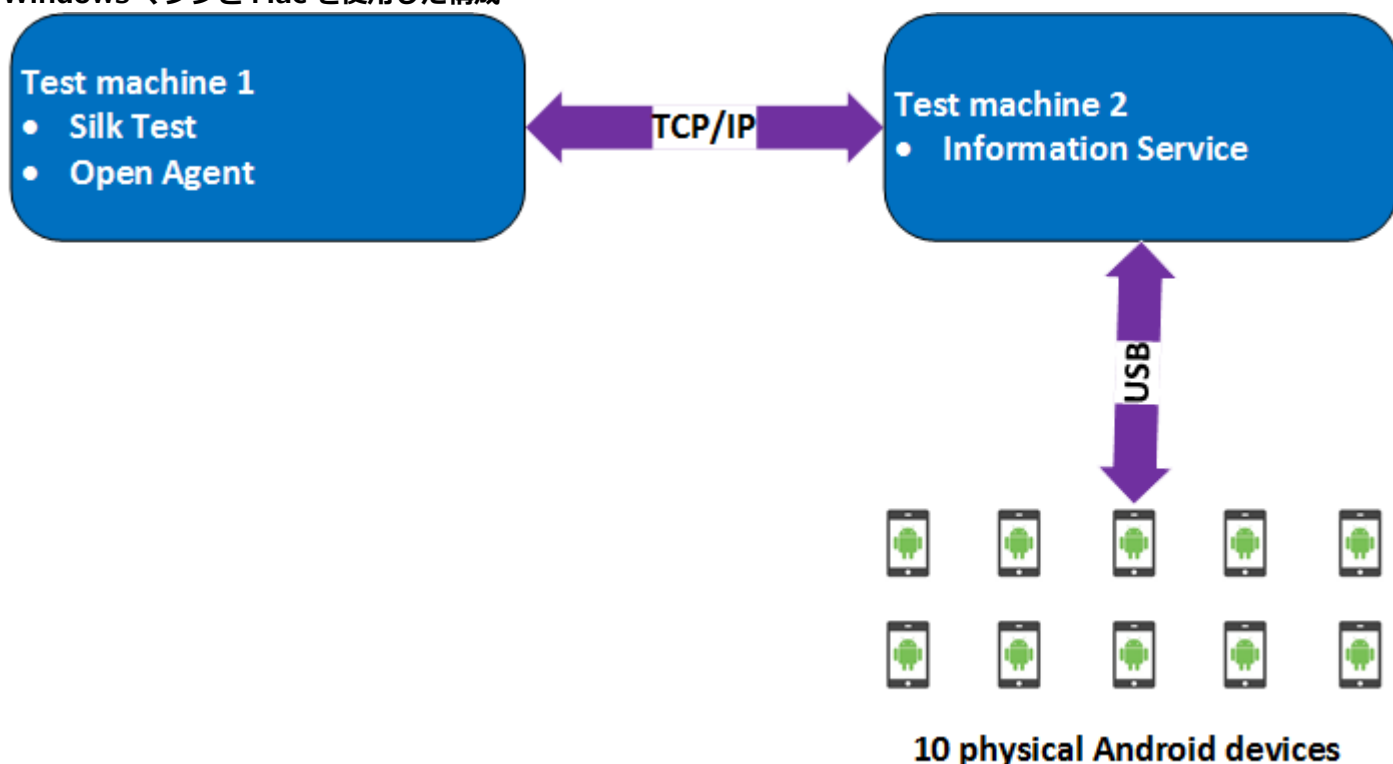
テスト マシン 1 は、次のハードウェア仕様の Lenovo ThinkPad T450 です。

- Intel® Core™ i7 - 5600U CPU @ 2.60 GHz
- 2 コア (4 スレッド)
- 8 GB RAM

テスト マシン 2 は、次のハードウェア仕様の Dell Precision T1700 です。

- Intel® Core™ i7 - 4770 CPU @ 3.40 GHz
- 4 コア (8 スレッド)
- 16 GB RAM

Windows マシンと Mac を使用した構成



さらに、2 台のテスト マシンを使用したテストで、Windows マシンに Silk Test Workbench をインストールし、Mac には Silk Test Information Service をインストールして最初のマシンのリモート ロケーションとして設定しました。このような構成を使用して、最大 10 台の物理 Android デバイスまでの並列テストを行いました。

テスト マシン 1 は、次のハードウェア仕様の Lenovo ThinkPad T450 です。

- Intel® Core™ i7 - 5600U CPU @ 2.60 GHz
- 2 コア (4 スレッド)
- 8 GB RAM


テスト マシン 2 は、次のハードウェア仕様の Apple Mac Mini です。

- Intel® Core™ i5 - 4782U CPU @ 2.60 GHz
- 2 コア (4 スレッド)
- 16 GB RAM

iOS


Silk Test Workbench では、iOS デバイスまたは iOS シミュレータ上のモバイル アプリケーションをテストすることができます。

iOS の以前のバージョンと比べて、iOS 9.3 に対して大幅な変更が Apple によってなされたため、Silk Test は iOS 9.3 以降のモバイル アプリケーションのテストをサポートします。サポートされている iOS のバージョンのリストについては、『[リリースノート](#)』を参照してください。

 **注:** iOS 11 上のモバイル アプリケーションのテストには Xcode 9 が必要です。Mac 上で Xcode 9 を使用している場合は、この Mac に接続 (または実行) されている iOS 11 より前のバージョンの iOS をインストールした物理デバイスおよびシミュレータ上でのテストはサポートされません。iOS 9.3 または iOS 10 の物理デバイスまたはシミュレータでテストする場合は、Xcode 8.3 を使用してください。



ヒント: iOS 9.3 より前のバージョンの iOS をテストする場合は、Silk Test 17.5 を使用してください。Silk Test 19.5 を使用した iOS 上でのテストについて、Silk Test 17.5 でのテストとの主な違いを以下の表に示します。

Silk Test19.5	Silk Test 17.5
iOS 9.3、iOS 10、iOS 11 をサポートします。	iOS 8.1、8.2、8.3、8.4、9.0、9.1、9.2、9.3 をサポートします。
Xcode 8.3、Xcode 9 をサポートします。	Xcode 6、Xcode 7 をサポートします。
同じユーザー セッションでの複数の物理 iOS デバイスのテストをサポートします。	複数の物理 iOS デバイスをテストする場合は、Mac 上に複数のユーザー セッションが必要です。
 ヒント: Silk Test 17.5 を使用してテストするために、複数のユーザー セッションを作成している場合、1 つ以外のすべてのユーザー セッションを削除することを Micro Focus は推奨します。	

iOS 上のモバイル アプリケーションのテストにおける前提条件

iOS デバイスや iOS シミュレータ上のモバイル アプリケーションをテストする前に、次の前提条件を満たしていることを確認してください。

- 現在のバージョンの Silk Test Information Service が Mac 上にインストールされている。詳細については、「[Silk Test Information Service を Mac にインストールする](#)」を参照してください。
- 物理 iOS デバイス上でアプリケーションをテストする場合は、以下の項目を確認してください。
 - デバイスが Mac に接続されている。
 - デバイスが iOS のサポートされているバージョンである。サポートされている iOS のバージョンのリストについては、『[リリース ノート](#)』を参照してください。
- iOS シミュレータ上でアプリケーションをテストする場合は、以下の項目を確認してください。
 - iOS シミュレータのイメージが Mac 上にインストールされている。
 - iOS シミュレータのイメージが iOS のサポートされているバージョンである。サポートされている iOS のバージョンのリストについては、『[リリース ノート](#)』を参照してください。
- 物理 iOS デバイス上でアプリケーションをテストする場合は、デバイスと Mac の両方ともタイムゾーンが同じである。
- サポートするバージョンの Xcode が Mac 上にインストールされている。
- Windows マシン上に Silk Test Workbench がインストールされている。
- Mac が Windows マシンと同じネットワークにあり、Windows マシンにリモート ロケーションとして追加されている。
- iOS デバイス上でネイティブ モバイル アプリをテストする場合は、Developer Account でサインされたアプリの .ipa ファイルが作成されている。詳細については、「[iOS アプリのテストの準備](#)」を参照してください。
- iOS シミュレータ上でネイティブ モバイル アプリをテストする場合は、ZIP 形式に圧縮したアプリが作成されている。詳細については、「[iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト](#)」を参照してください。
- iOS デバイスと iOS シミュレータの両方でネイティブ モバイル アプリをテストする場合は、サインした .ipa ファイルと ZIP した .app ディレクトリ（ファイル拡張子以外同じ名前）の両方が同じフォルダに作成されている。
- ネイティブ モバイル アプリをテストする場合は、iOS デバイスの ID がアプリをサインするのに使用した Developer Profile に関連付けられている。
- iOS デバイスのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > 一般 > パスコードロック** を選択します。
- テスト中に Mac の画面がオフにならないようにしてください。オフになると、**再生ステータス** ダイアログ ボックスに何も表示されなくなります。

- iOS シミュレータ上でモバイル アプリケーションをテストする場合は、テスト中に Mac のディスプレイ スリープが無効化されている。
- 物理 iOS デバイス上のネイティブ モバイル アプリをテストする場合は、デバイスで UI オートメーションが有効化されている。詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
- 物理 iOS デバイス上で Apple Safari を使用してのモバイル Web アプリケーションをテストする場合は、**Web インスペクタ** が有効化されている。詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
- Lightning コネクタを持つ iOS デバイスを使用することを Micro Focus では推奨しています。これは、Silk Test Workbench では、Lightning ケーブルで Mac に接続していない iOS デバイスのデバイス画面のライブ ビュー表示をサポートしていないためです。


物理 iOS デバイス上のネイティブ モバイル アプリケーションのテスト



注: Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「[ライセンス情報](#)」を参照してください。


iOS 上のモバイル アプリケーションのテストにおける前提条件については、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。ネイティブ モバイル アプリケーションをテストする際の既知の制限については、「[モバイル ネイティブ アプリケーションのテストにおける制限事項](#)」を参照してください。

物理 iOS デバイス上のネイティブ モバイル アプリケーション (アプリ) やハイブリッド アプリケーションをテストするには、次のタスクを実行します。


1. テストする iOS デバイスを準備します。
詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
 2. テストするアプリを準備します。
詳細については、「[iOS アプリのテストの準備](#)」を参照してください。
 3. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
 4. iOS デバイスが接続されている Mac を、Silk Test がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
-  **注:** 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。
5. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
 6. モバイル アプリケーション用のテストを作成します。
 7. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
 8. **モバイル** タブを選択します。
 9. アプリをテストするモバイル デバイスをリストから選択します。
 - 10 **参照** をクリックしてアプリ ファイルを選択するか、アプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。
このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
Silk Test Workbench は、モバイル デバイス上に指定したアプリをインストールします。
 - 11 **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定** > **Touch ID とパスコード** を選択します。
 - 12 すべての操作の記録を終えたら、記録を停止します。

13テストを再生します。

14テスト結果を分析します。

 **注:** iOS デバイスと iOS シミュレータの両方でネイティブ モバイル アプリをテストする場合は、サインした .ipa ファイルと ZIP した .app ディレクトリ（ファイル拡張子以外同じ名前）の両方が同じフォルダに作成されている。

iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト


 **注:** Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。

iOS 上のモバイル アプリケーションのテストにおける前提条件については、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。ネイティブ モバイル アプリケーションをテストする際の既知の制限については、「[モバイル ネイティブ アプリケーションのテストにおける制限事項](#)」を参照してください。

iOS シミュレータ上のネイティブ モバイル アプリケーション (アプリ) やハイブリッド アプリケーションをテストするには、次のタスクを実行します。

1. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
2. アプリの Xcode プロジェクトで、iOS シミュレータ用にアプリをコンパイルします。
Xcode UI からでも、コマンドラインからでもアプリをコンパイルできます。たとえば、iOS 10.0 で iOS シミュレータ用のアプリをコマンドラインでコンパイルするには、次のコマンドを実行します。

```
xcodesbuild -sdk iphonesimulator10.0
```
3. アプリの .app ディレクトリを .zip ファイルに Zip します。
デフォルトでは、.app ディレクトリは、~/Library/Developer/Xcode/DerivedData ディレクトリにあります。Xcode で **File > Project Settings** をクリックすれば、ディレクトリがある場所を確認できます。
4. iOS シミュレータがインストールされている Mac を、Silk Test Workbench がインストールされている Windows マシンに、リモートロケーションとして追加します。
詳細については、「[リモートロケーションの編集](#)」を参照してください。

 **注:** Mac にインストールされている 1 つの iOS シミュレータでのみテストを実行できます。複数の Silk Test Workbench ユーザーが、同じ Mac にインストールされている複数の iOS シミュレータ上で同時にテストを実行することはできません。


5. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
6. モバイル アプリケーション用のテストを作成します。
7. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
8. **モバイル** タブを選択します。
9. リストから iOS シミュレータを選択します。
- 10 **参照** をクリックして Zip したアプリ ファイルを選択するか、Zip したアプリ ファイルへの完全パスを **モバイル アプリ ファイル** テキスト フィールドに入力します。
このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
Silk Test Workbench は、iOS シミュレータ上に指定したアプリをインストールします。
- 11 **OK** をクリックします。

iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。

12すべての操作の記録を終えたら、記録を停止します。

13テストを再生します。


14テスト結果を分析します。

 **注:** iOS デバイスと iOS シミュレータの両方でネイティブ モバイル アプリをテストする場合は、サインした .ipa ファイルと ZIP した .app ディレクトリ (ファイル拡張子以外同じ名前) の両方が同じフォルダに作成されている。

物理 iOS デバイス上のモバイル Web アプリケーションのテスト

iOS 上のモバイル アプリケーションのテストにおける前提条件については、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。モバイル Web アプリケーションをテストする際の既知の制限については、「[モバイル Web アプリケーションのテストにおける制限事項](#)」を参照してください。

物理 iOS デバイス上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。


1. テストする iOS デバイスを準備します。
詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
2. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
3. iOS デバイスが接続されている Mac を、Silk Test がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
 **注:** 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。
4. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
5. モバイル アプリケーション用のテストを作成します。
6. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
7. モバイル Web アプリケーションをテストするには：
 - a) **Web** タブを選択します。
 - b) 使用するモバイル ブラウザーを選択します。
 - c) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。
8. **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。
9. すべての操作の記録を終えたら、記録を停止します。
10. テストを再生します。
11. テスト結果を分析します。

iOS シミュレータ上のモバイル Web アプリケーションのテスト

モバイル Web アプリケーションをテストする際の既知の制限については、「[モバイル Web アプリケーションのテストにおける制限事項](#)」を参照してください。

iOS シミュレータ上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。

1. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
2. iOS シミュレータがインストールされている Mac を、Silk Test がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。

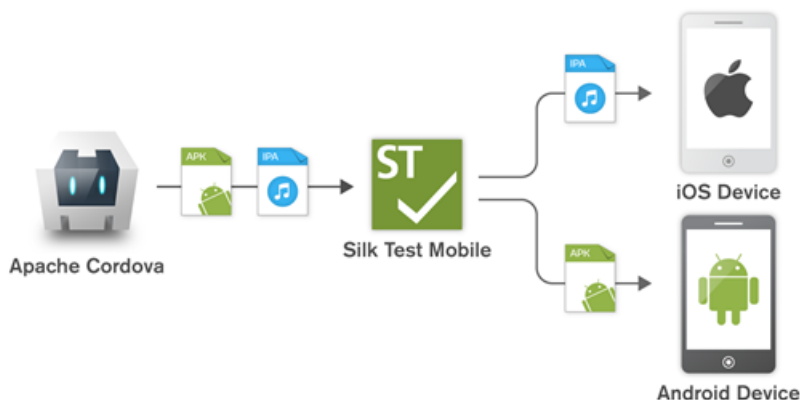
 **注:** 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。

3. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
4. モバイル アプリケーション用のテストを作成します。
5. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
6. モバイル Web アプリケーションをテストするには：
 - a) **Web** タブを選択します。
 - b) 使用するモバイル ブラウザーを選択します。
 - c) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。
7. **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。
8. すべての操作の記録を終えたら、記録を停止します。
9. テストを再生します。
10. テスト結果を分析します。

iOS 上のハイブリッド アプリケーションのテスト

ハイブリッド アプリケーション (アプリ) は、デバイス上で実行されるネイティブ アプリケーションのようなアプリですが、HTML5、CSS、JavaScript などの Web テクノロジーを使用して記述されたアプリです。

Silk Test Workbench は、ネイティブ コンテナに埋め込まれた単一の Web ビューで構成されたデバッグ ハイブリッド アプリのテストにする完全なブラウザー サポートを提供します。このようなハイブリッド アプリの一般的な例は、Apache Cordova アプリケーションです。



リモート デバッグを有効化しない非デバッグ ハイブリッド アプリや、複数の Web ビューを含んだハイブリッド アプリをテストするには、Silk Test Workbench フォールバック サポートを有効化するために、オプション `OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT` を `TRUE` に設定します。詳細については、「詳細オプションの設定」を参照してください。フォールバック サポートを有効化すると、Silk Test Workbench は Web ビューのコントロールをブラウザー コントロールではなく、ネイティブ モバイル コントロールとして解決して処理します。たとえば、以下のコードは、ブラウザー サポートを使用したときのリンクのクリックです。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, False)
_desktop.DomLink("//BrowserApplication//BrowserWindow//INPUT[@id='email']").Click()
```

フォールバック サポートを有効化すると、同じリンクをクリックするコードは次のようになります。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, True)
```

```
desktop.Find("//BrowserApplication//BrowserWindow//MobileTextField[@resource-id='email']").Click()
```

iOS 上のハイブリッド アプリをテストする手順は、モバイル ネイティブ アプリケーションをテストする手順と同じです。詳細については、「[物理 iOS デバイス上のネイティブ モバイル アプリケーションのテスト](#)」または「[iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト](#)」を参照してください。

iOS デバイス上のハイブリッド アプリをテストする前に、デバイスで **Web インспекタ** が有効化されていることを確認してください。詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。

iOS デバイスのテストの準備



注: Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「[ライセンス情報](#)」を参照してください。

モバイル アプリケーションをテストするために iOS デバイスを準備するには：

1. Mac 上で Xcode を起動します。
2. iOS デバイスを Mac に接続します。
3. iOS デバイスで、**設定 > デベロッパ** をクリックします。



ヒント: デベロッパ メニューが iOS デバイスに表示されていない場合は、デバイスと Mac を再起動します。

4. **Enable UI Automation** をオンにします。
5. Apple Safari 上でモバイル Web アプリケーションをテストするには、**設定 > Safari > 詳細** をクリックします。
6. **Web インспекタ** をオンにします。
7. iOS シミュレータ上でテストを行う場合は、**Rotate Device Automatically** をオンにします。
この設定は、**Silk Test Configuration Assistant** を使うか、手動でオンにできます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。手動でオンにするには、以下の操作を実行します。
 - a) Mac 上で iOS シミュレータを開始します。
 - b) Xcode 9 以降を使用している場合は、**Hardware** メニューを開きます。
以前のバージョンの Xcode を使用している場合は、**Debug** メニューを開きます。
 - c) **Rotate Device Automatically** をオンにします。

iOS アプリのテストの準備

Silk Test Workbench を使用して特定の デバイス上で特定の iOS アプリをテストできるようにするには、次の項目を考慮する必要があります。

- 特定の デバイスに手動でインストールできる iOS アプリに対してのみテストを自動化できます。iOS アプリにサインできるようにするために、*Apple Developer Program* のメンバーシップに登録する必要があります。詳細については、「[メンバーシップの選択](#)」を参照してください。*Apple Developer Program* のメンバーシップに登録せずにテストを行う場合は、「[パーソナル チーム プロファイルを使用した物理 iOS デバイス上でのテスト](#)」を参照してください。



注: App Store で配布するように作成された アプリや、任意の iOS デバイスに手動でインストールできるアプリを自動的にテストできません。


- 特定の デバイスで アプリをインストールして実行する前に、iOS デバイスを Apple Developer アカウントを使用して登録する必要があります。
- デバイスに iOS アプリをインストールするには、Xcode を使用してアプリの IPA ファイルを作成する必要があります。特定の iOS デバイスでのテスト用に IPA ファイルを作成するには、Apple Developer Program のメンバーとして Xcode の *Archive* 機能を使用します。この機能には 2 つのオプションがあります。

- *Apple Developer Enterprise Program* のメンバーの場合、**Save for Ad Hoc Deployment** オプションを使用できます。
- *Apple Developer Enterprise Program* のメンバー以外の *Apple Developer Program* のメンバーの場合、**Save for Development Deployment** オプションを使用できます。

詳細については、「[Exporting Your App for Testing \(iOS, tvOS, watchOS\)](#)」を参照してください。

シミュレータ上の特定のアプリを Silk Test Workbench を使用してテストできるようにするには、Xcode を使用してアプリの Zip ファイルを作成してから iOS シミュレータ上にインストールします。詳細については、Xcode のドキュメントを参照してください。

Silk Test Information Service を Mac にインストールする

 **注:** Information Service を Mac にインストールするには、Mac の管理者権限が必要です。


Mac 上の Apple Safari や、Mac に接続されている iOS や Android デバイス上のモバイルアプリケーションに対するテストを作成して実行するには、Mac に Silk Test Information Service (Information Service) をインストールしてから、**リモートロケーション** ダイアログ ボックスを使用して、Silk Test Workbench をインストールした Windows マシンと Mac を接続する必要があります。

Information Service を Mac にインストールするには：

1. Java JDK が Mac 上にインストールされていることを確認します。
2. iOS デバイス上でモバイルアプリケーションをテストする場合は、Xcode が Mac 上にインストールされていることを確認します。
3. Information Service セットアップ ファイル (SilkTestInformationService<バージョン>-<ビルド番号>.pkg) にアクセスします。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードした場合は、Silk Test インストール ディレクトリ (C:\Program Files (x86)\Silk\SilkTest など) の macOS フォルダを開きます。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードしなかった場合は、[Micro Focus SupportLine](#) からセットアップ ファイルをダウンロードできます。
4. SilkTestInformationService<バージョン>-<ビルド番号>.pkg ファイルを Mac にコピーします。
5. SilkTestInformationService<バージョン>-<ビルド番号>.pkg を実行して、Information Service をインストールします。
6. インストール ウィザードの指示に従います。
7. パスワードを尋ねられた場合、現在サインインしている Mac ユーザーのパスワードを入力します。
8. Apple Safari が開き、SafariDriver を信頼するかどうかを尋ねるメッセージ ボックスが表示されたら、**信頼** をクリックします。

 **注:** リモート接続ではなく、直接 Mac にログインしている場合には、SafariDriver だけをインストールできます。

9. インストールを完了するために、現在の Mac ユーザーをログアウトします。Information Service が正しくインストールされていることを確認するには、Mac にログインします。
- 10 macOS Mojave (10.14) 以降の Mac に Information Service をインストールした場合、Mac にログインした後で Silk Test に対するオートメーションの承認を許可する必要があります。許可を与える必要がある場合、Silk Test は要求の承認ダイアログを自動的に表示します。
 - a) **OK** をクリックして、許可して情報ダイアログを閉じます。
 - b) その後に表示されるすべての要求の承認ダイアログに対して、**OK** をクリックします。


 **重要:** これらの Silk Test に対する承認を許可しないと、Google Chrome 上の Web アプリケーションや iOS デバイスやシミュレーター上のモバイルアプリケーションを、この Mac でテストでき

なくなります。いずれかの承認ダイアログで、誤って **許可しない** をクリックしてしまった場合は、Mac 上でターミナルを開き、次のコマンドを実行してください。


```
sudo tccutil reset AppleEvents
```

その後、Mac を再起動して、再度承認ダイアログが表示されたら **OK** をクリックして許可します。

11画面の右上にある Silk Test アイコンをクリックすると、利用可能なデバイスをブラウザーを確認できます。

 **ヒント:** Silk Test アイコンが表示されない場合は、Mac を再起動してください。

iOS 上でのモバイル アプリケーションのテストのための Mac の準備

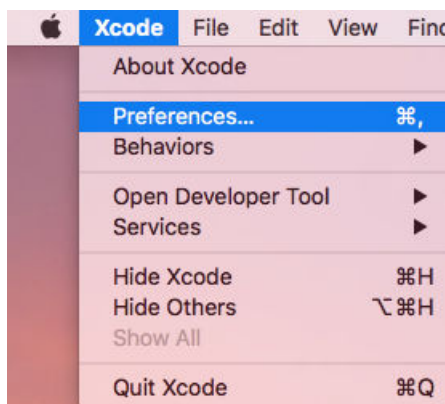
 **注:** Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。

iOS 上でモバイル アプリケーションをテストするためには、iOS デバイスを接続した、または iOS シミュレータを実行している Mac が必要です。この Mac には Xcode がインストールされている必要があります。iOS 上のモバイル アプリケーションのテストにおける前提条件についての詳細は、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。

iOS テストを物理 iOS デバイス上で実行するには、**Silk Test Configuration Assistant** の指示に従って WebDriverAgentRunner Xcode プロジェクトを設定します。**Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。

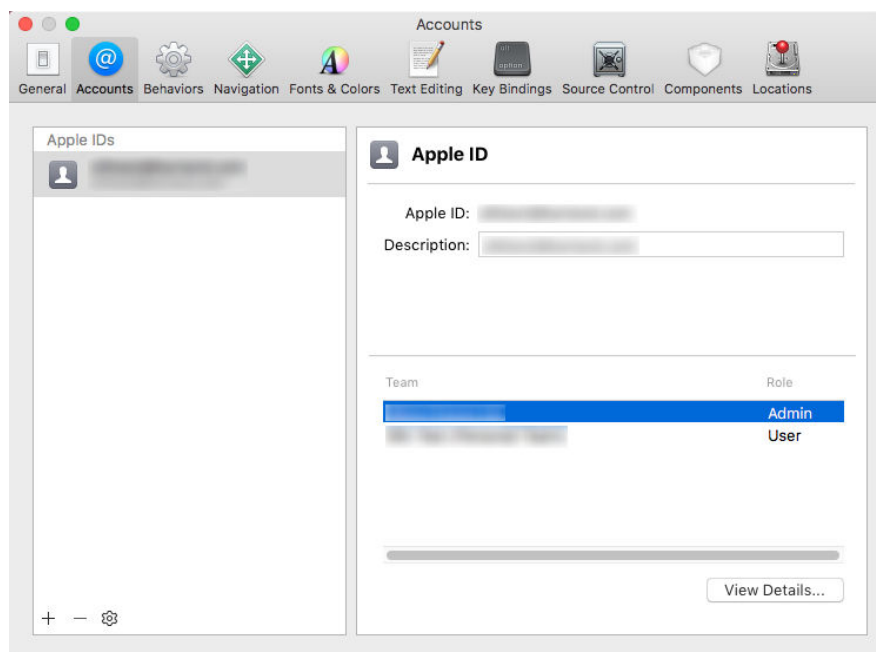
WebDriverAgentRunner Xcode プロジェクトを手動でビルドしたい場合には、次の手順を実行します。

1. Mac 上で Xcode を起動します。
2. **Xcode > Preferences** を選択します。



3. **Preferences** ウィンドウで、アカウントを選択します。

- a) **Accounts** タブを選択します。
- b) **Apple ID** を選択します。
- c) **Team** を選択します。
- d) **View Details** をクリックします。



4. **Apple Member Center** にアクセスし、開発チーム ID を取得します。
5. ターミナルで、`~/silk/silktest/conf/` に移動します。
6. `xcconfig` ファイル テンプレート `silktest.xcconfig.sample` を `silktest.xcconfig` という名前に変更します。
7. 開発チーム ID を `silktest.xcconfig` ファイルに追加します。
`DEVELOPMENT_TEAM = <your development team>`
8. Mac 上のターミナルで次のコマンドを実行すると、WebDriverAgentRunner プロジェクトが正しく準備できたかどうかを確認できます。
 - a) 物理 iOS デバイスの UDID (Unique Device ID) を見つけます。
`idevice_id -l`
 - b) WebDriverAgentRunner プロジェクトに移動します。
`cd /Application/Silk/Mobile/common/Appium/node_modules/appium-xcuitest-driver/WebDriverAgent`
 - c) WebDriverAgent がビルドできるかどうかをテストします。
`xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -xcconfig ~/.silk/silktest/conf/silktest.xcconfig -destination 'id=<udid>' test`
`<udid>` を先ほど見つけた UDID で置き換えます。



ヒント: `xcodebuild` コマンドが失敗する場合には、エラーメッセージの手順に従ってください。さらに、WebDriverAgentRunner プロジェクトで Preferences ウィンドウを開き、**General** タブの **Automatically manage signing** チェックボックスがオフになっていることを確認します。

9. 省略可能 : `infoservice.properties` ファイルには、Silk Test Information Service のポートや、Mac 上でのすべてのテストの実行で使用されるケイパビリティを指定することができます。
 詳細については、「[Silk Test Information Service プロパティの編集](#)」を参照してください。

パーソナル チーム プロファイルを使用した物理 iOS デバイス上でのテスト

Apple Developer Program のメンバーシップに登録していない場合は、パーソナル チーム プロファイルを使用して、物理 iOS デバイス上のアプリケーションをテストできます。

1. Mac 上で、 /Application/Silk/Mobile/common/Appium/node_modules/appium-xcuitestdriver/ WebDriverAgent に移動します。
2. WebDriverAgent.xcodeproj プロジェクトを Xcode で開きます。
3. **TARGETS** リストから、WebDriverAgentLib ターゲットを選択します。
 - a) **General** タブをクリックします。
 - b) **Automatically manage signing** を選択します。
 - c) 開発チームを選択します。

Signing Certificate は自動的に選択されます。
4. **TARGETS** リストから、WebDriverAgentRunner ターゲットを選択します。
 - a) **General** タブをクリックします。
 - b) **Automatically manage signing** を選択します。
 - c) 開発チームを選択します。

Signing Certificate は自動的に選択されます。
5. WebDriverAgentRunner ターゲットに対するプロビジョニング プロファイルの作成に Xcode が失敗する場合は、ターゲットのバンドル ID を手動で変更します。
 - a) **Build Settings** タブをクリックします。
 - b) **Product Bundle Identifier** の値を Xcode が受け付ける適切な値に変更します。
 たとえば、**Product Bundle Identifier** が *com.facebook.WebDriverAgentRunner* の場合、*io.appium.WebDriverAgentRunner* や *io.borland.WebDriverAgentRunner* などに変更します。
 - c) **General** タブをクリックします。

これでターゲットにプロビジョニング プロファイルが設定されました。
6. WebDriverAgent.xcodeproj プロジェクトを保存します。
7. すべてが期待通り動作することを確認するために、プロジェクトをビルドします。


```
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=<udid>' test IPHONEOS_DEPLOYMENT_TARGET=10.3
```
8. WebDriverAgent アプリの再インストール時に問題が発生する場合は、同じプロビジョニング プロファイルを使ってデバイスに追加のアプリを永続的にインストールします。たとえば、WebDriverAgent Xcode プロジェクトの IntegrationApp をインストールします。
 - a) **TARGETS** リストから、IntegrationApp ターゲットを選択します。
 - b) **General** タブをクリックします。
 - c) **Automatically manage signing** を選択します。
 - d) 開発チームを選択します。
9. IntegrationApp ターゲットに対するプロビジョニング プロファイルの作成に Xcode が失敗する場合は、上記の WebDriverAgentRunner ターゲットに対する対処と同じ手順で、ターゲットのバンドル ID を手動で変更します。
- 10 IntegrationApp ターゲットの設定に成功したら、IntegrationApp を物理 iOS デバイスにインストールして実行します。
 - a) ターゲットと iOS デバイスを選択します。
 - b) **Play** をクリックします。

アプリのデバイスへのインストールが成功しても、次のようなエラー メッセージがコンソールや Appium ログ ファイルに出力される場合があります。

```
2017-01-24 09:02:18.358 xcodebuild[30385:339674] Error
Domain=com.apple.platform.iphones Code=-12 "Unable to launch
com.apple.test.WebDriverAgentRunner-Runner" UserInfo={NSLocalizedString=Unable to
launch com.apple.test.WebDriverAgentRunner-Runner, NSUnderlyingError=0x7fa839cadc60
{Error Domain=DTXMessage Code=1 "(null)" UserInfo={DTXExceptionKey=The operation
couldn't be completed. Unable to launch com.apple.test.WebDriverAgentRunner-Runner because
it has an invalid code signature, inadequate entitlements or its profile has not been explicitly
trusted by the user. : Failed to launch process with bundle identifier
```

```
'com.apple.test.WebDriverAgentRunner-Runner'}}} 2017-01-24 09:02:18.358
xcodebuild[30385:339674] Error Domain=IDETestOperationsObserverErrorDomain Code=5
"Early unexpected exit, operation never finished bootstrapping - no restart will be attempted"
UserInfo={NSLocalizedString=Early unexpected exit, operation never finished bootstrapping
- no restart will be attempted} Testing failed: Test target WebDriverAgentRunner encountered an
error (Early unexpected exit, operation never finished bootstrapping - no restart will be
attempted)
```

この問題は、開発者がそのデバイスで信頼されていないため発生します。そのデバイス上でアプリを手動で実行しようとする、**信頼されていない開発元** メッセージが表示される場合があります。

デバイス上でこの問題を解決するには、**設定 > 一般 > プロファイル** または **設定 > 一般 > デバイス管理** (デバイスの種類と iOS のバージョンによって異なります) に移動します。そして、開発元を信頼し、アプリの実行を許可します。

Silk Test Information Service プロパティの編集

infoservice.properties ファイルを使用して、Silk Test Information Service のポートや、HTTPS によるセキュアな接続を使用するかどうかを指定したり、Silk Test Information Service を実行しているマシン上で Silk Test がテストを実行するたびに毎回適用されるケイパビリティ (Capabilities) を指定できます。

1. infoservice.properties.sample ファイルがあるディレクトリに移動します。
 - Windows マシン上で、%PROGRAMDATA%\Silk\SilkTest\conf に移動します (C:\ProgramData\Silk\SilkTest\conf など)。
 - macOS 上で、~/silk/silktest/conf/ に移動します。
2. infoservice.properties.sample ファイルの名前を infoservice.properties に変更します。
3. Silk Test Workbench と Information Service との通信に、HTTPS によるセキュアな接続を使用するかどうかを指定します。
 - HTTPS によるセキュアな通信を使用する場合は、infoservice.https.enabled を true に設定します。これは、デフォルトの設定です。
 - HTTPS によるセキュアな通信を使用しない場合は、infoservice.https.enabled を false に設定します。
4. 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、infoservice.default.https.port を設定します。

デフォルトの HTTPS ポートは 48561 です。ポート番号は、1 から 65535 の間の任意の数値を指定できます。
5. 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合に、すべての HTTP リクエストを HTTPS ポートにリダイレクトするには、infoservice.http-to-https.enabled を true に設定します。

デフォルト値は、false です。
6. 省略可能 : HTTPS によるセキュアな通信を使用しないように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、infoservice.default.port を設定します。

デフォルトのポートは 22901 です。
7. 省略可能 : Silk Test が HTTPS 接続で使用する証明書をユーザーの証明書で置き換える場合は、「[Information Service との HTTPS 接続に使用する証明書の置換](#)」を参照してください。
8. ケイパビリティを指定するには、次の行を infoservice.properties ファイルに追加します。

```
customCapabilities=<custom_capability_1>;<custom_capability_2>;...
```

例：指定した言語での iOS シミュレータの実行

Mac 上で iOS シミュレータを常に同じ言語で実行するには、カスタム ケイパビリティ *language* および *locale* を指定します。たとえば日本語の場合、次の行を `infoservice.properties` ファイルに追加します。

```
customCapabilities=language=ja;locale=ja_JP
```

Silk Test Information Service を Mac からアンインストールする

Mac 上の Apple Safari に対するテストを実行する必要がなくなった場合など、次の手順で Silk Test Information Service を Mac からアンインストールすることができます。

1. `uninstallInfoService.sh` のような新しいシェル ファイルを作成します。
2. 新しいファイルに以下のコードを入力します。

```
#!/bin/sh

if launchctl list | grep com.borland.infoservice ; then
  launchctl unload /Library/LaunchAgents/com.borland.infoservice.plist
  echo "unloading Launch Daemon"
fi

if [ -d "/Applications/Silk" ]
then
  sudo rm -rf /Applications/Silk
fi

if [ -f "/Library/LaunchAgents/com.borland.infoservice.plist" ]
then
  sudo rm /Library/LaunchAgents/com.borland.infoservice.plist
fi

if [ -f "/usr/local/bin/ideviceinstaller" ]
then
  sudo rm /usr/local/bin/ideviceinstaller
fi

exit 0
```

3. コマンドラインで `chmod +x uninstallInfoService.sh` を実行し、シェル ファイルの実行可能にします。
4. コマンドラインからシェル ファイルを実行します。

iOS デバイスの推奨設定

Silk Test Workbench を使用したテストを最適化するために、テストしたい iOS デバイスで次の設定を行ってください。

- 実際にユーザーが行った操作をテストに反映させるために、Apple Safari の自動入力とパスワードの保存を無効化します。設定 > Safari > パスワードと自動入力をタップし、ユーザー名とパスワード設定をオフにします。
- iOS デバイスのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、設定 > 一般 > パスコードロックを選択します。

XCUITest を使用した iOS 上の既存のスク립トの実行

 **注目:** 以前のバージョンの Silk Test Workbench では、*Instruments* を使用して iOS デバイスを自動化していました。iOS 9.3 から、*Instruments* のサポートは XCUITest フレームワークのサポートに

Apple によって置き換えられたため、Silk Test Workbench でも *Instruments* をサポートできなくなりました。この変更により、既存の iOS テスト スクリプトを Silk Test Workbench の現在のバージョンで実行できなくなる場合があります。

- XCUITest における `classname` 属性の動作は Instruments での動作と異なります。ほとんどの場合、Silk Test Workbench はこの変更に対応します。しかし、このような `classname` 属性が原因で既存のテスト スクリプトが動作しなくなった場合には、対応するオブジェクトの新しいロケータを記録する必要があります。
- オブジェクトの階層が変更されました。

インストール済みアプリのテスト

デバイス、エミュレーター、またはシミュレータ上に既にインストールされているネイティブ モバイル アプリをテストするには、接続文字列でアプリを指定します。

1. ネイティブ モバイル アプリをテストする既存のプロジェクトを開きます。
2. **アプリケーション構成の編集** ダイアログ ボックスを開きます。
3. 次のいずれかの方法で、既存のアプリの指定部分を接続文字列で置き換えます。
 - iOS アプリをテストする場合は、`bundleId` を使ってアプリを指定します。たとえば、`app=MyApp.ipa` を `bundleId=silktest.Insurancemobile` で置き換えます。
 - Android アプリをテストする場合は、`appActivity` と `appPackage` を使用してアプリを指定します。たとえば、`app=MyApp.apk` を `appActivity=.LoginActivity;appPackage=silktest.insurancemobile` で置き換えます。

詳細については、「[接続文字列](#)」を参照してください。

モバイル アプリケーションの記録

Silk Test Workbench とモバイル デバイスまたはエミュレータとの間の接続が一旦確立すると、デバイス上で実行する操作を記録できます。モバイル アプリケーションを記録するために、Silk Test Workbench では次の機能を持つ **記録** ウィンドウを使用します。

- テストするモバイル デバイスまたは Android エミュレータの画面を表示します。
- **記録** ウィンドウで操作を実行すると、モバイル デバイス上でも同じ操作が実行されます。
- 画面上のコントロールを操作すると、**記録** ウィンドウはデフォルトの操作を事前に選択します。
 - デフォルトの操作が Click の場合、コントロール上で左クリックすると、その操作が実行されます。右クリックすると、コントロールに対して利用可能な操作のリストが表示されます。この場合は、実行する操作を選択し、**OK** をクリックします。
 - デフォルトの操作が Click ではない場合、コントロールに対して有効なすべての操作がリストで表示されるので、実行したい操作を選択するか、単に **OK** をクリックして事前に選択された操作を受け入れます。

リストから操作を選択する場合、選択した操作のパラメータの値をパラメータ フィールドに入力することができます。Silk Test Workbench は自動的にパラメータを検証します。

- 記録中、Silk Test Workbench は記録ウィンドウの隣にマウスの位置を表示します。その表示をクリックすると、デバイス画面に絶対的な位置とアクティブ オブジェクトに相対的な位置を切り替えることができます。
- 記録を一時停止すると、画面上での操作は記録されないため、デバイスを記録を続けたい状態に変更することができます。
- 記録を停止すると、記録した操作でスクリプトが生成されるため、続いてテストの再生を行うことができます。

テストを再生するモバイル デバイスの選択

テストを再生するために使用するモバイル デバイスを定義できます。

- Silk Test Workbench の UI からテストを実行する場合、**モバイルデバイスの選択** ダイアログ ボックスが表示され、このダイアログ ボックスで選択したモバイル デバイス、Android エミュレータ、iOS シミュレータが使用され、テスト スクリプトで設定されているモバイル デバイス Silk Test Workbench は無視します。
- **モバイルデバイスの選択** ダイアログ ボックスが無効の場合 (**再び表示しない** チェック ボックスをオンにした場合)、個々のテスト スクリプトのアプリケーション構成によって、テストを実行するために使用するモバイル デバイスが決定されます。



注: モバイルデバイスの選択 ダイアログ ボックスを再び有効にするには、**ツール > オプション** をクリックして、**モバイル構成ダイアログの表示** オプションを **はい** に設定します。

- スクリプトをコマンド ラインや CI サーバーから実行する場合は、スクリプトのアプリケーション構成で接続文字列を指定します。

アプリケーション構成で指定したモバイル デバイスを上書きするには、mobiledevice パラメータを使用します。詳細については、STW.exe コマンド ラインの「パラメータ」トピックを参照してください。

- Silk Central からテストを実行する場合、接続文字列を指定する代わりに、Silk Central の実行定義の **配置** タブにある **モバイルデバイスの選択** 領域でモバイル デバイスを指定します。詳細については、『[Silk Central ヘルプ](#)』を参照してください。

デバイス プールなどがある場合に、特定のモバイル デバイスを指定したり、利用可能なデバイス群のサブセットをフィルタするために接続文字列を使用できます。最初に一致したデバイスが再生に使用されます。特に指定がない場合には、次のルールに従って一致したモバイル デバイスが使用されます (高い優先度順)。

- リモート ロケーションに接続されたモバイル デバイスよりも、ローカル マシンに接続されたモバイル デバイスが優先されます。
- ブラウザーの種類が接続文字列で指定されている場合、古いバージョンのブラウザよりも、新しいバージョンのブラウザが優先されます。
- 古いプラットフォームよりも、新しいプラットフォームが優先されます。
- 物理デバイスがエミュレータやシミュレータよりも優先されます。
- アルファベット順で後者のデバイス名のデバイスが優先されます。たとえば、"iphone 5" という名前のデバイスよりも、"iphone 6" という名前のデバイスが優先されます。

例 : リモート マシンに接続されている Android デバイス上のアプリに対する接続文字列

リモート マシンに接続されている Android デバイス上で MyApp.apk アプリをテストするには、接続文字列は次のようになります。

```
"platformName=Android;deviceName=MotoG3;host=http://10.0.0.1;app=MyApp.apk"
```

例 : Mac の iOS シミュレータ上のアプリに対する接続文字列

```
"platformName=iOS;platformVersion=10.0;deviceName=iPhone6;host=10.0.0.1;app=MyApp.ipa;isSimulator=true"
```

Mobile Center デバイスの使用


Mobile Center は、モバイル デバイスのテストを管理するためのモバイル用ゲートウェイです。

Mobile Center によって管理されたデバイスに Silk Test Workbench からアクセスするには、次の手順を実行します。

1. Silk Test Workbench と Silk Central を統合します。


詳細については、「[Silk Test Workbench と Silk Central の統合](#)」を参照してください。

2. Mobile Center を使用するように Silk Central を設定します。


 **注:** Mobile Center のインストール時に、適切なバージョンの Android SDK が使用されていることを確認してください。Silk Test Workbench で同じバージョンを使用するように設定するには、環境変数 `SILK_ANDROID_HOME` に `C:\¥Users¥<ユーザー>\¥AppData¥Local¥Android¥android-sdk` などを指定します。詳細については、『*Silk Central* ヘルプ』を参照してください。


3. iOS 上でテストする場合は、次の IPA ファイルがサインされていることを確認します。


- HP4M-Agent.ipa
- HPMC-AgentLauncher.ipa
- WebDriverAgentRunner-Runner.ipa

 **注:** Silk Test Workbench は、Mobile Center を介した iOS シミュレータのテストをサポートしません。

以上により、**アプリケーションの選択** ダイアログに Mobile Center デバイスが表示されます。テストするデバイスを選択してください。

 **注:** Silk Test Workbench と Mobile Center の両方で同時に同じモバイル デバイスをテストできません。Silk Test Workbench を使用してテストしたモバイル デバイスを、Mobile Center から引き続きテストする場合は、そのデバイスを再起動します。

 **注:** Mobile Center 上で管理されているデバイスをテストする場合は、Silk Test Workbench は TypeKeys メソッドや SetText メソッドを使用して **Enter** などのキー コードの入力をサポートしません。さらに、Silk Test Workbench は iOS デバイス上の **ホーム** ボタンの押下をサポートしません。

 **注:** Android エミュレータ上でテストする場合は、GPU HW アクセラレーションを無効にしてください。

Sauce Labs デバイスの使用

Sauce Labs は、自動テストプラットフォームを提供しており、さまざまなモバイル デバイスやモバイルプラットフォームのバージョン上でテストを行うことができます。自分自身のインフラストラクチャとしてデバイスを購入、保守する必要はありません。


Sauce Labs デバイスには Silk Central を介してアクセスすることができます。次の手順に従って設定してください。

1. Silk Test Workbench が Silk Central と統合されていることを確認してください。
詳細については、「*Silk Test Workbench* と *Silk Central* の統合」を参照してください。
2. Silk Central が Sauce Labs を使用するように設定されていることを確認してください。
詳細については、*Silk Central* ヘルプ を参照してください。

以上により、**アプリケーションの選択** ダイアログに Sauce Labs デバイスが表示されます。テストするデバイスを選択してください。

モバイル デバイスの接続文字列

接続文字列 は、テストに使用するモバイル デバイスを指定します。モバイル テストを実行する場合、Silk Test Workbench は接続文字列を使用してモバイル デバイ스에接続します。接続文字列は、アプリケーション構成の主要な一部です。テスト対象アプリケーションを構成するときに、接続文字列は設定されます。接続文字列を変更するには、**アプリケーション構成の編集** ダイアログ ボックスを使用します。

 **注:** Silk Central からテストを実行する場合、接続文字列を指定する代わりに、Silk Central の実行定義の **配置** タブにある **モバイル デバイスの選択** 領域でモバイル デバイスを指定します。詳細については、『*Silk Central* ヘルプ』を参照してください。

デバイス プールなどがある場合に、特定のモバイル デバイスを指定したり、利用可能なデバイス群のサブセットをフィルタするために接続文字列を使用できます。最初に一致したデバイスが再生に使用されま

す。特に指定がない場合には、次のルールに従って一致したモバイル デバイスが使用されます（高い優先度順）。

- リモート ロケーションに接続されたモバイル デバイスよりも、ローカル マシンに接続されたモバイル デバイスが優先されます。
- ブラウザーの種類が接続文字列で指定されている場合、古いバージョンのブラウザよりも、新しいバージョンのブラウザが優先されます。
- 古いプラットフォームよりも、新しいプラットフォームが優先されます。
- 物理デバイスがエミュレータやシミュレータよりも優先されます。
- アルファベット順で後者のデバイス名のデバイスが優先されます。たとえば、"iphone 5"という名前のデバイスよりも、"iphone 6"という名前のデバイスが優先されます。

次のコンポーネントが接続文字列で使用できます。


コンポーネント	説明
deviceName	モバイル デバイスの名前。物理モバイル デバイス上でテストをする場合、デバイス ID を代わりに使用します。ワイルドカードをサポートします。大文字と小文字は区別されません。
platformName	Android または iOS。必須。
deviceId	省略可能：モバイル デバイスの ID。物理モバイル デバイスでテストをする場合に、デバイス名の代わりに使用します。ワイルドカードをサポートします。大文字と小文字は区別されません。
platformVersion	省略可能：Android または iOS のバージョン。特定の Android または iOS のバージョンのモバイル デバイス上でのみテストする場合に、バージョンを指定します。ワイルドカードをサポートします。大文字と小文字は区別されません。
browserVersion	省略可能：特定のブラウザのバージョンでのみテストする場合に、ブラウザの種類と共に使用します。ワイルドカードをサポートします。大文字と小文字は区別されません。
host	省略可能：設定しない場合は、任意のリモート ロケーションがホストとして使用されます。ワイルドカードをサポートします。大文字と小文字は区別されません。
<ul style="list-style-type: none"> • app • appActivity • appPackage 	Android 上のネイティブ モバイル アプリケーションのテストでは必須です。アプリへのフルパス、または <i>appActivity</i> と <i>appPackage</i> の組み合わせで指定します。たとえば、 app=MyApp.apk、 appActivity=.LoginActivity;appPackage=silktest.insurancemobile などです。
<ul style="list-style-type: none"> • app • bundleId 	iOS 上のネイティブ モバイル アプリケーションのテストでは必須です。アプリへのフルパス、または <i>bundleId</i> で指定します。たとえば、app=MyApp.ipa、 bundleId=silktest.InsuranceMobile などです。
noReset	省略可能：ネイティブ モバイル アプリケーションをテストする場合に設定できます。 <i>app</i> が指定されている場合にのみ有効です。テストの前にアプリを再インストールしない場合は True。テストの前にアプリを再インストールする場合は False を指定します。デフォルト値は、False です。
isSimulator	省略可能：iOS シミュレータ上でのみテストを実行する場合に指定します。デバイス名を代わりに使用できます。
isPhysicalDevice	省略可能：物理デバイス上でのみテストを実行する場合に指定します。デバイス名を代わりに使用できます。

デバイス プールを使用して、テストで実際に使用されるデバイスを確認するには、MobileDevice クラスの GenerateConnectionString メソッドの戻り値を使用できます。

モバイル デバイスまたは Android エミュレータ上のモバイル Web アプリケーションのテスト

モバイル デバイスまたは Android エミュレータ上でモバイル Web アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. モバイル デバイス名 (MotoG3 など)、またはデバイス ID (11111111 など)。

 **注:** デバイス ID は可読性に欠けるため、デバイス名がユニークであれば、デバイス名を接続文字列に使用することを Micro Focus では推奨します。

2. プラットフォーム名。
3. ブラウザーのバージョン。ブラウザーの種類の設定との組み合わせでのみ使用されます。
4. 特定のリモート マシンの IP アドレスまたはホスト名 (10.0.0.1 など)。**リモート ロケーションの編集** ダイアログ ボックスで指定したリモート ロケーションの名前をホスト名として使用することもできます (*MyRemoteLocation* など)。リモート ロケーション名を使用する場合に、ワイルドカードを使用することもできます。ローカル マシンに接続されている Android デバイスをテストする場合は、ローカル マシンの IP アドレスまたはホスト名を指定します。

例 : 利用可能な任意の Android デバイスの接続文字列

```
"platformName=Android"
```

例 : ローカル マシンに接続されている Android デバイス上のブラウザーに対する接続文字列

ローカル マシンに接続されている Android デバイス上でモバイル ブラウザーをテストするには、接続文字列は次のようになります。

```
"deviceName=MotoG3;platformName=Android;host=localhost"
```

または

```
"platformName=Android;deviceId=11111111;host=localhost"
```

例 : リモート マシンに接続されている Android デバイス上のブラウザーに対する接続文字列

リモート Android デバイス上のモバイル ブラウザーをテストするには、接続文字列は次のようになります。

```
"deviceName=MotoG3;platformName=Android;host=10.0.0.1"
```

```
"deviceName=MotoG3;platformName=Android;host=MyRemoteLocation*"
```

例 : Mac に接続されている iOS デバイス上のブラウザーに対する接続文字列


リモート iOS デバイス上のモバイル ブラウザーをテストするには、接続文字列は次のようになります。

```
"deviceName=myiPhone6;platformName=iOS;host=10.0.0.1"
```

モバイル デバイスまたは Android エミュレータ上のネイティブ モバイル アプリケーションのテスト

モバイル デバイスまたは Android エミュレータ上でネイティブ モバイル アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. モバイル デバイス名 (MotoG3 など)、またはデバイス ID (11111111 など)。

 **注:** デバイス ID は可読性に欠けるため、デバイス名がユニークであれば、デバイス名を接続文字列に使用することを Micro Focus では推奨します。

2. プラットフォーム名。
3. 特定のリモート マシンの IP アドレスまたはホスト名 (10.0.0.1 など)。**リモート ロケーションの編集** ダイアログ ボックスで指定したリモート ロケーションの名前をホスト名として使用することもできます (*MyRemoteLocation* など)。リモート ロケーション名を使用する場合に、ワイルドカードを使用することもできます。ローカル マシンに接続されている Android デバイスをテストする場合は、ローカル マシンの IP アドレスまたはホスト名を指定します。

4. テストするアプリのファイルの名前または、ファイルが Web サーバー上にある場合には、ファイルの URL。たとえば、C:/MyApp.apk や MyApp.ipa など。

- Android アプリは、常に .apk ファイルを指定します。
- 物理デバイス上の iOS アプリは、常に .ipa ファイルを指定します。
- シミュレータ上の iOS アプリは、ZIP ファイルまたは、*app* という名前のディレクトリを指定します。

例：リモートマシンに接続されている Android デバイス上のアプリに対する接続文字列

リモートマシンに接続されている Android デバイス上で MyApp.apk アプリをテストするには、接続文字列は次のようになります。

```
"platformName=Android;deviceName=MotoG3;host=http://10.0.0.1;app=MyApp.apk"
```

例：Mac に接続されている iOS デバイス上のアプリに対する接続文字列

リモートマシンに接続されている iOS デバイス上で MyApp.ipa アプリをテストするには、接続文字列は次のようになります。

```
"platformName=iOS;deviceName=MyiPhone;host=http://10.0.0.1;app=MyApp.ipa"
```

iOS シミュレータ上のモバイル Web アプリケーションのテスト

iOS シミュレータ上でモバイル Web アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. プラットフォーム名 (iOS)。
2. プラットフォームのバージョン (10.0 など)。
3. モバイル デバイス名 (iPhone6 など)。
4. iOS シミュレータを実行している Mac の IP アドレスまたはホスト名。

例：Mac の iOS シミュレータ上のブラウザーに対する接続文字列

```
"platformName=iOS;platformVersion=10.0;deviceName=iPhone6;host=10.0.0.1;isSimulator=true"
```

iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト

Mac の iOS シミュレータ上でネイティブ モバイル アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. プラットフォーム名 (iOS)。
2. プラットフォームのバージョン (10.0 など)。
3. モバイル デバイス名 (iPhone6 など)。
4. リモートマシンの IP アドレスまたはホスト名 (10.0.0.1 など)。
5. テストするアプリの名前 (MyApp.ipa など)。

例：Mac の iOS シミュレータ上のアプリに対する接続文字列

```
"platformName=iOS;platformVersion=10.0;deviceName=iPhone6;host=10.0.0.1;app=MyApp.ipa;isSimulator=true"
```

モバイルデバイスの操作

モバイル デバイスを操作したり、テスト対象アプリケーションでスワイプのような操作を実行するには、次の手順を実行します。

1. **記録** ウィンドウで、**モバイル デバイス操作の表示** をクリックします。モバイル デバイスに対して実行できるすべての操作がリストされます。
2. リストからリストから実行したい操作を選択します。
3. Android デバイスまたはエミュレータで、スワイプを記録するには、マウスの左ボタンを押したままマウスを動かします。
4. テストの記録を続行します。

モバイルデバイスの開放

モバイル デバイスに対するテストを記録または再生する場合、Open Agent インスタンスはそのデバイスの所有権を確保します。これによって、Open Agent は、他の Silk Test ユーザーがそのデバイスを使用することを防止します。デバイス上でのテストの記録や再生が終了した後に、他の Silk Test ユーザーがデバイスを使用できるようにするために、Silk Test クライアントが閉じたとき、無人のテスト プロセスが完了したとき、または Open Agent が閉じたときに、Silk Test は自動的にデバイスを開放します。また、デバイスを手動で開放することもできます。



注: モバイルデバイスを開放すると、モバイル デバイス上のテスト対象アプリケーション (AUT) は閉じられます。

記録後のモバイルデバイスの開放

他の Silk Test ユーザーがモバイル デバイスでテストできるようにするために、記録後にデバイスを開放します。

記録が完了した後にモバイル デバイスを開放するには、次のいずれかを実行します。

- システム トレイから Open Agent を停止する。
- Silk Test Workbench を閉じる。並列テストが有効な場合には、この操作でのみデバイスが開放されます。



注: モバイルデバイスを開放すると、モバイル デバイス上のテスト対象アプリケーション (AUT) は閉じられます。

再生後のモバイルデバイスの開放

他の Silk Test ユーザーがモバイル デバイスでテストできるようにするために、再生後にデバイスを開放します。

再生が完了した後に手動でモバイル デバイスを開放するために、次のいずれかを実行することもできます。

- モバイル Web アプリケーションのテストを再生した場合、BrowserApplication クラスの Close メソッド、または CloseSynchron メソッドを使用します。これらのメソッドの詳細については、API ドキュメントを参照してください。

```
'VB .NET code
WebBrowser.Close()
```

- モバイル ネイティブ アプリケーションのテストを再生した場合、MobileDevice クラスの CloseApp メソッドを使用します。

たとえば、次のように入力します。

```
'VB .NET code
Dim mobileDevice = _desktop.MobileDevice()
mobileDevice.CloseApp()
```

- VB .NET テストの再生後にデバイスを開放するには、`Agent.DetachAll()` ステートメントをテスト スクリプトに追加します。

次の条件を満たしている場合は、モバイル デバイスは自動的に開放されます。

- Open Agent を閉じる。
- 無人テスト中にテスト プロセスが停止する。並列テストが有効な場合には、この操作でのみデバイスが開放されます。
- Silk Test Workbench を閉じる。並列テストが有効な場合には、この操作でのみデバイスが開放されます。



注: モバイル デバイスを開放すると、モバイル デバイス上のテスト対象アプリケーション (AUT) は閉じられます。

Silk Central によって管理されているデバイスの使用

モバイル テストを管理し、構成テストを実行するために、Silk Central を使用できます。

Silk Test Workbench と Silk Central を統合します。

Silk Test Workbench と Silk Central の統合についての詳細は、「*Silk Test Workbench と Silk Central の統合*」を参照してください。

Silk Central の特定のユーザーに対して構成されたデバイス プロバイダーが、**アプリケーションの選択**、**ブラウザーの選択**、**モバイル デバイスの選択** ダイアログ ボックスで、リモート ブラウザーやモバイル デバイスのリストに表示されます。Silk Test Workbench は、Silk Central によって管理されているデバイスを、その名前と、Silk Central で構成されたデバイス プロバイダーのホスト名/URL と共に表示します。

モバイル アプリケーションのテスト時のトラブルシューティング

[アプリケーションの選択] ダイアログにモバイル デバイスが表示されない理由

Silk Test Workbench がモバイル デバイスやエミュレータを認識できないと、**アプリケーションの選択** ダイアログの **モバイル** タブにはデバイスやエミュレータが表示されません。さらに、**アプリケーションの選択** ダイアログの **Web** タブにも、そのデバイスやエミュレータ上にインストールされているモバイル ブラウザーが表示されません。

Silk Test Workbench が、次の何れかが原因でモバイル デバイスまたはエミュレータを認識していない可能性があります。

原因	解決策
エミュレータが実行されていない。	エミュレータを開始します。
Android Debug Bridge (adb) がモバイル デバイスを認識しない。	<p>モバイル デバイスが adb によって認識されているかどうか確認するには：</p> <ol style="list-style-type: none"> 1. Android SDK をインストールしたフォルダで、Android Debug Bridge (adb) がある場所に移動します。Android SDK がインストールされていない場合、<code>C:\Program Files (x86)\Silk\SilkTest\ng\Mobile\windows\AndroidTools\platform-tools</code> に移動して、Silk Test Workbench がインストールした adb を使用します。 2. Shift を押しながら、ファイル エクスプローラ ウィンドウで右クリックします。

原因	解決策
	<ol style="list-style-type: none"> 3. コマンド ウィンドウをここで開く を選択します。 4. コマンド ウィンドウで、adb devices を入力して、アタッチしたすべてのデバイスのリストを得ます。 5. デバイスがリストされない場合、USB デバッグがデバイスで有効化されていること、および適切な USB ドライバがインストールされていることを確認します。 6. 「adb server is out of date」のようなエラーが発生する場合は、C:¥Program Files (x86)¥Silk ¥SilkTest¥ng¥Mobile¥windows¥AndroidTools ¥platform-tools の adb のバージョンがローカルの Android SDK の adb のバージョンと一致していることを確認してください。詳細については、「Open Agent とデバイスとの接続が不安定な場合の対処方法」を参照してください。
デバイスのオペレーティング システムのバージョンを Silk Test Workbench がサポートしていない。	サポートするモバイル オペレーティング システムのバージョンについては、 リリース ノート を参照してください。
デバイスの USB ドライバがローカル マシンにインストールされていない。	デバイスの USB ドライバをローカル マシンにインストールしてください。詳細については、「 USB ドライバをインストールする 」を参照してください。
USB デバッグがデバイスで有効化されていない。	USB デバッグをデバイスで有効化してください。詳細については、「 USB デバッグの有効化 」を参照してください。



注: どの解決策を適用しても解決できない場合は、デバイスを再起動してみてください。

URL に移動せずに Silk Test Workbench が Chrome for Android で URL を検索する理由

アドレスバーに入力された URL を、Chrome for Android が検索として解釈する場合があります。回避策として、URL に移動するコマンドをスクリプトに手動で追加できます。

adb サーバーが正しく起動しない場合にすべきこと

Android Debug Bridge (adb) サーバーが開始するとき、ローカル TCP ポート 5037 にバインドし、adb クライアントから送信されてくるコマンドをリッスンします。すべての adb クライアントは、ポート 5037 を使用して、adb サーバーと通信します。adb サーバーは、5555 から 5585 の範囲 (エミュレータやデバイスで使用される範囲) で奇数のポートをスキャンしてエミュレータやデバイス インスタンスを探します。adb はこれらのポートの変更を許しません。adb 開始中に問題が発生した場合、これらの範囲のポートの 1 つが、他のプログラムによって既に使用されているかどうか確認します。

詳細については、<http://developer.android.com/tools/help/adb.html> を参照してください。

Open Agent とデバイスとの接続が不安定な場合の対処方法

Android SDK、または Android Debug Bridge (adb) を使用するその他のツールをインストールしている場合、Silk Test Workbench が使用する adb サーバー以外のサーバーが実行中の可能性があります。バージョンの異なる adb サーバーが実行中の場合、Open Agent とデバイスとの接続が不安定になったり、接続できない場合があります。

このようなバージョンの不一致によるエラーを避けるには、環境変数 `SILK_ANDROID_HOME` に Android SDK ディレクトリへのパス (C:¥Users¥<ユーザー>¥AppData¥Local¥Android¥android-sdk など) を指定してください。Information Service が実行中の場合、Windows のサービス マネージャーを使用して、Silk Test Information Service を再起動して更新した環境変数を適用する必要があります。環境変数

が設定されていない場合は、Silk Test Workbench は Silk Test Workbench に同梱されたバージョンの adb を使用します。

エラー「メモリの割り当てに失敗しました : 8」が発生する理由

エミュレータを開始しているときに、システムが十分なメモリを割り当てることができない場合に、このエラーが表示されます。以下を行ってみてください。

1. エミュレータのメモリ オプションの RAM サイズを下げる
2. Intel HAXM の RAM サイズを下げる RAM サイズを下げるには、IntelHaxm.exe を再度実行して、**Change** を選択します。
3. **タスク マネージャ** を開き、十分なフリー メモリが利用可能かどうかを確認します。不足している場合、プログラムを閉じてメモリを開放してください。

iOS デバイスのテスト時に「Silk Test は指定したアプリを開始できません」というエラーが発生する理由

このエラーが発生する原因として、以下の理由が考えられます。

原因	解決策
iOS デバイスがデベロッパ モードになっていない。	次の 2 種類の方法のいずれかで、デベロッパ モードを有効化できます。 <ul style="list-style-type: none">• Xcode がインストールされている Mac にデバイスを接続し、テストするアプリをデバイスで開始します。• プロビジョニング プロファイルをデバイスに追加します。 <ol style="list-style-type: none">1. Xcode を開きます。2. Window > Devices を選択します。3. iOS デバイスを右クリックします。4. Show Provisioning Profiles を選択します。5. プロビジョニング プロファイルを追加します。
デバイスの iOS のバージョンを最近更新した。	<ol style="list-style-type: none">1. Xcode を開きます。2. Window > Devices を選択します。3. Xcode がシンボル ファイルを処理するまで待機します。
UI オートメーションが iOS デバイスで有効化されていない。	<ol style="list-style-type: none">1. 設定 > デベロッパ を選択します。2. Enable UI Automation をオンにします。
Web インスペクタ が iOS デバイスで有効化されていない (モバイル Web アプリケーションのテストの場合)。	<ol style="list-style-type: none">1. 設定 > Safari > 詳細 をクリックします。2. Web インスペクタ をオンにします。
テストするアプリがテストしようとしている iOS デバイスの iOS バージョン用にビルドされていない。	Xcode を使用してデバイスの iOS バージョン用にアプリをビルドします。
ソフトウェア・アップデート ダイアログ ボックスが iOS デバイス上で開いている。	ダイアログ ボックスを閉じ、ソフトウェアの自動アップデートを無効化します。 <ol style="list-style-type: none">1. 設定 > iTunes & App Store > 自動ダウンロード を選択します。2. アップデート をオフにします。

Android デバイスの動的ハードウェア コントロールに戻るボタンだけが表示される理由

テストの開始時に Android デバイスや Android エミュレータの画面がロックされると、デバイスやエミュレータが動的ハードウェア コントロールに戻る ボタンだけを表示する場合があります。

この問題を解決するには、Open Agent を停止し、デバイスを再起動してから、デバイスの設定を画面のロックをしないように設定してください。

Android デバイスまたはエミュレータにキーボードが表示されなくなる理由

Unicode 文字列をサポートするために、Silk Test Workbench は標準キーボードをカスタム キーボードに置き換えます。そして、テストの完了時に元のキーボードに戻します。テスト中にエラーが発生すると、カスタム キーボードが設定されたまま、元に戻らない場合があります。

この問題を解決するには、**設定 > 言語と入力 > 現在のキーボード** を開き、手動で元のキーボードに戻してください。

テスト中にデバイスが応答しなくなる理由

テストの開始時に、デバイス、エミュレータ、シミュレータの画面がロックされると、Silk Test Workbench は画面のロックを解除できず、デバイス、エミュレータ、シミュレータが操作に応答しなくなる場合があります。

この問題を解決するには、Open Agent を停止し、デバイスの設定を画面のロックをしないように設定してください。

Information Service を Mac にインストールできない理由

システム環境設定の **セキュリティとプライバシー** で、**一般** タブの **ダウンロードしたアプリケーションの実行許可** 設定が **Mac App Store と確認済みの開発元からのアプリケーションを許可** (デフォルト値) に設定されている場合、Information Service セットアップを開いているときに次のエラー メッセージが表示されます。

"SilkTestInformationService<バージョン>.pkg" は、開発元が未確認のため開けません。

この問題を解決するには、次のいずれかを行います。

- セットアップ ファイルを右クリックして、**開く** を選択します。警告メッセージが表示されても、ファイルを開くことができます。
- **ダウンロードしたアプリケーションの実行許可** 設定を **すべてのアプリケーションを許可** に設定します。
- ファイルを開いた後、システム環境設定の **セキュリティとプライバシー** の **一般** タブを開き、**このまま開く** をクリックします。

Android アプリの記録時に記録ウィンドウが真っ暗になる理由

金融取引を処理するアプリなど、高レベルなセキュリティを必要とする Android アプリでは、Silk Test Workbench がアプリのキャプチャをできないようにするために、**FLAG_SECURE** フラグが設定されている可能性があります。Silk Test Workbench は、記録時に Android デバイスのスクリーンショットやビデオを利用しますが、テストする Android アプリにこのフラグが設定されていると、**記録中** ウィンドウにはデバイスの真っ黒な画面が表示されます。Silk Test でこのようなアプリをテストするには、テスト中に **FLAG_SECURE** フラグを設定しないよう、アプリの開発チームに依頼してください。

Android エミュレータでのテスト時に Silk Test Workbench がビデオを表示しない理由

エミュレータがレンダリングにコンピュータのグラフィック カードを使用している場合、Silk Test Workbench のビデオ キャプチャが機能しない場合があります。この問題を解決するには、ソフトウェアでグラフィックでエミュレートします。

1. **Android Virtual Device Manager** を開きます。
2. エミュレータの **Actions** 列の **Edit** をクリックします。

3. Virtual Device Configuration ダイアログ **Emulated Performance** 領域で、リストから **Software** を選択します。

クラウド環境でのテスト時に Silk Test Workbench がビデオを表示しない理由

クラウド環境でテストする場合、必要なポートがオープンになっていないことなどが原因で、テストの記録や再生時にビデオの表示が機能しない場合があります。

この問題を解決するには、`infoservice.properties` ファイルに WebDriver ホストの URL リストを指定します。このプロパティ ファイルについての詳細は、「*Silk Test Information Service* プロパティの編集」を参照してください。`infoservice.disableScreencastHosts` オプションをファイルに追加し、次のように入力します。

```
infoservice.disableScreencastHosts=<URL_1>,<URL_2>, ...
```

例：

```
infoservice.disableScreencastHosts=http://my-webdriver-server-url.com:80/wd/hub
```

ワイルドカードとしてアスタリスク (*) を使用して、`*my-webdriver-server-url.com` のような URL パターンを指定することができます。

この設定により、指定したホストでの記録と再生時に、Silk Test Workbench はビデオの代わりに連続したスクリーンショットを表示するようになります。

Xcode のインストール バージョンを変更する方法

Xcode の最新版にアップグレードしてしまった場合など、使用している Xcode のバージョンを Silk Test Workbench がサポートしていない場合、iOS でテストする際にエラー メッセージが表示される場合があります。

インストールした Xcode のバージョンをサポートするバージョンで置換するには、サポートするバージョンの Xcode を <https://developer.apple.com/download/more/> からダウンロードし、サポートされていないバージョンをダウンロードしたバージョンで置換します。サポートする Xcode のバージョンについての情報は、『[リリース ノート](#)』を参照してください。

Mac のディスクの空き容量がなくなった場合の対処

Silk Test Workbench は iOS デバイスの自動化に Instruments を使用します。このツールは、`/Library/Caches/com.apple.dt.instruments` ディレクトリに大きなログ ファイルを生成するため、Mac のディスクの空き容量を圧迫している場合があります。この問題を解決するために、手動であるいは cron ジョブを使用して、これらのログ ファイルを定期的に削除することを Micro Focus は推奨します。たとえば、毎日同じ時間にファイルを削除するには、次の手順を実行します。

1. ターミナルで「`sudo crontab -e`」を入力します。crontab を root として編集できるエディタが開きます。
2. 次の行を crontab に追加します。

```
0 2 1 * * find /Library/Caches/com.apple.dt.instruments -mtime +10 -delete
```
3. crontab を保存します。

この例では、10 日より前のすべてのログ ファイルが、毎日午前 2 時にディレクトリから削除されます。

エラー メッセージ「WebDriver エージェントにテスト用の署名ができません」によってテストが失敗する理由

物理 iOS デバイス上でテストを行う場合、通常このエラーは WebDriverAgent アプリのビルド プロセス中に、プロビジョニング プロファイルでサインされていないか問題があったことを意味します。

デバイスが接続されている Mac マシンで次のコマンドを実行して、実際の問題を確認できます。

```
cd /Applications/Silk/Mobile/common/Appium/node_modules/appium-xcuitest-driver
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=<udid>' test
```

/Applications/Silk/Mobile/common/Appium/node_modules/appium-xcuitest-driver フォルダにある Resources フォルダが存在し、そのフォルダに WebDriverAgent.bundle ファイルがあることを確認します。存在しない場合は、このフォルダを作成し、空の WebDriverAgent.bundle ファイルを作成します。たとえば、次のコマンドを実行します。

```
mkdir -p Resources/WebDriverAgent.bundle
```

Developer Tools Access による他のプロセスの制御の要求を抑える方法

iOS 上でテストの実行を開始すると、つぎのメッセージのようなメッセージ ボックスが表示される場合があります。

Developer Tools Access は、デバッグを続けるためにほかのプロセスを制御する必要があります。これを許可するには、パスワードを入力してください。パスワードを入力して許可します。

このメッセージを抑制するには、ターミナルで次のコマンドを実行します。

```
sudo /usr/sbin/DevToolsSecurity --enable
```

iPad 上のモバイル Web アプリケーションのテスト時に矩形領域がずれる理由

iPad 上のモバイル Web アプリケーションのテスト時にコントロールを囲む矩形領域がずれている場合、複数のブラウザー タブが開いており、タブ バーが表示されている場合があります。この問題を回避するには、1 つを残して、ほかのすべてのタブを閉じてください。

Silk Test Workbench のアップデート後にデバイス上でテストの記録と再生が動作しない理由

Silk Test Workbench の新しいバージョンにアップデートすると、Silk Test Workbench の以前のバージョンでモバイル テスト用に使用してきた物理モバイル デバイス上の Appium アプリも自動的にアップデートされます。何らかの理由でこれらのアプリが自動的にアップデートされないと、そのデバイス上でテストの記録と再生が正常に動作しない場合があります。

Silk Test Workbench のアップデート後に、Android デバイスでこのような問題が発生した場合は、次のアプリをデバイスから手動でアンインストールしてください。

- Appium Android Input Manager
- Appium Settings
- io.appium.uiautomator2.server
- io.appium.uiautomator2.server.test
- Unlock

Silk Test Workbench のアップデート後に、iOS デバイスでこのような問題が発生した場合は、WebDriverAgentRunner をデバイスから手動でアンインストールしてください。

モバイル アプリケーションを記録できない理由

Silk Test Workbench では、Appium を使ってモバイル アプリケーションをテストします。Appium でネットワークプロキシ設定が設定されていると、Silk Test Workbench の記録を妨害する場合があります。モバイル デバイスまたはエミュレータのネットワークプロキシ設定を無効にしてみてください。

Android デバイス上でテストを実行できない理由

一部の Android デバイスでは、Silk Test Workbench などを使ったデバイス上のモバイル アプリケーションのテストができないよう、あらかじめ設定されている場合があります。たとえば、Xiaomi Mi Mix 2 の前提条件は他のデバイスと同様ですが、さらに次の設定が必要になります。Xiaomi Mi Mix 2 でテストを実行する前に、次の手順を実行してください。

1. デバイスの開発者モードを有効にします。
2. **設定 > その他の設定 > 開発者向けオプション** に移動します。
3. **USB デバッグ** を有効にします。
4. **スリープモードにしない** を有効にします。

5. **USB 経由でインストールする** を有効にします。
6. **USB デバッグ** を有効にします。
7. **MIUI の最適化をオンにする** を無効にします。

テストの再生に Chrome for Android を使用する方法

デフォルトでは、**ブラウザーの選択** ダイアログ ボックスを使用して、再生に使用するブラウザーを選択できます。

スクリプトをコマンド ラインや CI サーバーから実行する場合は、スクリプトのアプリケーション構成で接続文字列を指定できます。アプリケーション構成で指定したブラウザーを上書きするには、`silktest.configurationName` 環境変数を使用します。

BrowserApplication クラスの `browsertype` プロパティを使用して、再生に使用するブラウザーの種類を設定することもできます。ただし、`browsertype` は Chrome for Android の明示的な値を含みません。

テストを再生するブラウザーとして Chrome for Android を使用するように指定するには、`browsertype` に `GoogleChrome` を設定し、Android をプラットフォームとして指定します。Android が指定されると、デスクトップ マシン上の Google Chrome の代わりに Chrome for Android を使用して、Silk Test Workbench はテストを実行します。

使用例

次のサンプル コードは、`silktest.configurationName` を使用して Nexus 7 上の Chrome for Android を使用するテストの基本状態を設定する方法を示しています。

SET

```
silktest.configurationName="platformName=Android;deviceName=Nexus 7;host=10.0.0.1 - Chrome"
```

次の のサンプル コードは、`browsertype` を使用して Chrome for Android を使用するテストの基本状態を設定する方法を示しています。

'VB .NET code

```
Dim baseState = New BrowserBaseState(BrowserType.GoogleChrome, "demo.borland.com/InsuranceWebExtJS/")
baseState.SetConnectionString("platformName=Android")
baseState.Execute()
```

モバイル Web アプリケーションのテストにおける制限事項

モバイル ブラウザ上でのテストの再生とロケーターの記録のサポートは、サポートされている他のブラウザほど完全なものではありません。モバイル Web アプリケーションに対するテストの再生とロケーターの記録の既知の制限事項を以下に示します。

- 次のクラス、インターフェイス、メソッド、プロパティは、モバイル Web アプリケーションでは現時点ではサポートされません。
 - BrowserApplication クラス。
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetActiveTab メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド

- GetTabCount メソッド
- ImageClick メソッド
- OpenTab メソッド
- SelectTab メソッド
- DomElement クラス。
 - DomDoubleClick メソッド
 - DomMouseMove メソッド
 - GetDomAttributeList メソッド
- IKeyable インターフェイス。
 - PressKeys メソッド
 - ReleaseKeys メソッド
- Silk Test Workbench は、iOS 上の Apple Safari を使用した HTML フレームおよび iframe のテストをサポートしません (HTML フレームおよび iframe 上のテキスト解決を含む)。

テキスト解決は次のメソッドを含みます。

- TextCapture
- TextClick
- TextExists
- TextRectangle
- 横固定モードでの記録はシステム バーに仮想ボタンを含むエミュレータに対してサポートされません。このようなエミュレータは、回転を正しく検出せずに、横固定モードのシステム バーを画面の下部ではなく画面の右側に配置します。ただし、このようなエミュレータは縦固定モードで記録することができます。
- モバイル アプリケーションに対する XPath 式では、HTML DOM の HTML 属性だけがサポートされません。Silk Test Workbench は、XPath 式のプロパティをサポートしません。
- Android 上でのモバイル Web アプリケーションのテストでは、Silk Test Workbench は、拡大縮小をサポートしません。
- BrowserWindow クラスの以下の JavaScript 警告処理メソッドが、Original Android Stock (AOSP) ブラウザー上でのテストでは機能しません。
 - AcceptAlert メソッド
 - DismissAlert メソッド
 - GetAlertText メソッド
 - IsAlertPresent メソッド
- 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。
- モバイル Web アプリケーションのテストを開始する前に、ブラウザーのタブが開いていないことを確認してください。



ヒント: iPad 上で Apple Safari のタブを無効に出来ます。設定 > Safari を選択して、**タブバーを表示** をオフにすると無効になります。

- モバイル Web アプリケーションのテスト中に、ブラウザーのタブは 1 つだけ開くことができます。
- Silk Test Workbench は、ネイティブ モバイル アプリケーションによって開かれたモバイル Web アプリケーションのテストをサポートしません。

ネイティブ モバイル アプリケーションのテストにおける制限事項

ネイティブ モバイル アプリケーションに対するテストの再生とロケータの記録の既知の制限事項は次の通りです。

- 次のクラス、インターフェイス、メソッド、プロパティは、ネイティブ モバイル アプリケーションでは現時点ではサポートされません。
 - IKeyable インターフェイス。
 - PressKeys メソッド
 - ReleaseKeys メソッド
 - MobileDevice クラス。
 - iOS 上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドはサポートされません。
 - Android 6.0 より前のバージョンの Android 上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドを使用する場合は、**擬似ロケーションを許可** を有効にする必要があります。設定は、Android デバイスまたはエミュレータの設定を開き、**開発者向けオプション** をタップします。
 - Android 6.0 以降上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドを使用する場合は、**Appium Settings** をアプリとして設定する必要があります。設定は、Android デバイスまたはエミュレータの設定を開き、**開発者向けオプション** > **仮の現在地情報アプリを選択** をタップします。そして、**Appium Settings** を選択します。



注: Appium Settings という項目は、Android デバイスまたはエミュレータ上の Appium でテストを既に実行した場合にのみ表示されます。

- iOS 上でのテスト時に、XCUIElementTypeSwitch クラスの getValue メソッドがチェック状態に応じて、文字列 0、1 ではなく、文字列 false、true を返します。
- 横固定モードでの記録はシステム バーに仮想ボタンを持つ Android エミュレータではサポートされません。このようなエミュレータは、回転を正しく検出せずに、横固定モードのシステム バーを画面の下部ではなく画面の右側に配置します。ただし、このようなエミュレータは縦固定モードで記録することができます。
- モバイル アプリケーションに対する XPath 式では、HTML DOM の HTML 属性だけがサポートされます。Silk Test Workbench は、XPath 式のプロパティをサポートしません。
- 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。
- Silk Test Workbench は、Android と iOS の両方とも、ネイティブ モバイル アプリケーションのテスト時にテキスト解決をサポートしません。

テキスト解決は次のメソッドを含みます。

- TextCapture
- TextClick
- TextExists
- TextRectangle
- Silk Test Workbench は、複数の Web ビューを持つネイティブ モバイル アプリケーションのテストをサポートしません。
- iOS 上でのテスト時に、isVisible プロパティの状態は、要素が非表示であったとしても常に true になります。
- iOS 上でのテスト時に、複数のステップを持つスワイプ操作は、あるポイントへスワイプし、マウス ポインタをリリースした後、次のポイントへスワイプします。iOS の以前のバージョンでは、この操作はスワイプ間でマウス ポインタをリリースしません。
- iOS 上でのテスト時に、Silk Test Workbench はピンチ以外のマルチタッチ操作をサポートしません。
- iOS 上でのテスト時に、Silk Test Workbench は PinchIn メソッドをサポートしません。
- iOS 上でのテスト時に、警告ダイアログ ボックスに対して承認と解除のみを行えます。**キャンセル** ボタンは利用できないため、Silk Test Workbench はダイアログを解除できません。デフォルトの操作はダイアログの承認です。
- Android 上でのテスト時に、Silk Test Workbench は **Animation** クラスのコントロールに対する自動同期を行いません。

- Android 上でのトーストのテスト時には、次の制限事項があります。
 - 記録時に、トースト用の矩形領域は、トーストの実際の位置とは関係なく、Silk Test Workbench の記録 ウィンドウの下部に表示されます。
 - 記録時と再生時に、トーストがすぐに表示された場合でも、Silk Test Workbench によるトーストの検出に、常に 5 秒間かかります。
- iOS 上でのテスト時に、Silk Test Workbench は `UIView.animate` 関数または `UIView.animateWithDuration` 関数を呼び出すコントロールに対する自動同期を行いません。

アプリケーション デリゲートでアニメーションの速度を速くすることで、問題が回避できる可能性があります。

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[NSObject: AnyObject]?) -> Bool {
    //...
    if NSProcessInfo.processInfo().environment["automationName"] == "Silk Test" {
        // Speed animations up (recommended)
        window!.layer.speed = 100;
    }
}
```


このようなアニメーションを完全に無効にすることは、アプリケーションの挙動が変わってしまう可能性があるため、Micro Focus では推奨していません。しかし、アニメーションの速度を速くしても同期問題が解決できない場合は、次のようにしてアプリケーション デリゲートでアニメーションを完全に無効化することもできます。

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[NSObject: AnyObject]?) -> Bool {
    //...
    if NSProcessInfo.processInfo().environment["automationName"] == "Silk Test" {
        UIView.setAnimationsEnabled(false)
    }
}
```

- iOS 上でのテスト時には、さらに次の制限事項があります。
 - テストの記録および再生時にパフォーマンスの低下を感じる場合があります。
 - iOS の内部的な変更の影響で一部のコントロールのロケータが変ったため、既存のテストが動作しない場合があります。
 - フォーカスを持たないテキスト フィールドが、テキスト フィールドとして解決されない場合があります。テキスト フィールドを正しく解決するために、操作する前にテキスト フィールドのクリックなどを実行して、テキスト フィールドにフォーカスを移してください。

ネイティブ モバイル アプリに対するメソッドの動的呼び出し

動的呼び出しを使うと、モバイル ネイティブ アプリに対して Appium WebDriver のメソッドを直接呼び出すことができます。これは、Appium WebDriver のメソッドが Silk Test Workbench API を介して公開されていない場合に有用です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの複数の動的メソッドは `InvokeMethods` メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、`GetDynamicMethodList` メソッドを使用します。

サポートされているメソッド

- Android 上のネイティブ モバイル アプリのテストでは、[Appium Java-client API](#) の `AndroidDriver` クラスで利用可能なメソッドを Silk Test Workbench はサポートします。

- iOS 上のネイティブ モバイル アプリのテストでは、[Appium Java-client API](#) の `IOSDriver` クラスで利用可能なメソッドを Silk Test Workbench はサポートします。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- プリミティブ型 (boolean、integer、long、double、string)
プリミティブ型 (int など) とオブジェクト タイプ (java.lang.Integer など) の両方がサポートされます。プリミティブ型は必要に応じて拡大変換されます。たとえば、int が必要な場所で long を渡すことができます。
- 列挙型
列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが列挙型 `ScreenOrientation` のパラメータを必要とする場合、次の文字列値を使用できます： `LANDSCAPE`、`PORTRAIT`。
- リスト
リスト、配列、または可変長引数のパラメータを持つメソッドを呼び出すことができます。リストの要素がターゲットの配列型に代入可能の場合、配列型への変換は自動的に行われます。

戻り値

戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では `null` が、VB では `Nothing` が返されます。

例

次のサンプル コードには、動的呼び出しを使用する共通の例を示します。

```
' VB .NET code
' Getting the page source
Dim pageSource As String = _desktop.MobileDevice("//
MobileDevice").Invoke("getPageSource")

' Resetting an app
_desktop.MobileDevice("//MobileDevice").Invoke("resetApp")

' Changing the device orientation
_desktop.MobileDevice("//MobileDevice").Invoke("rotate", "LANDSCAPE")
_desktop.MobileDevice("//MobileDevice").Invoke("rotate", "PORTRAIT")

' Dynamic invoke on MobileObject (calls get redirected to the underlying
web element for WebDriver)
With _desktop.MobileDevice("//MobileDevice")
    .MobileObject("CheckBox 2").Invoke("click")
End With
```

モバイル Web サイトでのオブジェクトのクリック

自動テストの記録と再生中にオブジェクトをクリックするとき、モバイル Web サイトではデスクトップ Web サイトと比較して、次のような困難があります。

- 拡大/縮小率やデバイス ピクセル比が異なる
- さまざまなモバイル デバイスによって画面サイズが異なる

- モバイル デバイス間でのフォントとグラフィックサイズが異なる (通常、デスクトップ ブラウザの Web サイトよりも小さい)。
- さまざまなモバイルデバイスによってピクセル サイズと解像度が異なる

Silk Test Workbench は、このような困難をものともせず、モバイル Web サイトの適切なオブジェクトをクリックできます。

モバイル デバイスでテストを記録するときに、Silk Test Workbench は Click の記録時に座標を記録しません。ただし、クロス ブラウザ テストの場合、再生中に座標が許されています。また、Click に座標を手動で追加することもできます。Silk Test Workbench は、これらの座標をオブジェクトの HTML 座標として解釈します。モバイル デバイスのテストの再生時に BrowserWindow の内側の適切なオブジェクトをクリックするために、Silk Test Workbench はオブジェクトの HTML 座標に現在の拡大/縮小率を適用します。デバイスのピクセル座標は、オブジェクトの HTML 座標に現在の拡大/縮小率をかけた座標です。

モバイル Web サイトの現在表示されている領域にオブジェクトが表示されていない場合、Silk Test Workbench は Web サイトの適切な位置にスクロールします。

例

HTML ページで 100 x 20 ピクセルの固定サイズの DomButton をテストするコードを以下に示します。

VB

```
DomButton domButton = _desktop.Find(Of DomButton)("locator for the button");
domButton.Click(MouseButton.LEFT, new Point(50, 10));
```

異なるモバイル デバイスまたは異なる拡大/縮小率で再生すると、たとえば DomButton は、デバイス画面上では実際は 10 ピクセルの幅かもしれませんが、現在の拡大/縮小率の影響を受けず、上記のコードを使用したときに Silk Test Workbench は要素の中央をクリックします。これは、Silk Test Workbench が座標を HTML 座標として解釈し、現在の拡大/縮小率を適用するためです。

既存のモバイル Web テストの使用方法

Silk Test 17.0 以降では、モバイル Web テストの扱いが、前のバージョンの Silk Test とは異なります。この変更により、以前のモバイル Web テストが Silk Test 17.0 以降では動作しなくなる可能性があります。このトピックでは、Silk Test 17.0 で行われた変更について説明し、既存のモバイル Web テストを Silk Test 17.0 以降で使用できるように変更する方法を説明します。

Silk Test 17.0 でモバイル Web テストに対して行われた変更は、以下の通りです。

- Silk Test の以前のバージョンでは、Windows マシンに USB で接続された iOS デバイスをテストすることができました。Silk Test 17.0 以降では、OS X マシン (Mac) に接続された iOS デバイスに対してのみテストすることができます。
- 以前のバージョンの Silk Test を使用して Android デバイス上のモバイル Web アプリケーションをテストしていた場合、Silk Test 17.0 以降で Web アプリケーションをテストするには、Android デバイスのプロキシ設定を手動で削除する必要があります。Silk Test 17.0 以降では、プロキシは使用しません。プロキシが設定されていると、「プロキシ サーバーに接続できません」というメッセージがデバイスに表示されます。

.NET のサポート

Silk Test は、以下の .NET アプリケーションのテストを組み込みでサポートしています。

- Windows Forms (Win Forms) アプリケーション

- Windows Presentation Foundation (WPF) アプリケーション
- Microsoft Silverlight アプリケーション

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

Windows Forms のサポート

Silk Test Workbench は、.NET スタンドアロン アプリケーションとノータッチ Windows Forms (Win Forms) アプリケーションのテストを組み込みでサポートしています。ただし、スタンドアロン アプリケーションでは、side-by-side 実行はサポートされていません。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

オブジェクト解決

アプリケーション中の要素に指定された name が使用可能な場合、ロケーターの automationId 属性として使用されます。この結果、多くのオブジェクトは、この属性のみを使用して一意に識別できます。

サポートするコントロール

Win Forms テストで使用可能な記録/再生コントロールの完全な一覧については、「[Windows Forms クラス リファレンス](#)」を参照してください。

Windows Forms クラス リファレンス


Windows Forms アプリケーションを設定すると、Silk Test Workbench は標準の Windows Forms コントロールのテストのサポートを組み込みで提供します。

Windows Forms アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Windows Forms アプリケーションがサポートする属性は次のとおりです。

- automationid
- caption
- windowid
- priorlabel (caption のないコントロールの場合、自動的に priorlabel が caption として使用されます。caption のあるコントロールの場合、caption を使う方が簡単な場合があります。)

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Windows Forms アプリケーションのカスタム属性


Windows Forms アプリケーションは、あらかじめ定義された自動化用プロパティ automationId を使用して、Windows Forms コントロールに対して安定した識別子を指定します。

Silk Test Workbench は、ロケーターを識別するために、自動的にこのプロパティを使用します。Windows Forms アプリケーションのロケーターは次のようになります。

```
/FormsWindow//PushButton[@automationId='btnBasicControls']
```

Windows Forms メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアルテストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある setTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

Invoke メソッド

Windows Forms または WPF コントロールでは、Invoke メソッドを使用して、以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

Invoke メソッドの最初の例

Silk Test Workbench の DataGridView 型のオブジェクトでは、MSDN が System.Windows.Forms.DataGridView 型に定義しているすべてのメソッドを呼び出すことができます。

System.Windows.Forms.DataGridView クラスのメソッド IsExpanded を呼び出すには、次のコードを使用します。

```
//VB .NET code  
Dim isExpanded As Boolean = dataGrid.Invoke("IsExpanded", 3)
```

Invoke メソッドの 2 番目の例

AUT 内の静的メソッド `String.Compare(String s1, String s2)` を呼び出すには、次のコードを使用します。

```
//VB .NET code
Dim result as Integer = (Integer)
mainWindow.Invoke("System.String.Compare", "a", "b")
```

Invoke メソッドの 3 番目の例

この例では、ユーザーが生成したメソッド `GetContents` を動的に呼び出す方法を示します。

テスト対象アプリケーション (AUT) のコントロールの操作に使用するコードを作成できます (この例では `UltraGrid`)。 `UltraGrid` の内容を取得するために、複雑な動的呼び出しを作成するのではなく、新しいメソッド `GetContents` を生成し、この新しいメソッドを動的に呼び出すことができます。

Visual Studio で、AUT 内の次のコードによって `GetContents` メソッドを `UltraGridUtil` クラスのメソッドとして定義します。

```
//C# code, because this is code in the AUT
namespace UltraGridExtensions {
    public class UltraGridUtil {
        /// <summary>
        /// Retrieves the contents of an UltraGrid as nested list
        /// </summary>
        /// <param name="grid"></param>
        /// <returns></returns>
        public static List<List<string>>
GetContents(Infragistics.Win.UltraWinGrid.UltraGrid grid) {
            var result = new List<List<string>>();
            foreach (var row in grid.Rows) {
                var rowContent = new List<string>();
                foreach (var cell in row.Cells) {
                    rowContent.Add(cell.Text);
                }
                result.Add(rowContent);
            }
            return result;
        }
    }
}
```

`UltraGridUtil` クラスのコードを AUT に追加する必要があります。これは、次のようにして行います。

- アプリケーション開発者は、クラスのコードを AUT にコンパイルできます。アセンブリがすでにロードされている必要があります。
- テストの実行時に AUT にロードされる新しいアセンブリを作成できます。アセンブリをロードするには、次のコードを使用します。

```
FormsWindow.LoadAssembly(String assemblyFileName)
```

次のようにして、フルパスで指定してアセンブリをロードします。

```
mainWindow.LoadAssembly("C:/temp/ultraGridExtensions.dll")
```

`Location` メソッドを使用してアセンブリの場所を見つけることもできます。

```
//VB.NET code
Dim assemblyLocation =
```

```
GetType(UltraGridExtensions.UltraGridUtil).Assembly.Location  
mainWindow.LoadAssembly(assemblyLocation)
```

UltraGridUtil クラスのコードが AUT 内にある場合は、次のコードをテスト スクリプトに追加して、GetContents メソッドを呼び出すことができます。

```
var contents = (IList)  
mainWindow.Invoke("UltraGridExtensions.UltraGridUtil.GetContents",  
ultraGrid);
```

Invoke メソッドを呼び出す mainWindow オブジェクトは、AUT を特定しているだけなので、同じ AUT の他のオブジェクトに置き換えてもかまいません。

InvokeMethods メソッド

Windows Forms または WPF コントロールでは、InvokeMethods メソッドを使用して、ネストされたメソッドのシーケンスを呼び出すことができます。以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

例 : カスタム データ グリッドのセルの内容のテキストでの取得

Infragistics ライブラリのカスタム データ グリッドのセルの内容をテキストで取得するには、AUT で次の C# コードを使用できます。

```
string cellText = dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
```

次の C# コードのサンプルは、最初の行の 3 番目のセルの内容をテキストで取得します。

```
string cellText = dataGrid.Rows[0].Cells[2];
```

InvokeMethods メソッドを使用して同じ例をスクリプト化すると、比較的複雑なスクリプトになります。これは、対応するパラメータを持つ 5 つのメソッドを InvokeMethods メソッドに渡さなければならないためです。

```
' VB .NET code  
Dim dataGrid = mainWindow.WPFControl("@automationId='Custom Data  
Grid'")  
  
' Get text contents of third cell in first row.  
Dim rowIndex = 0  
Dim column = 2  
  
Dim methodNames = New List(Of String)()  
methodNames.Add("Rows") ' Get the list of rows  
from the grid.  
methodNames.Add("get_Item") ' Get a specific row from  
the list of rows by using the indexer method.  
methodNames.Add("Cells") ' Get the list of cells from  
the the row.  
methodNames.Add("get_Item") ' Get a specific cell from  
the list of cells by using the indexer method.  
methodNames.Add("Text") ' Get the text of the cell.  
  
Dim parameters = New List(Of List(Of Object))()  
parameters.Add(New List(Of Object)()) ' Parameters  
for the Rows property.  
parameters.Add(New List(Of Object) From {rowIndex}) ' Parameters  
for the get_Item method.
```

```

parameters.Add(New List(Of Object)()) ' Parameters
for the Cells property.
parameters.Add(New List(Of Object) From {columnIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object)()) ' Parameters
for the Text property.

Dim cellText As String = dataGrid.InvokeMethods(methodNames,
parameters)

```

このような場合に、より簡単にするアプローチは、テスト対象アプリケーションにコードを追加して、InvokeMethods メソッドを使用することです。たとえば、GetCellText メソッドを AUT に追加します。

```

// C# code, if the AUT is implemented in C#.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
}

```

```

' VB code, if the AUT is implemented in VB.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
}

```

テスト スクリプトから GetCellText メソッドを動的に呼び出して、セルの内容をテキストで取得します。

```

'VB .NET code
Dim cellText As String = mainWindow.Invoke("GetCellText", dataGrid,
rowIndex, columnIndex)

```

詳細については、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」を参照してください。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- MSDN が定義するコントロールのパブリック メソッドとプロパティ。
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point や Rect など) が含まれます。

- 列挙型

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合、次の文字列値を使用できます：Visible、Hidden、Collapsed。

- .NET 構造体とオブジェクト

.NET 構造体とオブジェクトパラメータはリストとして渡す必要があります。リスト内の要素は、テストアプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場

合、2つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector型が2つの整数値を引数に取るコンストラクタを持つためです。

- その他のコントロール
コントロールパラメーターは、TestObjectとして渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

Windows Presentation Foundation (WPF) のサポート

Silk Test Workbench は、Windows Presentation Foundation (WPF) アプリケーションのテストを組み込みでサポートしています。Silk Test Workbench は、スタンドアロン WPF アプリケーションをサポートしており、.NET バージョン 3.5 以降に組み込まれているコントロールを記録し、再生できます。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

WPF テストで使用可能なコントロールの完全な一覧については、「[WPF クラス リファレンス](#)」を参照してください。

Silk Test Workbench WPF がサポートするすべての WPF クラスは、WPFWindow や WPFListBox のように接頭辞 *WPF* で始まります。

WPF コントロールでサポートされるメソッドとプロパティは、実際の実装とランタイム状態によって異なります。メソッドとプロパティは、対応するクラスに対して定義されたリストと異なる場合があります。特定の状況でサポートされるメソッドとプロパティを判別するには、以下のコードを使用します。

- `GetPropertyList()`
- `GetDynamicMethodList()`

WPF の詳細については、[MSDN](#) を参照してください。


WPF クラス リファレンス

WPF アプリケーションを設定すると、Silk Test Workbench は標準の WPF コントロールのテストのサポートを組み込みで提供します。

Windows Presentation Foundation (WPF) アプリケーションの属性

WPF アプリケーションがサポートする属性は次のとおりです。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別され

ますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード？および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

オブジェクト解決

WPF スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、あるいは *name* を指定できます。アプリケーション中の要素に指定された *name* が利用可能な場合、ロケータの *automationId* 属性として使用されます。この結果、多くのオブジェクトは、この属性のみを使用して一意に識別できます。たとえば、*automationId* を持つロケータは、以下のようになります：

```
// WPFButton[@automationId='okButton']"
```

automationId や他の属性を定義した場合、再生中に *automationId* だけが使用されます。*automationId* が定義されていない場合には、コンポーネントを解決するのに *name* が使用されます。*name* も *automationId* もどちらも定義されていない場合には、*caption* 値が使用されます。*caption* が定義されていない場合は、*className* が使用されます。*automationId* は非常に役立つプロパティであるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト アプリケーションの開発者によって提供された ID	<pre>//WPFButton[@automationId='okButton']"</pre>
<i>name</i>	コントロールの名前。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に名前を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この名前を使用します。	<pre>//WPFButton[@name='okButton']"</pre>
<i>caption</i>	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	<pre>//WPFButton[@automationId='Ok']"</pre>
<i>className</i>	WPF の .NET 単純クラス名 (名前空間なし)。クラス名属性を使用すると、Silk Test Workbench が解決する標準 WPF コントロールから派生したカスタム コントロールを識別するのに役立ちます。	<pre>//WPFButton[@className='MyCustomButton']"</pre>

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して WPF コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、Silk Test Workbench がロケータを作成する際には *automationId* が使用されません。

以下の例では、アプリケーション開発者がアプリケーションの WPF ボタンに対して *name* と *automationId* を XAML コードに定義する方法を示します。

```
<Button Name="okButton" AutomationProperties.AutomationId="okButton"
Click="okButton_Click">Ok</Button>
```

WPF アプリケーションのカスタム属性

WPF アプリケーションは、あらかじめ定義された自動化用プロパティ `AutomationProperties.AutomationId` を使用して、次のように WPF コントロールに対して安定した識別子を指定します。

```
<Window x:Class="Test.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="MainWindow" Height="350" Width="525">
  <Grid>
    <Button AutomationProperties.AutomationId="AID_buttonA">The
Button</Button>
  </Grid>
</Window>
```

Silk Test Workbench は、ロケータを識別するために、自動的にこのプロパティを使用します。WPF アプリケーションのロケータは次のようになります。

```
/WPFWindow[@caption='MainWindow']/WPFButton[@automationId='AID_buttonA']
```

WPFIItemsControl クラスから派生したクラス

Silk Test Workbench は、2 つの方法を使用して `WPFIItemsControl` から派生したクラス (`WPFListBox`、`WPFTreeView`、`WPFMenu` など) を操作することができます。

- コントロールでの作業

ほとんどのコントロールには、標準的なユースケースのためのメソッドやプロパティがあります。項目は、テキストや索引によって識別されます。

例：

```
listBox.Select("Banana")
listBox.Select(2)
```

```
tree.Expand("/Fruit/Banana")
```

- `WPFListBoxItem`、`WPFTreeViewItem`、`WPFMenuItem` などの個々の項目での作業

高度なユースケースの場合、個々の項目を使用します。たとえば、リストボックスの特定の項目のコンテキストメニューを開いたり、項目に相対的な場所をクリックしたりする場合に個々の項目を使用します。

例：

```
Dim item as WPFListBoxItem = listBox.WPFListBoxItem("banana")
item.OpenContextMenu()
```

カスタム WPF コントロール

一般的に、Silk Test Workbench では、すべての標準 WPF コントロールの記録と再生がサポートされています。

Silk Test Workbench は、カスタム コントロールが実装された方法を基にしてカスタム コントロールを処理します。次の方法を使用してカスタム コントロールを実装することができます。

- `UserControl` から派生したクラスを定義する


複合コントロールを作成する典型的な方法です。Silk Test Workbench は、これらのユーザー コントロールを `WPFUserControl` として認識し、含まれるコントロールを完全にサポートしています。

- ListBox などの標準 WPF コントロールから派生したクラスを定義する
Silk Test Workbench は、これらのコントロールを派生元の標準 WPF コントロールのインスタンスとして扱います。ユーザー コントロールの振る舞いがその基底クラスの実装と大きく異なる場合には、子の記録、再生、解決は機能しない可能性があります。
- テンプレートを使用して視覚デザインを変更した標準コントロールを使用する
低レベルの再生が機能しない可能性があります。その場合には、「高レベル」再生モードに切り替えます。再生モードを変更するには、**オプション** ダイアログ ボックスを使用して、**再生全般** 設定リストで **再生モード** を変更します。

Silk Test Workbench は、一般的に機能テストに無関係なコントロールは除外します。たとえば、レイアウトを目的として使用されるコントロールは含まれません。しかし、カスタム コントロールが除外されたクラスから派生している場合、除外されたコントロールを記録/再生の対象とするためには、関連する WPF クラスの名前を指定します。

WPF メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

Invoke メソッド

Windows Forms または WPF コントロールでは、Invoke メソッドを使用して、以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

Invoke メソッドの最初の例

Silk Test Workbench の DataGrid 型のオブジェクトでは、MSDN が System.Windows.Forms.DataGrid 型に定義しているすべてのメソッドを呼び出すことができます。

System.Windows.Forms.DataGrid クラスのメソッド IsExpanded を呼び出すには、次のコードを使用します。

```
//VB .NET code
Dim isExpanded As Boolean = dataGrid.Invoke("IsExpanded", 3)
```

Invoke メソッドの 2 番目の例

AUT 内の静的メソッド String.Compare(String s1, String s2) を呼び出すには、次のコードを使用します。

```
//VB .NET code
Dim result as Integer = (Integer)
mainWindow.Invoke("System.String.Compare", "a", "b")
```

Invoke メソッドの 3 番目の例

この例では、ユーザーが生成したメソッド GetContents を動的に呼び出す方法を示します。

テスト対象アプリケーション (AUT) のコントロールの操作に使用するコードを作成できます (この例では UltraGrid)。UltraGrid の内容を取得するために、複雑な動的呼び出しを作成するのではなく、新しいメソッド GetContents を生成し、この新しいメソッドを動的に呼び出すことができます。

Visual Studio で、AUT 内の次のコードによって GetContents メソッドを UltraGridUtil クラスのメソッドとして定義します。

```
//C# code, because this is code in the AUT
namespace UltraGridExtensions {
    public class UltraGridUtil {
        /// <summary>
        /// Retrieves the contents of an UltraGrid as nested list
        /// </summary>
        /// <param name="grid"></param>
        /// <returns></returns>
        public static List<List<string>>
GetContents(Infragistics.Win.UltraWinGrid.UltraGrid grid) {
            var result = new List<List<string>>();
            foreach (var row in grid.Rows) {
                var rowContent = new List<string>();
                foreach (var cell in row.Cells) {
                    rowContent.Add(cell.Text);
                }
                result.Add(rowContent);
            }
            return result;
        }
    }
}
```

UltraGridUtil クラスのコードを AUT に追加する必要があります。これは、次のようにして行います。

- アプリケーション開発者は、クラスのコードを AUT にコンパイルできます。アセンブリがすでにロードされている必要があります。
- テストの実行時に AUT にロードされる新しいアセンブリを作成できます。

アセンブリをロードするには、次のコードを使用します。

```
FormsWindow.LoadAssembly(String assemblyFileName)
```

次のようにして、フルパスで指定してアセンブリをロードします。

```
mainWindow.LoadAssembly("C:/temp/ultraGridExtensions.dll")
```

Location メソッドを使用してアセンブリの場所を見つけることもできます。

```
//VB.NET code
Dim assemblyLocation =
GetType(UltraGridExtensions.UltraGridUtil).Assembly.Location
mainWindow.LoadAssembly(assemblyLocation)
```

UltraGridUtil クラスのコードが AUT 内にある場合は、次のコードをテスト スクリプトに追加して、GetContents メソッドを呼び出すことができます。

```
var contents = (IList)
mainWindow.Invoke("UltraGridExtensions.UltraGridUtil.GetContents",
ultraGrid);
```

Invoke メソッドを呼び出す mainWindow オブジェクトは、AUT を特定しているだけなので、同じ AUT の他のオブジェクトに置き換えてもかまいません。

InvokeMethods メソッド

Windows Forms または WPF コントロールでは、InvokeMethods メソッドを使用して、ネストされたメソッドのシーケンスを呼び出すことができます。以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

例 : カスタム データ グリッドのセルの内容のテキストでの取得

Infragistics ライブラリのカスタム データ グリッドのセルの内容をテキストで取得するには、AUT で次の C# コードを使用できます。

```
string cellText = dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
```

次の C# コードのサンプルは、最初の行の 3 番目のセルの内容をテキストで取得します。

```
string cellText = dataGrid.Rows[0].Cells[2];
```

InvokeMethods メソッドを使用して同じ例をスクリプト化すると、比較的複雑なスクリプトになります。これは、対応するパラメータを持つ 5 つのメソッドを InvokeMethods メソッドに渡さなければならないためです。

```
' VB .NET code
Dim dataGrid = mainWindow.WPFControl("@automationId='Custom Data Grid'")

' Get text contents of third cell in first row.
Dim rowIndex = 0
Dim column = 2

Dim methodNames = New List(Of String)()
methodNames.Add("Rows") ' Get the list of rows
from the grid.
```

```

methodNames.Add("get_Item")           ' Get a specific row from
the list of rows by using the indexer method.
methodNames.Add("Cells")              ' Get the list of cells from
the the row.
methodNames.Add("get_Item")           ' Get a specific cell from
the list of cells by using the indexer method.
methodNames.Add("Text")               ' Get the text of the cell.

```

```

Dim parameters = New List(Of List(Of Object))()
parameters.Add(New List(Of Object))() ' Parameters
for the Rows property.
parameters.Add(New List(Of Object) From {rowIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object))() ' Parameters
for the Cells property.
parameters.Add(New List(Of Object) From {columnIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object))() ' Parameters
for the Text property.

```

```

Dim cellText As String = dataGrid.InvokeMethods(methodNames,
parameters)

```

このような場合に、より簡単にするアプローチは、テスト対象アプリケーションにコードを追加して、InvokeMethods メソッドを使用することです。たとえば、GetCellText メソッドを AUT に追加します。

```

// C# code, if the AUT is implemented in C#.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
}

```

```

' VB code, if the AUT is implemented in VB.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
}

```

テスト スクリプトから GetCellText メソッドを動的に呼び出して、セルの内容をテキストで取得します。

```

'VB .NET code
Dim cellText As String = mainWindow.Invoke("GetCellText", dataGrid,
rowIndex, columnIndex)

```

詳細については、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」を参照してください。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- MSDN が定義するコントロールのパブリック メソッドとプロパティ。
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point や Rect など) が含まれます。

- 列挙型

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合、次の文字列値を使用できます：Visible、Hidden、Collapsed。

- .NET 構造体とオブジェクト

.NET 構造体とオブジェクトパラメータはリストとして渡す必要があります。リスト内の要素は、テストアプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。

- WPF コントロール

WPF コントロールパラメータは TestObject として渡すことができます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。
- すべてのその他の型の場合は文字列

返された .NET オブジェクトに対して ToString を呼び出せば、文字列表現を取得できます。

例

たとえば、アプリケーション開発者が次のメソッドとプロパティを持つ Calculator カスタム コントロールを作成したとします。

```
public void Reset()
public int Add(int number1, int number2)
public System.Windows.Vector StrechVector(System.Windows.Vector vector,
double
factor)
public String Description { get;}
```

テスト担当者は、テスト内からメソッドを直接呼び出すことができます。例：

```
customControl.Invoke("Reset")
Dim sum as Integer = customControl.Invoke("Add", 1, 2)
' the vector can be passed as list of integer
Dim vector = New List(Of Integer)
vector.Add(3)
vector.Add(4)
' returns "6;8" because this is the string representation of the .NET
object
Dim stretchedVector As String = customControl.Invoke("StrechVector",
vector, 2.0)
Dim description As String = customControl.GetProperty("Description")
```

記録/再生の対象とする WPF クラスの設定

Silk Test Workbench は、一般的に機能テストに無関係なコントロールは除外します。たとえば、レイアウトを目的として使用されるコントロールは含まれません。しかし、カスタム コントロールが除外された

クラスから派生している場合、除外されたコントロールを記録/再生の対象とするためには、関連する WPF クラスの名前を指定します。

記録や再生の対象にしたい WPF クラスの名前を指定します。たとえば、*MyGrid* というカスタム クラスが WPF Grid クラスから継承された場合、*MyGrid* カスタム クラスのオブジェクトは記録や再生に使用できません。Grid クラスはレイアウト目的のためにのみ存在し、機能テストとは無関係であるため、Grid オブジェクトは記録や再生に使用できません。この結果、Grid オブジェクトはデフォルトでは公開されません。機能テストに無関係なクラスに基づいたカスタム クラスを使用するには、カスタム クラス (この場合は *MyGrid*) を **OPT_WPF_CUSTOM_CLASSES** オプションに追加します。これによって、記録、再生、検索、プロパティの検証など、すべてのサポートされる操作を指定したクラスに対して実行できるようになります。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。
3. **WPF** をクリックします。
4. **カスタム WPF クラス名** グリッドで、記録や再生中に公開するクラスの名前を入力します。
複数のクラス名を指定する場合にはカンマで区切ります。
5. **OK** をクリックします。

記録/再生時の事前読み込みの設定

記録および再生中に、WPFComboBox や WPFListBox のような WPFItemsControl 内の項目を事前に入力するかどうかを定義します。WPF 自体が特定のコントロールの項目を遅延読み込みするため、項目がビューにスクロールされない場合、それらの項目は Silk Test Workbench では使用できません。ビューにスクロールされないとアクセスできない項目にアクセスするには、事前入力をオンにします。これはデフォルトの設定です。ただし、一部のアプリケーションでは Silk Test Workbench によってバックグラウンドで項目が事前入力されると問題が発生し、そのためアプリケーションがクラッシュすることがあります。この場合、事前入力をオフにします。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。
3. **WPF** をクリックします。
4. **項目の事前読み込み** を **はい** に設定します。
5. **OK** をクリックします。

Silverlight アプリケーションのサポート

Microsoft Silverlight (Silverlight) は、リッチ インターネット アプリケーションを記述し、実行するためのアプリケーション フレームワークで、Adobe Flash と同様の機能と目的を備えています。Silverlight の実行時環境は、大部分の Web ブラウザでプラグインとして使用できます。

Silk Test Workbench は、Silverlight アプリケーションのテストを組み込みでサポートしています。Silk Test Workbench は、ブラウザ内部と同様ブラウザ外部でも実行される Silverlight アプリケーションをサポートしており、.NET バージョン 3.5 以降でコントロールを記録し、再生できます。

Silverlight をベースとする以下のアプリケーションがサポートされます。


- Internet Explorer で実行される Silverlight アプリケーション
- ブラウザー外実行 Silverlight アプリケーション

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール


Silk Test Workbench は、Silverlight コントロールの記録と再生をサポートしています。

Silverlight テストで使用可能なコントロールの完全な一覧については、「[Silverlight クラス リファレンス](#)」を参照してください。

 **注:** Silk Test 14.0 以降では、Silk Test Workbench は、画面上で操作可能でかつ表示されている Silverlight コントロールのみを認識します。この変更は、Silk Test 14.0 より前のバージョンの Silk Test を使用して記録されたテストの動作に影響を与える可能性があります。Silk Test 14.0 以降を使用してこのようなテストを実行するには、不可視な、または利用可能でないすべての Silverlight コントロールをテストから削除してください。

前提条件

Microsoft Windows XP で Silverlight アプリケーションをテストする場合、サービスパック 3 をインストールし、Windows 7 で提供される Microsoft User Interface Automation の Windows XP 用の更新プログラムを適用する必要があります。更新プログラムは、<http://www.microsoft.com/download/en/details.aspx?id=13821> からダウンロードできます。

 **注:** Microsoft User Interface Automation は、Silverlight サポート用にインストールする必要があります。Windows オペレーティングシステムを使用していて、Silverlight サポートが機能しない場合は、使用しているオペレーティングシステム用の Microsoft User Interface Automation の更新プログラムを <http://support.microsoft.com/kb/971513> からダウンロードしてインストールしてください。


Silverlight クラス リファレンス

Silverlight アプリケーションを設定すると、Silk Test Workbench は標準の Silverlight コントロールのテストのサポートを組み込みで提供します。

Silverlight コントロールを識別するためのロケータ属性

Silverlight コントロールでサポートされているロケータ属性は次のとおりです。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。


動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

Silverlight スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、*name*、または任意の動的ロケータ属性を指定できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下ようになります：
`// SLButton[@automationId="okButton"]`

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	<code>// SLButton[@automationId="okButton"]</code>

属性の種類	説明	例
caption	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//SLButton[@caption="Ok"]
className	Silverlight コントロールの .NET 単純クラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 Silverlight コントロールから派生したカスタム コントロールを識別するのに役立ちます。	//SLButton[@className='MyCustomButton']
name	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	//SLButton[@name="okButton"]

 **注目:** XAML コードの *name* 属性は、ローケータ属性 *name* ではなく、ローケータ属性 *automationId* にマップされます。


Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して Silverlight コントロールのローケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* が固有の場合は Silk Test Workbench がローケータを作成する際に使用されます。

以下の表は、アプリケーション開発者がテキスト「Ok」を持つ Silverlight ボタンをアプリケーションの XAML コードに定義する方法を示しています。

オブジェクトの XAML コード	Silk Test からオブジェクトを検索するためのローケータ
<Button>Ok</Button>	//SLButton[@caption="Ok"]
<Button Name="okButton">Ok</Button>	//SLButton[@automationId="okButton"]
<Button AutomationProperties.AutomationId="okButton">Ok</Button>	//SLButton[@automationId="okButton"]
<Button AutomationProperties.Name="okButton">Ok</Button>	//SLButton[@name="okButton"]

Silverlight メソッドの動的呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアルテストでは使用できません。

オブジェクトの動的メソッドは *Invoke* メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、*GetDynamicMethodList* メソッドを使用します。

オブジェクトの複数の動的メソッドは *InvokeMethods* メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、*GetDynamicMethodList* メソッドを使用します。

動的プロパティの取得には *GetProperty* メソッドを、動的プロパティの設定には *SetProperty* メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、*GetPropertyList* メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```



注: 通常、ほとんどのプロパティは読み取り専用で、設定できません。



注: ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型。

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point、Rect など) が含まれます。

- 列挙型。

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合には、Visible、Hidden、Collapsed の文字列値を使用できます。

- .NET 構造体とオブジェクト。

.NET 構造体とオブジェクトパラメータはリストとして渡します。リスト内の要素は、テスト アプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。

- その他のコントロール。

コントロールパラメータは TestObject として渡すことができます。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- MSDN が定義する AutomationElement クラスのすべてのパブリック メソッドとプロパティ。詳細については、<http://msdn.microsoft.com/en-us/library/system.windows.automation.automationelement.aspx> を参照してください。
- MSUIA が公開するすべてのメソッドとプロパティ。利用可能なメソッドとプロパティは「パターン」で分類されます。パターンとは、MSUIA 固有の用語です。すべてのコントロールは、いくつかのパターンを実装します。一般的なパターンについての概要およびすべての利用可能なパターンについては、<http://msdn.microsoft.com/en-us/library/ms752362.aspx> を参照してください。カスタムコントロールの開発者は、MSUIA パターン セットを実装することによって、カスタムコントロールのテストサポートを提供できます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。
- すべてのその他の型の場合は文字列。

この文字列表現を取得するには、テスト対象アプリケーションの返された .NET オブジェクトに対して ToString メソッドを呼び出します。

例

Silverlight の TabItem。これは TabControl の項目です。

```
tabItem.Invoke("SelectedItemPattern.Select")
mySilverlightObject.GetProperty("IsPassword")
```

Silverlight でのスクロール

Silk Test Workbench では、Silverlight コントロールに応じて、2 種類のスクロール方法とプロパティを提供します。

- 1 つめの種類のコントロールには、それ自体でスクロール可能なコントロールが含まれ、スクロールバーは子として明示的に表示されません。たとえば、コンボ ボックス、ペイン、リスト ボックス、ツリー コントロール、データグリッド、オート コンプリート ボックスなどがあります。
- 2 つめの種類のコントロールには、それ自体ではスクロール不可能なコントロールが含まれ、スクロール用にスクロールバーが子として表示されます。たとえば、テキスト フィールドがあります。

Silk Test Workbench にこのような違いがあるのは、Silk Test Workbench のコントロールがこの 2 通りの方法でスクロールを実装するためです。

スクロールをサポートするコントロール

この場合、スクロール方法とプロパティは、スクロールバーを含むコントロールで使用できます。したがって、Silk Test Workbench ではスクロールバー オブジェクトは表示されません。

使用例

以下のコマンドでは、リスト ボックスが一番下までスクロールされます。

```
listBox.SetVerticalScrollPercent(100)
```

以下のコマンドでは、リスト ボックスが 1 ユニットずつ下方へスクロールされます。

```
listBox.ScrollVertical(ScrollAmount.SmallIncrement)
```

スクロールをサポートしないコントロール

この場合、スクロールバーが表示されます。コントロール自体で可能なスクロール方法とプロパティはありません。水平スクロールバーと垂直スクロールバーの各オブジェクトを使用すると、対応する API 関数でパラメータとして増分または減分、または最終位置を指定することでコントロール内をスクロールできます。増分または減分として ScrollAmount 列挙の値を使用できます。詳細については、Silverlight の製品マニュアルを参照してください。最終位置は、オブジェクトの位置に関連し、アプリケーション設計者によって定義されます。

使用例

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが 15 の位置までスクロールされます。

```
textBox.SLVerticalScrollBar().ScrollToPosition(15)
```

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが一番下までスクロールされます。

```
textBox.SLVerticalScrollBar().ScrollToMaximum()
```

Silverlight アプリケーションのテスト時のトラブルシューティング

Silk Test Workbench で Silverlight アプリケーションの内部を確認できず、記録時に緑色の矩形領域が描画されない


次の理由により、Silk Test Workbench は Silverlight アプリケーションの内部を確認できなくなっています。

原因	解決策
使用している Silverlight のバージョンが 3 以前である	Silverlight 3 (Silverlight Runtime 4) または Silverlight 4 (Silverlight Runtime 4) を使用します。
使用している Silverlight アプリケーションがウィンドウレスモードで実行されている	Silk Test Workbench は、ウィンドウレスモードで実行される Silverlight アプリケーションをサポートしません。このようなアプリケーションをテストするには、Silverlight アプリケーションが実行されている Web サイトを変更する必要があります。つまり、Silverlight アプリケーションがホストされている HTML または ASPX ファイルのオブジェクト タグの windowless パラメータを false に設定する必要があります。 以下のコードは、windowless パラメータを false に設定する例を示します。 <pre><object ...> <param name="windowless" value="false"/> ... </object></pre>

Visual COBOL のサポート

Silk Test Workbench は、Visual COBOL アプリケーションに対するテストの記録と再生をサポートします。.NET COBOL から Silk Test Workbench API を使用して、次の Visual COBOL アプリケーションに対して自動テストのスクリプトを記述することもできます。

- ダイアログ システム アプリケーション
- CGI アプリケーション
- COBOL Win32 アプリケーション
- .NET COBOL - WPF および Windows Forms アプリケーション
- COBOL JVM - Swing アプリケーション
- バックエンドにある COBOL を呼び出す非 COBOL のフロントエンド アプリケーション

 **注:** コントロールによっては、Silk Test Workbench は低レベルの記録のみをサポートします。

サポートする Visual COBOL のバージョンについての情報は、『[リリース ノート](#)』を参照してください。

Rumba のサポート

Rumba は、世界トップクラスの Windows デスクトップ端末エミュレーション ソリューションです。Silk Test は、Rumba の記録および再生を組み込みでサポートしています。

Rumba でのテスト時には、以下の点を考慮してください。

- Rumba のバージョンは、Silk Test のバージョンと互換性がある必要があります。バージョン 8.1 以前の Rumba はサポートされていません。

- Rumba のグリーン スクリーンの周囲にあるコントロールはすべて WPF の基本機能 (または Win32) を使用しています。
- サポートされている Rumba デスクトップ タイプは、以下のとおりです。
 - メインフレーム ディスプレイ
 - AS400 ディスプレイ
 - Unix ディスプレイ

Rumba テストで使用できる記録および再生のコントロールの完全な一覧については、「*Rumba クラス リファレンス*」を参照してください。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『*リリース ノート*』を参照してください。

Rumba クラス リファレンス

Rumba アプリケーションを設定すると、Silk Test Workbench は標準の Rumba コントロールのテストのサポートを組み込みで提供します。

Rumba の有効化と無効化

Rumba は、世界トップクラスの Windows デスクトップ端末エミュレーション ソリューションです。Rumba は、メインフレーム、ミッドレンジ、UNIX、Linux、および HP サーバーとの接続ソリューションを提供します。

サポートの有効化

Rumba スクリプトを記録および再生する前に、サポートを有効にする必要があります。

1. Rumba デスクトップ クライアント ソフトウェアバージョン 8.1 以降をインストールします。
2. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > Rumba プラグイン > Silk Test Rumba プラグインの有効化**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Rumba プラグインの有効化** をクリックします。

サポートの無効化

(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > Rumba プラグイン > Silk Test Rumba プラグインの無効化**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Rumba プラグインの無効化** をクリックします。

Rumba コントロールを識別するためのロケータ属性


ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジ ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。サポートされている属性は次のとおりです。

caption	コントロールが表示するテキスト。
priorlabel	フォームの入力フィールドには通常入力の目的を説明するラベルがあるため、 priorlabel の目的は隣接するラベル フィールド RumbaLabel のテキストによってテキスト入力フィールド RumbaTextField を識別することです。テキスト フィールドの同じ行の直前にラベルがない場合、または右側のラベルが左側のラベルよりテキスト フィールドに近い場合、テキスト フィールドの右側にあるラベルが使用されます。

StartRow この属性は記録されていませんが、手動でロケータに追加することができます。**StartRow** を使用して、この行で始まるテキスト入力フィールド、**RumbaTextField** を識別します。

StartColumn この属性は記録されていませんが、手動でロケータに追加することができます。**StartColumn** を使用して、この列で始まるテキスト入力フィールド、**RumbaTextField** を識別します。

すべての動的ロケータ属性。 動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Rumba での画面検証の使用

Rumba に対する画面検証を自動的に挿入するには、**オプション** ダイアログ ボックスで **記録 > 全般 > 画面検証を記録する** をオンにします。

画面検証を手動で挿入するには、以下を実行します。


1. テストで、**検証タイプのロジックの作成** ボタンをクリックし、**テスト ロジック デザイナ - 検証** を開きます。
2. **次へ** をクリックします。
3. **画面のコンテンツ** を選択します。
ツール > オプション > 記録 > Rumba > 除外オブジェクト で特定されるすべての除外オブジェクトが使用されます。この手順を完了後に、テストの **プロパティ** ウィンドウでこれらをさらにカスタマイズできます。
4. **次へ** をクリックします。
5. **識別** ボタンをクリックします。
6. 識別する Rumba 画面でコントロールを選択します。画面全体がキャプチャされます。
7. **次へ** をクリックします。
8. **完了** をクリックします。

Unix ディスプレイのテスト

Rumba の Unix ディスプレイは完全にテキスト ベースで、メインの **RUMBA 画面** コントロール以外に UI コントロールはありません。Unix ディスプレイ上のテストを再生するには、SendKeys メソッドを使用して、Unix ディスプレイにキーを送信します。Silk Test Workbench は、Unix ディスプレイ上での記録をサポートしません。

SAP のサポート

Silk Test Workbench は、Windows ベースの GUI モジュールを基にした SAP クライアント/サーバー アプリケーションのテストを組み込みでサポートしています。

 **注:** Silk Test Workbench のプレミアム ライセンスを所有している場合にのみ、Silk Test Workbench で SAP アプリケーションをテストできます。ライセンス モードについての詳細は、「ライセンス情報」を参照してください。

 **注:** Internet Explorer や Firefox 上から SAP NetWeaver を使用する場合、Silk Test Workbench は、xBrowser テクノロジ ドメインを使用してアプリケーションをテストします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。


サポートするコントロール

SAP のテストで利用可能な記録および再生コントロールの完全な一覧については、「[SAP クラス リファレンス](#)」を参照してください。


サポートされている属性の一覧については、「[SAP アプリケーションの属性](#)」を参照してください。

SAP クラス リファレンス

SAP アプリケーションを設定すると、Silk Test Workbench は標準の SAP コントロールのテストのサポートを組み込みで提供します。

 **注:** Silk Test Workbench が SAP コントロールを解決できるようにするには、サーバー上で SAP GUI Scripting を有効にします。SAP GUI Scripting の有効化についての情報は、『[Introduction to SAP Gui Scripting](#)』を参照してください。SAP GUI Scripting が有効になっていないと、Silk Test Workbench は、すべてのコントロールを Win32 コントロール、およびカスタム コントロールとして解決します。

SAP クラス リファレンスに含まれるクラス (インクルードしたプロパティとメソッドを含む)は、Silk Test Workbench から直接アクセス可能な SAP オートメーション モジュールの一部です。


 **注:** インターフェイスに備わるアルゴリズムや動作を含み、インターフェイスは Silk Test Workbench のコントロール下にはありません。単純なデータ型が含まれているかぎり、ビジュアルテストはリストした SAP クラスのプロパティとメソッドを利用します。たとえば、SAPTable クラスの GetCell メソッドをビジュアルテストから呼び出すことはできませんが、VB .NET スクリプトからそのメソッドを呼び出すことができます。

SAP アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジードメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。


SAP がサポートする属性は次のとおりです。

- automationId
- caption

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

SAP メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアルテストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注：** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注：** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- SAP オートメーション インターフェイスによって定義されているすべての public メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型
Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point や Rect など) が含まれます。
- UI コントロール
UI コントロールは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

例

以下の VB .NET スクリプトの例では、SAP メソッドを動的に呼び出す方法を示します。

```
Dim _desktop As Desktop = Agent.Desktop
Dim wnd As SapWindow = _desktop.SapWindow("wnd 0")
Dim result As Object

result = wnd.Invoke("IsVKeyAllowed", 8) ' boolean return value
Console.WriteLine(
    "invoke result=" & result & " type=" & result.GetType().ToString())
wnd.Invoke("ShowMessageBox", "A Title", "Some text...", 1, 1)
```

SAP コントロールの動的呼び出し

Silk Test Workbench で SAP コントロールに対する操作を記録できない場合、SAP で利用できるレコーダーで操作を記録してから、記録されたメソッドを Silk Test Workbench スクリプトで動的に呼び出すことができます。これによって、記録できない SAP コントロールに対する操作を再生できます。

1. コントロールに対して実行する操作を記録するには、SAP で利用できる **SAP GUI スクリプト作成** ツールを使用できます。
SAP GUI スクリプト作成 ツールの詳細については、SAP のドキュメントを参照してください。
2. 記録された操作を **SAP GUI スクリプト作成** ツールによって保存された場所から開き、記録されたメソッドを確認します。
3. Silk Test Workbench で、記録されたメソッドをスクリプトから動的に呼び出します。

使用例

たとえば、SAP UI で *Test* というラベルが付いた、ボタンとリスト ボックスの組み合わせである特別なコントロールを押し、コントロールのサブメニュー *subsub2* を選択する操作を再生したい場合、SAP で利用できるレコーダーでこの操作を記録できます。結果のコードは、以下のようになります。

```
session.findById("wnd[0]/usr/cntlCONTAINER/shellcont/shell").pressContextButton "TEST"
session.findById("wnd[0]/usr/cntlCONTAINER/shellcont/shell").selectContextMenuItem "subsub2"
```

ここで、メソッド `pressContextButton` と `selectContextMenuItem` を Silk Test Workbench のスクリプトから動的に呼び出すには、次のコードを使用できます。

```
.SapToolBarControl("shell ToolbarControl").Invoke("pressContextButton", "TEST")
.SapToolBarControl("shell ToolbarControl").Invoke("selectContextMenuItem", "subsub2")
```

このコードを再生すると、SAP UI のコントロールが押下され、サブメニューが選択されます。

SAP の自動セキュリティ設定の構成

SAP アプリケーションを起動する前に、セキュリティ警告設定を構成する必要があります。このようにしないと、テストで SAP アプリケーションが再生されるたびにセキュリティ警告「スクリプトから GUI に接続しようとしています」が表示されます。

1. Windows の **コントロール パネル** で **SAP システム設定** を選択します。 **SAP システム設定** ダイアログ ボックスが開きます。
2. **デザイン選択** タブで、**スクリプトが実行中 SAP GUI に追加されるとき通知** をオフにします。

SAP eCATT の操作

Silk Test Workbench は、SAP の *eCATT (extended Computer Aided Test Tool)* と統合することにより、HTML テクノロジーに基づく製品について完全なテスト サポートを SAP ユーザーに提供します。これにより、eCATT テスト スクリプト開発者は、HTML 製品のあらゆるテストを一元管理できます。Silk Test Workbench では、ANSI または Unicode データベースのデータの操作と保存ができます。

Silk Test Workbench を eCATT のユーザー インターフェイス内で使用すると、Silk Test Workbench の資産は eCATT テストと共に取得され、SAP/R3 データベースに格納されます。テスト担当者は、自分の結果の概要を一元的に把握できます。eCATT と Silk Test Workbench テストの結果は結合され、eCATT で同じ画面に表示できます。

開発者は、eCATT と Silk Test Workbench との間でパラメータを受け渡すことによって、真の意味で統合されたテストシナリオを作成できます。また、テスト担当者は、SAP テストデータ コンテナのデータを Silk Test Workbench で使用できるようになります。最も重要な利点は、同じ SAP システムにアクセスできるテスト担当者がそれぞれのテストを共有できるということです。

Silk Test Workbench では、スクリプト引数を使用して、eCATT と Silk Test Workbench との間でパラメータを渡します。

Silk Test Workbench と eCATT 間のインターフェイスには、COM と RFC という 2 種類の通信が使用されます。eCATT における Silk Test Workbench の GUI インターフェイスは COM 経由で行われます。RFC 通信は、Silk Test Workbench が SAP/R3 データベースとの間でデータの保存と読み込みを行うために使用されます。データには、1 つまたは複数の Silk Test Workbench プロジェクトと、SAP/R3 データベースにラージ オブジェクトとして格納される、関連する Silk Test Workbench 資産のすべてが含まれます。これらのラージ オブジェクトのことを *BLOB* と呼びます。

Silk Test Workbench と SAP eCATT の統合


Silk Test Workbench と SAP eCATT の統合によって、テストプロセスを最初から最後まで管理できるようになります。一度 Silk Test Workbench と SAP eCATT を統合すれば、Silk Test Workbench と eCATT を次の 2 つの方法で使用することができます。

- eCATT 内から Silk Test Workbench を起動して、テスト スクリプトの記録、再生、デバッグ、および編集を行うことができます。また、資産を BLOB に保存したり、SAP/R3 データベースから BLOB を読み込んだり、テスト スクリプト引数を管理したりできます。こちらの方法を推奨します。
- Silk Test Workbench のスタンドアロン版を eCATT から別に開始します。Silk Test Workbench と eCATT 間で、データを BLOB で転送できます。

eCATT との統合のセットアップ

Silk Test Workbench と eCATT 間の統合を設定するには、以下の手順を実行します。

1. 永続的な変更を行うために、SAP サーバーに管理者としてログインします。
2. アプリケーション サーバーでスクリプトを有効化します。追加の情報については、SAP ヘルプのトピック「アプリケーションサーバでのスクリプト有効化」を参照してください。
3. SAP サーバー上で、eCATT の外部ツールとして Silk Test Workbench を登録します。追加の情報については、SAP ヘルプのトピック「eCATT での外部ツール使用準備」を参照してください。
4. 各クライアントに対して、クライアントのフロントエンドでスクリプトを有効化します。

 **注:** Silk Test Workbench が SAP アプリケーションに対して記録しようとするたびに確認ダイアログ ボックスが表示されることを避けるために、**実行中の GUI にスクリプトが接続する時に通知する** チェック ボックスをオフにします。

追加の情報については、SAP ヘルプのトピック「フロントエンドでのスクリプト有効化」を参照してください。

5. SAP サーバー上で、新しいユーザー アカウントを作成します。
6. eCATT と組み合わせて Silk Test Workbench を使用しているクライアント マシンに、Silk Test Workbench と SAPGUI クライアントの両方をインストールします。

eCATT への Silk Test Workbench の登録

Silk Test Workbench と eCATT を統合するには、*ECCUST_ET* SAP テーブルに Silk Test Workbench を登録する必要があります。SET_EXTERNAL_TOOL メソッドは、*ECCUST_ET* テーブルに必要な項目を作成します。SET_EXTERNAL_TOOL 汎用モジュールは SE37 トランザクションを使用して呼び出すことができます。

1. 最初の画面で、汎用モジュール名 SET_EXTERNAL_TOOL. を入力します。
2. 汎用モジュール メニューから、**テスト > 単一テスト** を選択します。

3. 次のウィンドウで、次のパラメータと値を入力します。

パラメータ	値
TOOL_NAME	Silk Test Workbench
PROG_ID	SilkTest.STWEcattControl
TOOL_DESC	Silk Test Workbench for eCATT
TOOL_DATABASE	STW_EDIT
TOOL_RUN_DB	STW_EXEC
TOOL_NO_PWD	<blank>
TOOL_NO_DB	<blank>

4. **実行** ボタン (F8) を押します。

特定のユーザー アカウントの作成

外部ツールを使用して eCATT 統合を利用するために、システム管理者はシステムに標準ユーザーを生成する必要があります。これは、SE38 の ECATT_GENERATE_ET_USER プログラムを各システムに対して一度だけ実行することによって行われます。

レポートを実行した後に、新しく作成したユーザー ロールをアクティブにするには、次の手順を実行します。

1. PFCG トランザクションでロール SAP_ECET を入力し、**Change** を選択します。表示される警告は無視してかまいません。
2. **Authorizations** タブに切り替えて、**Change Authorization Data** を選択します。
3. **SAP_ECET** ツリーのトップ ノード上にカーソルを移動し、**Authorizations > Generate** を選択します。
4. **Back** をクリックして、ロールのメンテナンス画面に戻ります。
5. **Save** をクリックします。

Silk Test Workbench のスタンドアロン版の構成

このセクションのトピックでは、Silk Test Workbench のスタンドアロン版を eCATT からとは別に開始した場合に、Silk Test Workbench と eCATT 間で BLOB でデータを転送したい場合に、行わなければならない追加の構成について説明します。Silk Test Workbench を eCATT 内から開始した場合は、この設定を行う必要はありません。

SAP/R3 ログイン パラメータの設定

SAP/R3 ログイン パラメータは、SAP/R3 データベースにログオンするために使用されます。Silk Test Workbench を使用して SAP/R3 データベースにログオンし、Silk Test Workbench と eCATT の間で BLOB の取得や転送ができます。

1. 以下のいずれかを実行します。
 - a) **BLOB の保存** ダイアログ ボックスから **ログイン パラメータの変更** ボタンをクリックします。
 - b) Silk Test Workbench のメニューから **ツール > SAP eCATT > SAP/R3 ログイン パラメータ** をクリックします。

SAP/R3 ログイン ダイアログ ボックスが表示されます。eCATT に接続している場合は、設定されたログオン パラメータが表示されます。この場合、パスワードを再入力するだけでログオンできます。すでに接続している場合は、ステップ 3 に進んでください。

現在 eCATT に接続していない場合、SAP eCATT データベースにワンタイム ログオンを実行する必要があります。SAP ではログオン情報が維持されるため、一度ログインしたあとは、パスワードを入力するだけで再接続してデータを転送できます。

2. 3桁の SAP 固有のクライアントセッション番号を **クライアント** フィールドに入力します。クライアント番号の詳細については、SAP SDK のマニュアルを参照してください。
3. SAP/R3 データベースのログオン ユーザー ID を **ユーザー名** フィールドに入力します。
4. SAP/R3 のパスワードを **パスワード** フィールドに入力します。
5. 有効な SAP 言語コードを **言語** フィールドに入力します。
6. **OK** をクリックして接続するか、**キャンセル** をクリックして接続せずに終了します。

RFC 接続パラメータの設定または変更

RFC インターフェイスは、Silk Test Workbench と SAP/R3 データベース間で BLOB、引数コンテナ、および実行情報を転送するのに使用されます。Silk Test Workbench と SAP eCATT の間で BLOB の保存と読み込みを行うには、接続情報が必要となります。以下の手順に従って、RFC インターフェイスの接続パラメータを変更してください。

1. eCATT インターフェイスを使用している場合は、Silk Test Workbench のアクティブ セッションに切り替えます。
2. 以下のいずれかを実行します。
 - a) **BLOB の保存** ダイアログ ボックスから **RFC パラメータの変更** ボタンをクリックします。
 - b) Silk Test Workbench のメニューから **ツール > SAP eCATT > RFC 接続パラメータ** をクリックします。

RFC 接続パラメータ ダイアログ ボックスが表示されます。

3. 以下のいずれかを実行します。

現在のセッションに対して RFC 接続をすでに確立してある場合は、**RFC 接続パラメータ** ダイアログ ボックスにパラメータが表示されます。これらのパラメータを変更する必要がない場合は、ステップ 4 へ進んでください。

現在のセッションに対して RFC 接続をまだ確立していない場合や、RFC 接続パラメータを変更する必要がある場合は、必要な接続情報を入力します。接続情報は以下のとおりです。

SAP ログオン ID を使用する すでに SAP GUI を使用して **SAP ログオン ID** を作成してある場合は、このボックスをオンにしてログオンを選択します。SAP GUI に項目がない場合は、このチェックボックスと **SAP ログオン ID** は無効になっています。

SAP ログオン ID リストからログオンを選択します。

AS ホスト RFC 接続に使用される SAP アプリケーション サーバーのホスト マシンの名前。これには、ホスト名の前に付く SAP ルーティング文字列を含めることができます。

RFC の種類 Silk Test Workbench の SAP eCATT への RFC 接続に使用されるサーバーの種類。R3 システムの場合、RFC の種類は 3 です。

システム番号 RFC 接続の SAP システム番号。

4. SAP サーバーが SAP GUI クライアントおよび Silk Test Workbench 以外の別のネットワーク上にある場合は、SAP ルーター文字列を eCATT ABAB コードに手動で入力する必要があります。以下のステップを実行します。
 - a) トランザクション se80 を開始します。
 - b) Class/Interface cl_apl_ecatt_extprog を編集します。
 - c) メソッド Launch に移動して次のコードを追加します： `ashost = '<ROUTER-STRING+IP>'`。
ここで、<ROUTER-STRING+IP> は SAP ルーター文字列、IP は SAP サーバーのアドレスです。
5. **OK** をクリックして変更を適用し、ダイアログ ボックスを閉じます。BLOB の保存と SAP/R3 のログインが Silk Test Workbench と eCATT の間で行われると、BLOB が SAP/R3 データベースに転送されます。BLOB が転送されたら、そのテスト スクリプトを eCATT で使用できるようになります。

Silk Test Workbench を eCATT から開始した場合の eCATT スクリプトの記録とテスト

Silk Test Workbench を eCATT から開始した場合、eCATT を使用してテストを定義し、Silk Test Workbench を eCATT の外部テスト ツールとして使用してスクリプトを記録することができます。これにより、テストを一元管理でき、Silk Test Workbench 内のスクリプト作成機能を使用してターゲット アプリケーションの記録とテストを実行できるようになります。Silk Test Workbench で記録されるテストは、.NET スクリプトかビジュアル テストのいずれかになります。

eCATT でのテスト スクリプトの定義

1. テスト スクリプトを eCATT で起動します。それには、eCATT の最初の画面で **テスト スクリプト** を選択し、名前として S_ECATT を入力し、バージョン番号を入力します。
2. eCATT で、Silk Test Workbench を外部ツールとして指定します。
3. ツールバーの **オブジェクトの適用** ボタンをクリックします。SAP eCATT でスクリプト エディタが表示されます。

eCATT でのテスト スクリプト属性の指定

1. eCATT を使用してテスト スクリプト属性を指定します。詳細な手順については、<http://help.sap.com> にある eCATT ヘルプを参照してください。
2. アプリケーション ツールバーの **スクリプト** ボタンをクリックします。**外部ツールのユーザエントリ** ダイアログ ボックスが表示されます。
3. ST_EDIT データベースのユーザー名とパスワードを入力します。
これは、Silk Test Workbench を外部ツールとして使用するために必要です。
4. テスト スクリプトの作成場所とする外部ツール プロジェクトの名前を入力します。この名前は、Silk Test Workbench 内のプロジェクトの名前です。
5. ST_EDIT データベースをクリアしたい場合は、チェック ボックスをオンにします

eCATT から Silk Test Workbench が起動され、eCATT で作成された新しいテスト スクリプトが **コード** ウィンドウに表示されます。これで、テスト スクリプトを記録してテストできます。または、**コード** ウィンドウを閉じて、ビジュアル テストを作成および記録して、ターゲット アプリケーションをテストすることができます。

Silk Test Workbench を使用したスクリプトの記録とテスト

1. ターゲット アプリケーションが実行されており、使用可能な状態であることを確認します。
2. Silk Test Workbench で、.NET スクリプトを記録するか、ビジュアル テストを作成して記録します。
3. 省略可能：任意の引数を作成します。
詳細については、*Silk Test Workbench* における eCATT スクリプト引数のサポート を参照してください。
4. Silk Test Workbench を使用して .NET スクリプトまたはビジュアル テストを再生します。再生用に作成された結果を表示することもできます。
5. テストおよび関係するすべての Silk Test Workbench 資産を BLOB に保存します。保存された BLOB データは SAP/R3 データベースに転送され、eCATT で使用できます。

スタンドアロン Silk Test Workbench を使用した eCATT スクリプトの作成

Silk Test Workbench で eCATT スクリプトを作成し、追加の VB.NET コードを記録または記述してテストしたあと、そのテスト スクリプトを eCATT で使用できるように SAP/R3 データベースに転送できます。

この場合は、BLOB でテスト スクリプト、プロジェクト、および関連資産が eCATT に渡されるまで、Silk Test Workbench はスタンドアロン環境で実行されます (eCATT から起動しません)。

1. Silk Test Workbench を起動します。
2. テスト スクリプトを作成して記録します。



注: eCATT で作成したテスト スクリプトを BLOB で Silk Test Workbench データベースの一時記憶域に読み込み、使用することもできます。

3. テスト スクリプトを再生します。
4. テスト スクリプトの再生用に作成された結果をすべて表示します。
5. eCATT でのテスト スクリプトに必要な引数をすべて作成します。
6. テスト スクリプトとそのプロジェクト、および関連資産を eCATT に転送するには、テスト スクリプトを BLOB に保存します。Silk Test Workbench はスタンドアロン モードで実行されているので、[SAP/R3 ログイン] ダイアログ ボックスが表示されます。
7. SAP/R3 データベース ログイン パラメータを設定し、BLOB 転送を行うための SAP/R3 データベースに接続します。



注: Silk Test Workbench では SAP/R3 データベースに接続するのに有効な SAP/R3 ログイン ユーザー名とパスワードが必要です。具体的な手順については「SAP/R3 ログイン パラメータの設定」を参照してください。

8. 現在のセッションに対して RFC 接続を確立する必要があります。BLOB データを eCATT に転送するために RFC 接続パラメータを設定します。接続が確立されると、eCATT でテスト スクリプトを使用できるようになります。

BLOB の操作

バイナリ ラージ オブジェクト (BLOB) は、SAP eCATT と Silk Test Workbench との間でテスト スクリプトまたはビジュアル テストを転送するのに使用されるコンテナです。BLOB 内の情報には、.NET スクリプトまたはビジュアル テストに属するすべての関連テスト資産 (結果を除く) が含まれ、他の .NET スクリプトまたはビジュアル テストも含まれます。

BLOB には、関連テスト資産以外に、テスト スクリプトまたはビジュアル テストに属するすべてのオプション設定も含まれます。

BLOB は XML ドキュメントです。BLOB の形式は、Silk Test Workbench の内部でのみ使用されます。BLOB のデータ自体は、外部のシステムによって解釈することはできません。

最後に保存したバージョンの BLOB だけが SAP eCATT データベースに保存されます。以前のバージョンは保持されません。BLOB がユーザーによる編集/デバッグのために開かれると、この BLOB はデータベースに再送信されるまで別のユーザーによってアクセスすることはできません。BLOB を再送信すると、以前のバージョンは上書きされます。

BLOB は、Silk Test Workbench で作成 (保存) されます。BLOB を保存すると、その BLOB は、Silk Test Workbench と eCATT 間の RFC インターフェイスを介して、Silk Test Workbench から SAP/R3 データベースに転送されます。

BLOB 追跡情報の入力

1. 開いているテスト スクリプトのオプションがすべて適切に設定されていることを確認します。特に、Silk Test Workbench の再生オプションは eCATT での再生に影響を与えます。
2. Silk Test Workbench のメニューから以下のいずれかをクリックします。
 - a) BLOB を保存して Silk Test Workbench での作業を続けるには、**ツール > SAP eCATT > BLOB の保存 > 保存して続行** をクリックします。
 - b) BLOB を保存して BLOB が SAP/R3 データベースに転送されたときに eCATT に移動するには、**ツール > SAP eCATT > BLOB の保存 > 保存して eCATT へ移動** をクリックします。保存が完了すると、Silk Test Workbench が閉じ、eCATT にフォーカスが設定されます。

BLOB の保存 ダイアログ ボックスが表示されます。このダイアログ ボックスには、最後に読み込まれた BLOB に関する情報が表示されます。これらの情報を入力し直して、新しい BLOB を保存できます。

3. BLOB の一意の名前を [BLOB ID] フィールドに入力します。
4. BLOB のバージョン番号を [バージョン] フィールドに入力することもできます。バージョンを指定しない場合は、「バージョン 1」が使用されます。



注: バージョン番号は検証も増分もされません。現在のバージョンの BLOB だけが SAP/R3 データベースに保存されます。

5. 診断データをローカルの XML ファイルに保存する場合は、[ファイルにダンプ] チェック ボックスをオンにします。
6. BLOB の内容を入力して BLOB の情報の入力を続けます。

BLOB の内容の入力または変更

エクスポートするプロジェクト リストには、Silk Test Workbench データベースにあるプロジェクトのうち自分がアクセス権を持つものがすべて表示されます。



注: 接続情報 セクションは、スタンドアロン版の Silk Test Workbench を使用しているときのみ表示されます。eCATT から Silk Test Workbench を開始した場合には表示されません。

1. 該当するチェック ボックスをオンにして、BLOB に挿入するプライマリ テスト スクリプトを含むプロジェクトを選択します。他のテスト スクリプトに対する呼び出しがプライマリ スクリプトに含まれている場合は、それらのプロジェクトも選択します。プロジェクトを選択すると、定義済みの記録および再生設定グループのすべてが **[記録設定]** および **[再生設定]** リストに表示されます。これらのグループは、Silk Test Workbench の記録および再生のオプションで設定されます。
2. 定義済みの記録オプションのグループを BLOB に含めるには、**記録設定** リスト ボックスからその記録設定グループを選択します。
3. 定義済みの再生オプションのグループを BLOB に含めるには、**再生設定** リスト ボックスからその再生設定グループを選択します。
4. Silk Test Workbench のプライマリ プロジェクトを **プライマリプロジェクト** ボックスから選択します。プライマリ プロジェクトとは、BLOB のプライマリ スクリプトが属する Silk Test Workbench プロジェクトのことです。
5. **資産の種類** ボックスで、BLOB に保存されるテストの種類を決めます。.NET スクリプトまたはビジュアル テストをリストから選択します。
6. **プライマリ テスト スクリプト/ビジュアル テスト** ボックスには、選択したプロジェクトにある使用可能なテスト スクリプトまたはビジュアル テストのすべてが表示され、現在読み込まれている BLOB からの .NET スクリプトまたはビジュアル テスト (これらは一時的に Silk Test Workbench データベースに保存されます) も表示されます。[資産の種類] での選択に応じて、.NET スクリプトまたはビジュアル テストがリストに表示されます。この BLOB のプライマリ .NET スクリプトまたはビジュアル テストをリストから選択します。
7. **保存** をクリックして BLOB を SAP/R3 データベースに保存します。



注:

デフォルトでは、STW_EDIT および STW_EXEC の Silk Test Workbench の DSN は大文字ですが、SAP サーバーにインストールされている DSN の大文字/小文字と一致させる必要があります。たとえば、BLOB の保存時に「STW_Edit が使用できません」というエラーが表示された場合は、DSN 名の大文字/小文字を確認してください。場合によっては、Silk Test Workbench の DSN を「STW_EDIT」から「STW_Edit」に変更する必要があります。

Silk Test Workbench のデフォルトの再生または記録オプションを変更する場合、設定の名前がなくなります。BLOB を保存するには、このような名前のない設定に対して名前を付けることが強制されます。再生と記録の設定は「プロファイル」に保存できます。詳細については、「再生オプションの保存」および「記録オプションの保存と再利用」を参照してください。

Silk Test Workbench で [SAP/R3 ログイン] ダイアログ ボックスが表示され、データベース ログイン パラメータを設定でき、BLOB 転送を行うための SAP/R3 データベースに接続できます。

SAP/R3 データベースにすでに接続している場合は、Silk Test Workbench でデータベースのパスワードを入力するだけで済みます。まだ SAP/R3 データベースに接続していない場合は、Silk Test Workbench で初回接続時に SAP/R3 のログイン ユーザー名とパスワードを入力する必要があります。具体的な手順については「SAP/R3 ログインパラメータの設定」を参照してください。

RFC 接続がすでに確立されている場合は、**AS ホストとクライアントの情報** が **BLOB の保存** ダイアログボックスに表示されます。これらの情報が表示されない場合は、現在のセッションで RFC 接続は確立されていません。その場合、BLOB データを eCATT に転送するには、RFC 接続パラメータを設定する必要があります。「RFC 接続パラメータの設定または変更」を参照してください。

Silk Test Workbench における eCATT スクリプト引数のサポート

eCATT スクリプト引数は、eCATT 環境における重要な概念の 1 つです。これらの引数は、eCATT と Silk Test Workbench テストとの間の「インターフェイス」となります。引数には、eCATT で BLOB テストスクリプトまたはビジュアルテストを実行するために現在の BLOB で必要な追加のデータが含まれています。これらの引数によって、テストスクリプトやビジュアルテストを eCATT 環境で使用できるようになります。

引数は、Silk Test Workbench または eCATT で作成することができ、両方の環境の間でインポートしたりエクスポートしたりできます。

Silk Test Workbench における引数のサポートは、以下の要素で構成されます。

Silk Test Workbench の引数メンテナンス機能

Silk Test Workbench の引数メンテナンス機能では、**eCATT 引数コンテナ** ダイアログボックスを使用して、新しい引数を作成したり、既存の引数を編集または削除したりできます。**eCATT 引数コンテナ** ダイアログボックスを使用して、BLOB を介して eCATT スクリプトとの間でインポートまたはエクスポートできる引数を定義します。定義した引数は、Silk Test Workbench/CATT COM インターフェイスを使用して、eCATT スクリプトと Silk Test Workbench .NET スクリプトまたはビジュアルテストとの間で受け渡すことができます。

Silk Test Workbench は、一度に 1 つの BLOB (引数と関連資産を持つメインスクリプト) しか処理できないので、**eCATT 引数コンテナ** ダイアログボックスには常に現在の BLOB のスクリプト引数が表示されます。

Silk Test Workbench では、BLOB を SAP データベースに格納するとき、BLOB の引数に指定された値を読み取り、引数コンテナで転送して、BLOB と共に格納します。Silk Test Workbench の引数メンテナンス機能は文字列データ型スクリプト引数でのみ機能します。

eCATT スクリプト引数を使用する

Silk Test Workbench .NET スクリプトの場合は、Silk Test Workbench のスクリプト言語にある 2 つのメソッドを使用して、スクリプトとの間で引数値が受け渡されます。Workbench.SetECATTArg() メソッドは、SAP eCATT スクリプトに渡す引数値を作成します。Workbench.GetECATTArg() メソッドは、SAP eCATT 引数コンテナから渡された引数値を取得します。

ビジュアルテストの場合は、eCATT 引数の設定変数ステップを作成して、定義された引数値を eCATT スクリプトにエクスポートします。eCATT 引数の内容の取得変数ステップを作成して、定義された引数値を eCATT スクリプトからインポートしてローカル変数に保存し、その値をビジュアルテストで使用できます。

eCATT スクリプト引数の作成

1. BLOB が読み込まれている間、**ツール > SAP eCATT > 引数コンテナ** をクリックします。**eCATT 引数コンテナ** ダイアログボックスが表示されます。現在読み込まれている BLOB に対して定義された引数がデータテーブルに表示されます。
2. **追加** をクリックします。**引数の追加** ダイアログボックスが表示されます。
3. **名前** フィールドに、引数に付ける名前を入力します。

4. 引数の転送の方向を選択します (**方向** フィールドのリストから **EXPORT** または **IMPORT** を選択します)。
5. **デフォルト値** フィールドに、引数のデフォルト値を入力します。デフォルト値は、eCATT でテスト スクリプトによって使用されるデータです。デフォルト値の詳細については、eCATT のマニュアルを参照してください。
6. **値** フィールドに、引数の実際の値を入力します。この値はテスト スクリプトから取得でき、eCATT で表示可能です。
7. 引数の説明を **説明** フィールドに入力することもできます。説明は、引数の名前がわかりにくい場合にそれを識別しやすくするためのものです。
8. **追加** をクリックして、現在の BLOB の引数コンテナに引数を追加します。**eCATT 引数コンテナ** ダイアログ ボックスで、引数がコンテナ内の項目として表示されます。
9. 他の引数を追加するには、ステップ 1 から 9 を繰り返します。
- 10 現在読み込まれている BLOB に対して必要な引数の作成をすべて終えたら、**閉じる** をクリックして **eCATT 引数コンテナ** ダイアログ ボックスに戻ります。
- 11 **OK** をクリックします。

eCATT スクリプト引数の編集

最近読み込まれた BLOB のプライマリ .NET スクリプトまたはビジュアル テストの引数は、まだ読み込まれている間に変更することができます。現在読み込まれている BLOB の eCATT スクリプト引数を編集するには、Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスを使用します。

1. 引数を変更する BLOB が読み込まれている間、**ツール > SAP eCATT > 引数コンテナ** をクリックします。**eCATT 引数コンテナ** ダイアログ ボックスが表示されます。このダイアログ ボックスのデータ テーブルには、現在読み込まれている BLOB に対して定義された引数がすべて表示されます。
2. パラメータを編集する引数を選択します。
3. **編集** をクリックします。**引数の編集** ダイアログ ボックスが表示されます。
4. 必要に応じて引数の値を変更します。

名前 引数の名前。

方向 引数が「IMPORT」(eCATT から Silk Test Workbench へ) なのか「EXPORT」(Silk Test Workbench から eCATT へ) なのかを決めます。

デフォルト値 引数のデータのデフォルト値です。デフォルト値は、eCATT スクリプト引数に値が指定されていない場合に、eCATT でテスト スクリプトによって使用されるデータです。デフォルト値の詳細については、eCATT のマニュアルを参照してください。

値 引数の実際の値を入力します。この値はスクリプトから取得でき、eCATT で表示可能です。

説明 説明は、引数の名前がわかりにくい場合にそれを識別しやすくするためのものです。

5. **編集** をクリックします。引数のパラメータが **eCATT 引数コンテナ** ダイアログ ボックスに表示されます。
6. 引数コンテナの他の引数を編集する場合は、ステップ 1 と 2 を繰り返します。
7. 現在読み込まれている BLOB に対して必要な引数の編集をすべて終えたら、**閉じる** をクリックして **eCATT 引数コンテナ** ダイアログ ボックスに戻ります。
8. **OK** をクリックします。

引数コンテナからの eCATT 引数の削除

最近読み込まれた BLOB のスクリプト引数を、いつでもその引数コンテナから削除することができます。ただし、Silk Test Workbench では、引数を削除する前に確認のメッセージが表示されないので注意してください。

1. BLOB が読み込まれている間、**ツール > SAP eCATT > 引数コンテナ** をクリックします。**eCATT 引数コンテナ** ダイアログ ボックスが表示されます。このダイアログ ボックスのデータ テーブルには、最

後に読み込んだ BLOB に対して定義された引数がすべて表示されます。各引数が別々の行に表示されます。

2. 削除する引数がある行を選択して **削除** をクリックします。引数を含む行が引数コンテナから削除されます。
3. 削除する他の引数に対してステップ 2 を繰り返します。
4. **OK** をクリックします。

eCATT スクリプトからビジュアル テストへの引数のインポート

引数コンテナから eCATT スクリプト引数をインポートして、任意のビジュアル テストで引数の値を使用できます。渡された引数の内容は、ビジュアル テストのローカル変数に保存されます。

ビジュアル テストで eCATT スクリプトから引数をインポートするには、以下を実行します。

1. eCATT スクリプト引数をインポートするビジュアル テストを開きます。
2. どの引数を eCATT スクリプトからインポートするかを決定します。
3. Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、ビジュアル テストで使用できる eCATT スクリプト インポート引数を作成します。
4. ビジュアル テストで、インポートする引数を保存するローカル変数を作成します。
定義したローカル変数はビジュアル テストの **<<開始>>** ステップに表示されます。ただし、変数は任意のテスト ステップに挿入できます。
5. ビジュアル テストでテスト ステップを選択し、**挿入 > SAP eCATT > 引数の取得** をクリックします。
Silk Test Workbench によって、選択したステップの直後にステップが作成されます。この新しく作成されたステップは、eCATT 引数の値を取得し、この値を変数に格納するのに使用されます。これにより、この値をビジュアル テストで使用できるようになります。ステップのプロパティが **プロパティ** ペインに表示されます。



ヒント: テストステップの最適な構成を保つため、このテスト ステップをビジュアル テストの開始またはその近くに配置します。

6. テストステップの **プロパティ** ペインで、**並べ替えカテゴリ** アイコンをクリックして、プロパティをカテゴリごとにグループ化します。
7. **割り当てカテゴリ** プロパティを更新します。
ステップの **割り当て** プロパティでは、インポートする eCATT 引数と、引数の値を格納するローカル変数を指定します。
 - a) **eCATT 変数名** プロパティに、ビジュアル テストで使用する値を持つ eCATT スクリプト引数の名前を入力します。
指定した eCATT スクリプト引数は、ビジュアル テストの **eCATT 引数コンテナ** で定義した引数名の 1 つと一致しなければなりません。
 - b) **Local variable name** プロパティには、インポートした引数を格納するために作成した変数を含む、ビジュアル テストに対して定義されているすべてのローカル変数がリストされます。指定した eCATT スクリプト引数の値を格納するローカル変数をリストから選択します。

eCATT 引数 '[eCATT スクリプト引数名]' の内容を取得して、ローカル変数 '[ローカル変数名]' に格納します

8. **割り当てカテゴリ** プロパティを更新してビジュアル テストを保存すると、テスト ステップの記述は次のようになります。

```
Get contents of eCATT argument '[eCATT script argument name]' and put into local variable '[local variable name]'
```
9. 省略可能：複数の eCATT スクリプト引数をビジュアル テストにインポートする場合は、手順を繰り返してください。

eCATT スクリプト引数がビジュアル テストにインポートされ、ローカル変数に保存されると、これらのローカル変数は自動ステップの入力値として使用できるようになります。たとえば、ログイン ID に対する


eCATT スクリプト引数の値を、ログイン ID フィールドに対して値を入力する自動ステップのテキスト入力として使用することができます。

ビジュアル テストから eCATT スクリプトに引数をエクスポートする

ビジュアル テストで使用されている引数を eCATT スクリプトに渡すことができます。eCATT スクリプト引数は BLOB に保存する必要があり、ビジュアル テストの引数コンテナ内でエクスポート引数として定義する必要があります。

eCATT スクリプト引数をビジュアル テストから eCATT スクリプトにエクスポートするには、以下を実行します。

1. 引数を eCATT スクリプトにエクスポートするビジュアル テストを開きます。
2. どの引数を eCATT スクリプトにエクスポートするかを決定します。
Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、eCATT スクリプト エクスポート引数を作成します。
3. ビジュアル テストで **<<開始>>** ステップを選択し、**挿入 > SAP eCATT > 引数の設定** をクリックします。
Silk Test Workbench は、eCATT 引数のエクスポート値を設定するステップを作成します。これにより、この値を eCATT スクリプトで使用できるようになります。ステップのプロパティが **プロパティ** ペインに表示されます。

 **ヒント:** テストステップの最適な構成を保つため、このテスト ステップをビジュアル テストの開始またはその近くに配置します。

4. ステップの **プロパティ** ペインで、**並べ替えカテゴリ** アイコンをクリックして、プロパティをカテゴリごとにグループ化します。
5. [割り当てカテゴリ] プロパティを更新して、eCATT 引数の名前とそれに割り当てる値を定義します。
 - a) **eCATT 変数名** プロパティで、値をエクスポートする eCATT スクリプト引数の名前を入力します。
指定する eCATT スクリプト引数は、ビジュアル テストの **eCATT 引数コンテナ** 内で定義されている引数名と一致している必要があります。
 - b) **値** プロパティで、引数の値を入力するか、値の領域を右クリックしてリテラル、変数、式の結果、または ActiveData 値を割り当てます。

[割り当てカテゴリ] プロパティを更新してビジュアル テストを保存すると、ステップは次のようになります。

```
Set eCATT argument '[eCATT script argument name]' to '[value]'
```

6. 省略可能 : ビジュアル テストから eCATT スクリプトに複数の引数をエクスポートする場合は、この手順を繰り返してください。
引数がエクスポートされ、BLOB が保存されると、eCATT スクリプトで引数値を使用できます。

eCATT スクリプトから .NET スクリプトへの引数のインポート

引数コンテナから eCATT スクリプト引数を取得し、Silk Test Workbench の任意の VB .NET スクリプトで引数の値を使用します。

eCATT スクリプトから Silk Test Workbench の VB .NET スクリプトへ引数をインポートするには、以下を実行します。

1. eCATT スクリプト引数をインポートする VB .NET スクリプトを開きます。
2. どの引数を eCATT スクリプトから VB .NET スクリプトにインポートするかを決定します。
Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、VB .NET スクリプトで使用できる eCATT スクリプト インポート引数を作成します。
3. .NET スクリプトで、インポートする引数を保存するローカル変数を作成します。

例 :

```
Dim argName As String
```

4. Workbench クラスの GetECATTArg 関数を使用して、eCATT スクリプト引数を取得します。

例：

```
ArgName = Workbench.GetECATTArg("NAME")
```

5. 省略可能：GetECATTArg 関数は文字列を返します。.NET スクリプトで型キャストを使用して、日付など他の型に変更できます。

eCATT スクリプト引数が .NET スクリプトにインポートされると、引数は自動ステップの入力値として使用できるようになります。たとえば、ログイン ID に対する eCATT スクリプト引数の値を、ログイン ID フィールドに対して値を入力する自動ステップのテキスト入力として使用することができます。

.NET スクリプトから eCATT スクリプトに引数をエクスポートする

Silk Test Workbench の VB .NET スクリプトで使用した引数を eCATT スクリプトに渡して、スクリプトで引数を使用できます。

eCATT スクリプト引数を .NET スクリプトから eCATT スクリプトにエクスポートするには、以下を実行します。

1. 引数を eCATT スクリプトにエクスポートする .NET スクリプトを開きます。
2. どの引数を eCATT スクリプトにエクスポートするかを決定します。
3. Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、.NET スクリプトで使用できる eCATT スクリプト エクスポート引数を作成します。
4. Workbench クラスの SetECATTArg 関数を使用して、eCATT スクリプト引数をエクスポートします。

例：

```
Workbench.SetECATTArg("OUTPUTNAME","TestName")
```

この例は、スクリプト引数 *TestName* を .NET スクリプトから eCATT のスクリプト引数 *OUTPUTNAME* にエクスポートしています。

 **注：** 指定した eCATT スクリプト引数は、**eCATT 引数コンテナ** で定義した引数名の 1 つと一致しなければなりません。

eCATT から Silk Test Workbench でテストを実行する


ビジュアル テストまたは .NET スクリプトを eCATT で実行するには、ビジュアル テストまたは .NET スクリプトを含む BLOB を作成し保存する必要があります。**BLOB の保存** ダイアログ ボックスの **BLOB コンテンツ** 領域にある **プライマリ テスト スクリプト/ビジュアル テスト** 設定によって、どのテストを実行するかを決定するために eCATT によって使用されます。

ビジュアル テストまたは .NET スクリプトを Silk Test Workbench で実行する eCATT でスクリプトを実行するには：

1. テストを含んだ BLOB が保存されていない場合、BLOB を保存します。Silk Test Workbench で、**ツール > SAP eCATT > BLOB の保存 > 保存して eCATT へ移動**.**BLOB の保存** ダイアログ ボックスの **BLOB コンテンツ** 領域にある **プライマリ テスト スクリプト/ビジュアル テスト** 設定によって、どのテストを実行するかを決定するために eCATT によって使用されます。
2. eCATT で **オブジェクト ディレクトリ エントリ登録** ダイアログ 傑巣が開きます。**パッケージ** の名前を入力し、**保存** をクリックします。以上でスクリプトを eCATT UI で実行する準備が整いました。
3. eCATT UI で **実行** をクリックします。**開始オプション** ダイアログ ボックスが開きます。
4. **外部ツール** 領域で、Silk Test Workbench データベースの **ユーザー名** と **パスワード** を入力します。
5. 省略可能：テストが実行されている間に Silk Test Workbench の UI で何が起きているのか確認したい場合は、**外部ツールの開始モード** リスト ボックスから**外部ツールのユーザインタフェース**を使用して**実行**を選択します。
リスト ボックスから **デバッグ モード** を選択した場合、スクリプトは最初の行で停止します。
6. **実行** をクリックします。eCATT は Silk Test Workbench を開始し、情報を送信してスクリプトを実行します。

eCATT で Silk Test Workbench テスト結果を表示する

Silk Test Workbench でビジュアル テストまたは .NET スクリプトを実行して生成されたテスト結果を eCATT で表示することができます。eCATT が Silk Test Workbench でビジュアル テストまたは .NET スクリプトを実行すると、ログが eCATT に作成されます。このログには、Silk Test Workbench テストの結果が含まれています。

 **注:** Silk Test Workbench テスト結果に含まれるスクリーンショットは、ログには含まれません。

Windows API ベースのアプリケーションのサポート

Silk Test Workbench は、Microsoft Windows API ベースのアプリケーションのテストを組み込みでサポートしています。アクセシビリティを有効にすると Microsoft のアプリケーションのいくつかのオブジェクトが Silk Test Workbench によってより詳細に認識されます。たとえば、アクセシビリティを有効にしないと、Silk Test Workbench は Microsoft Word のメニューバーおよびバージョン 7.0 より後の Internet Explorer に表示されるタブについて基本的な情報のみを記録します。ただし、アクセシビリティを有効にすると、Silk Test Workbench によってそれらのオブジェクトがすべて認識されます。必要な場合、新しいウィンドウを定義すると、Silk Test Workbench によるオブジェクトの認識を向上させることもできます。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

Windows ベースのテストで利用可能な記録および再生コントロールの完全な一覧については、「[Win32 クラス リファレンス](#)」を参照してください。

Win32 クラス リファレンス


Win32 アプリケーションを設定すると、Silk Test Workbench は標準の Windows API ベースのコントロールのテストのサポートを組み込みで提供します。

Windows API ベースのクライアント/サーバー アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジ ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Windows API ベースのクライアント/サーバー アプリケーションがサポートする属性は次のとおりです。

- caption
- windowid
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケーターに使用されます。コントロールの **priorlabel** 値 (テキスト ボックスなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Win32 テクノロジ ドメインにおける priorLabel の決定方法

Win32 テクノロジ ドメインにおいて priorLabel を決定する場合、同じウィンドウ内のすべてのラベルとグループが対象のコントロールとみなされます。以下の条件に従って、コントロールが決定されます。

- コントロールの上または左側にあるラベル、およびコントロールを囲むグループが priorLabel の候補とみなされます。
- 最も単純なケースでは、コントロールに最も近いラベルが priorLabel として使用されます。
- コントロールからの距離が等しい 2 つのラベルが存在する場合、次の条件に基づいて priorLabel が決定されます。
 - 一方のラベルがコントロールの左側にあり、他方が上にある場合、左側のものが優先されます。
 - 両方のラベルがコントロールの左側にある場合、上にあるものが優先されます。
 - 両方のラベルがコントロールの上にある場合、左側のものが優先されます。
- 最も近いコントロールがグループ コントロールである場合、まずグループ内のすべてのラベルが上記の規則に従って決定されます。グループ内に適切なラベルが見つからない場合は、グループのキャプションが priorLabel として使用されます。

埋め込み Chrome アプリケーションのテスト

埋め込み Chrome アプリケーションは、Chromium コアをベースとした埋め込み Web ブラウザー エンジンを使用したデスクトップ アプリケーションです。このようなアプリケーションでは、デスクトップ アプリケーションに Web ブラウザーの機能を追加できます。Chromium Embedded Framework (CEF) や Electron フレームワークを使用して、このようなアプリケーションを作成できます。

Silk Test Workbench は、--remote-debugging-port コマンド ライン引数を指定してリモート デバッグを有効にした埋め込み Chrome アプリケーションのテストをサポートします。Silk Test Workbench では、Java (Java AWT や Swing アプリケーションなど) をベースにした埋め込み Chrome アプリケーションのテストをサポートしません。

Silk Test Workbench を使用して埋め込み Chrome アプリケーションをテストするには、アプリケーションの実行可能ファイルにデバッグ ポートを設定する必要があります。コマンド ラインからアプリケーションを開始して、リモート デバッグ ポートを設定します。

- Silk Test Workbench は、-remote-debugging-port 引数が埋め込み Chrome アプリケーションのコマンド ライン引数に設定されているかどうかを確認します。引数が設定されている場合、Silk Test Workbench は **埋め込み Chrome サポートの有効化** フィールドに適切な実行可能ファイルとデバッグ用ポートを自動的に設定します。
- -remote-debugging-port 引数が埋め込み Chrome アプリケーションのコマンド ライン引数に設定されていない場合は、実行可能ファイルとポートを **埋め込み Chrome サポートの有効化** フィールドに次の手順で指定する必要があります。
 1. Silk Test Workbench UI で、**オプションの編集** を選択します。
 2. **オプション** ダイアログで、**詳細設定** タブを選択します。
 3. **埋め込み Chrome サポートの有効化** オプションで、実行可能ファイルとポートの値ペアを、カンマ区切りで次のように指定します。

```
<application name>.exe=<port number>
```



注: Silk Test Workbench によるリモート デバッグを許可しない埋め込み Chrome アプリケーションをテストすることはできません。



注: Silk Test Workbench では、Electron アプリの非ブラウザー メニューのテストをサポートしません。

例

たとえば、コマンドラインから *myApp* アプリケーションを次のように実行します。

```
myApp.exe --remote-debugging-port=9222
```

この場合、**埋め込み Chrome サポートの有効化** オプションに実行可能ファイルとポートの値ペアを次のように指定します。

```
myApp.exe=9222
```


MFC (Microsoft Foundation Class) のサポート

MFC (Microsoft Foundation Class) のクラス ID は、時間がたつと変更されることがあるため、安定したロケータを生成するために使用することはできません。不安定なロケータの生成を避けるため、Silk Test Workbench では次の属性をロケータとして使用します。

- MFC クラス名 (MFC コントロールの Windows クラス名が Afx: で始まる場合)
- Windows クラス名 (MFC コントロールの Windows クラス名が Afx: で始まらない場合)

Silk Test Workbench は MFC バージョン 140 のみをサポートします。また、サポートする組み合わせは次の通りです。

- Release、x86、MBCS
- Release、x86、Unicode
- Debug、x86、MBCS
- Debug、x86、Unicode
- Release、x64、MBCS
- Release、x64、Unicode
- Debug、x64、MBCS
- Debug、x64、Unicode

 **注:** Silk Test Workbench 18.5 以前で生成した MFC コントロールのロケータを持つ既存のテストを実行する場合は、影響するテスト スクリプトで OPT_COMPATIBILITY オプションをバージョン 18.5.0 以前に設定してください。

```
'VB .NET code  
Agent.SetOption("OPT_COMPATIBILITY", "18.5.0")
```

クロスブラウザ テスト

Silk Test Workbench を使用すると、単一のポータブルなビジュアル テストやテスト スクリプトできまざまなブラウザに対して非常に高度な Web アプリケーションの機能を簡単に検証できます。Silk Test Workbench は、最新の Web テクノロジーを使用した、効率的でメンテナンスしやすいクロスブラウザ テストのトップレベルのサポートを提供します。

テスト自動化における主要な困難のひとつが、テストの作成と保守にかかるコストの効率化です。ブラウザによって動作が異なるため、Web アプリケーションの検証を生産的に実行することは困難です。Silk Test Workbench は、クロスブラウザ テストにおける次の 3 つの領域をうまく扱うため、ユーザーはテストの記述に集中することができます。

組み込みの同期処理 これにより、サポートするすべてのブラウザ上で実行するスクリプトを作成することができます。AJAX や HTML5 などの高度な動的 Web アプリケーションに特有の非同期イベントを手動で同期する必要はありません。Silk Test Workbench は、HTML や AJAX だけでなく、Apache Flex、Microsoft Silverlight、HTML5/AJAX など、すべての主要な Web 環境に対する同期モードをサポートします。詳細については、「[xBrowser のページ同期](#)」を参照してください。

統合オブジェクトモデル Silk Test Workbench では、さまざまなブラウザ上で幅広く実行するテストを作成、保守できます。すべてのブラウザに対応する統合オブジェクト モデルにより、ユーザーはテストの作成と保守を単一のブラウザを中心に行うことができます。Silk Test Workbench は、すべての他のブラウザ上のオブジェクトに同じ方法でアクセスできるようにすることで、さまざまなブラウザに対する回避策を探すことなく、時間を節約してテストの作成に集中できます。

クロスブラウザスクリプトの記録 スクリプトを記録すると、修正することなくすべての他のブラウザで再生できます。これにより、テスト スクリプトの作成と保守にかかる時間と労力は劇的に減少します。シミュレーションではなく、テストは実際のブラウザ上で実行されます。つまり、テストは、エンド ユーザーの動作とまったく同じように動作します。

Silk Test Workbench では、次のブラウザを使用して Web アプリケーションのテストを再生できます。

- Internet Explorer
- Mozilla Firefox (Microsoft Windows 上または macOS 上)
- Google Chrome (Microsoft Windows 上または macOS 上)
- Microsoft Edge
- Chrome for Android (Android デバイス上)
- Apple Safari (macOS 上または iOS デバイス上)
- 埋め込みブラウザ コントロール



注: 次のブラウザのいずれかを使用して Web アプリケーションのテストを記録できます。

- Internet Explorer
- Microsoft Edge
- Mozilla Firefox (Microsoft Windows 上または macOS 上)
- Google Chrome 50 以降 (Microsoft Windows 上または macOS 上)
- モバイル ブラウザ (モバイル デバイス上)

クロス ブラウザー テスト用にスクリプトを記録する場合、Google Chrome、Mozilla Firefox、または Microsoft Edge を使用することを Micro Focus では推奨しています。Internet Explorer を使って Silk Test Workbench で記録したスクリプトは、ほかのブラウザで記録したスクリプトと若干異なる場合があります。



注: Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザ アドオンを無効にします。Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリース ノート](#)』を参照してください。

サンプル アプリケーション

Silk Test のサンプル Web アプリケーションには、以下の URL からアクセスします。


- <http://demo.borland.com/InsuranceWebExtJS/>
- <http://demo.borland.com/gmopost>
- <http://demo.borland.com/gmoajax>

テストを再生するブラウザーの選択

テストを再生するために使用するブラウザーを定義できます。

- Silk Test Workbench の UI からテストを実行する場合、**ブラウザーの選択** ダイアログ ボックスが表示され、このダイアログ ボックスで選択したブラウザーが使用され、テスト スクリプトで設定されているブラウザーを Silk Test Workbench は無視します。

- **ブラウザーの選択** ダイアログ ボックスが無効の場合 (**再び表示しない** をチェックした場合)、個々のテスト スクリプトのアプリケーション構成によってテストを実行するために使用するブラウザーが決定されます。

 **注: ブラウザーの選択** ダイアログ ボックスを再び有効にするには、**ツール > オプション** をクリックして、**ブラウザー構成ダイアログの表示** オプションを **はい** に設定します。

- スクリプトをコマンド ラインや CI サーバーから実行する場合は、スクリプトのアプリケーション構成で接続文字列を指定します。

アプリケーション構成で指定したブラウザーは、`silktest.configurationName` 環境変数、または `browser` コマンドライン パラメータ を使用して上書き指定できます。

- Silk Central からテストを実行する場合は、テストするブラウザーそれぞれの構成を持つ構成スイートを作成します。そして、適切な構成名を指定します。詳細については、『[Silk Central ヘルプ](#)』を参照してください。

silktest.configurationName 環境変数を使用したブラウザーの設定の例

- ブラウザーとして Internet Explorer を使用する場合は、次のように入力します。

```
SET silktest.configurationName=InternetExplorer
STW.exe -script MyScript
```

- ブラウザーとして Microsoft Edge を使用する場合は、次のように入力します。

```
SET silktest.configurationName=Edge
STW.exe -script MyScript
```

- ブラウザーとして Mozilla Firefox を使用する場合は、次のように入力します。

```
SET silktest.configurationName=Firefox
STW.exe -script MyScript
```

- ブラウザーとして Google Chrome を使用する場合は、次のように入力します。

```
SET silktest.configurationName=GoogleChrome
STW.exe -script MyScript
```

- ブラウザーとして Mac 上の Apple Safari を使用する場合は、次のように入力します。

```
SET silktest.configurationName=host=10.0.0.1 - Safari
STW.exe -script MyScript
```

この例では、`host` は Apple Safari をテストする Mac です。ホストは、リモート ロケーションとして Silk Test Workbench がインストールされているマシンに接続されている必要があります。詳細については、「[リモート ロケーションの編集](#)」を参照してください。

- ブラウザーとして Android デバイス上の Google Chrome を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモート マシンに接続されている場合は、次のように入力します。

```
SET
silktest.configurationName="platformName=Android;deviceName=MotoG3;
deviceId=11111111;host=10.0.0.1 - Chrome"
STW.exe -script MyScript
```

- ブラウザーとして iOS デバイス上の Apple Safari を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモート マシンに接続されている場合は、次のように入力します。

```
SET silktest.configurationName="platformName=iOS;deviceName=iPad
mini;deviceId=11111111;host=10.0.0.1"
STW.exe -script MyScript
```

さらに、アプリケーション構成でブラウザーを指定する必要があります。



ヒント: すべての例で、環境変数 `silktest.configurationName` を設定する代わりに、Java システムプロパティ - `Dsilktest.configurationName` を設定して、ブラウザを設定することもできます。たとえば、ブラウザとして Mac 上の Apple Safari を使用する場合は、次のように入力することもできます。

```
-Dsilktest.configurationName=host=10.0.0.1 - Safari  
STW.exe -script MyScript
```



ヒント: Silk Test Workbench の UI から再生または記録を開始すると、**ブラウザの選択** ダイアログ ボックスが開き、システムで現在利用可能なブラウザのリストが表示されます。

browser パラメータを使用したブラウザの設定の例

- ブラウザーとして Internet Explorer を使用する場合は、次のように入力します。
`STW.exe -browser InternetExplorer -script MyScript`
- ブラウザーとして Microsoft Edge を使用する場合は、次のように入力します。
`STW.exe -browser Edge -script MyScript`
- ブラウザーとして Mozilla Firefox を使用する場合は、次のように入力します。
`STW.exe -browser Firefox -script MyScript`
- ブラウザーとして Google Chrome を使用する場合は、次のように入力します。
`STW.exe -browser GoogleChrome -script MyScript`
- ブラウザーとして Mac 上の Apple Safari を使用する場合は、次のように入力します。

```
STW.exe -connectionstring host=10.0.0.1 - Safari -script MyScript
```

この例では、`host` は Apple Safari をテストする Mac です。ホストは、リモートロケーションとして Silk Test Workbench がインストールされているマシンに接続されている必要があります。詳細については、「リモートロケーションの編集」を参照してください。

- ブラウザーとして Android デバイス上の Google Chrome を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。

```
STW.exe -connectionstring  
"platformName=Android;deviceName=MotoG3;host=10.0.0.1 - Chrome" -  
script MyScript
```

- ブラウザーとして iOS デバイス上の Apple Safari を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。

```
STW.exe -connectionstring "platformName=iOS;deviceName=iPad  
mini;host=10.0.0.1" -script MyScript
```

さらに、アプリケーション構成でブラウザを指定する必要があります。

xBrowser でのテスト オブジェクト

Silk Test Workbench では、以下のクラスを使用して Web アプリケーションがモデル化されます。

クラス	説明
BrowserApplication	Web ブラウザのメイン ウィンドウを公開し、タブ化するための方法を提供します。
BrowserObject	BrowserApplication に含まれるすべてのオブジェクトの基底クラスを表します。
BrowserWindow	タブおよび埋め込みブラウザ コントロールへのアクセスを提供し、異なるページに移動するための方法を提供します。
DomElement	Web アプリケーションの DOM ツリー (フレームを含む) を提供し、すべての DOM 属性へのアクセスを提供します。一部の DOM 要素では、特殊なクラスを使用できます。
DomButton	Web ページの最上位のコンテナを表します。DomElement を介して DOM ツリーを公開します。
DomCheckBox	<input type='checkbox'> タグを使用して指定された、すべての DOM 要素を表します。
DomForm	<form> タグを使用して指定された、すべての DOM 要素を表します。
DomLink	<a> タグを使用して指定された、すべての DOM 要素を表します。
DomListBox	<select> タグを使用して指定された、すべての DOM 要素を表します。
DomRadioButton	<input type='radio'> タグを使用して指定された、すべての DOM 要素を表します。
DomTable	<table> タグを使用して指定された、すべての DOM 要素を表します。
DomTableRow	<tr> タグを使用して指定された、すべての DOM 要素を表します。
DomTextField	次のタグのいずれかを使用して指定された、すべての DOM 要素を表します。 <ul style="list-style-type: none"> • <input type='text'> • <input type='password'> • <input type='file'> • <textarea>

xBrowser オブジェクトに対するオブジェクト解決

xBrowser テクノロジ ドメインは動的オブジェクト解決をサポートします。

テストケースは、オブジェクトを検索し、識別するためにロケータ文字列を使用します。一般的なロケータには、"`LocatorName[@locatorAttribute='value']`" のようにロケータ名と少なくとも 1 つのロケータ属性が含まれます。

ロケータ名 Java SWT などの他の種類のテクノロジーでは、テスト オブジェクトのクラス名を使用してロケータ名が作成されます。xBrowser では、DOM 要素のタグ名もロケータ名として使用できます。以下のロケータは、同じ要素を示しています。

1. タグ名を使用した場合 : "`//a[@href='http://www.microfocus.com']`"
2. クラス名を使用した場合 : "`//DomLink[@href='http://www.microfocus.com']`"


再生速度を最適化するには、クラス名ではなくタグ名を使用します。

ロケータ属性 すべての DOM 属性は、ロケータ文字列属性として使用できます。たとえば、要素 `<button automationid='123'>Click Me</button>` はロケータ `"/button[@automationid='123']"` を使用して識別できます。

ロケータの記録 Silk Test Workbench では、テストを記録したり、**オブジェクトの識別** ダイアログ ボックスを使用したりするときに、組み込みロケータ生成プログラムが使用されます。特定のアプリケーションの結果を向上するように、ロケータ生成プログラムを構成することができます。

xBrowser のページ同期

同期は、すべてのメソッド呼び出しの前後に自動的に実行されます。メソッド呼び出しは、同期条件が満たされるまで開始せず、終了もしません。

 **注:** プロパティのアクセスは同期されません。

同期モード

Silk Test Workbench には、HTML および AJAX 用の同期モードがあります。


HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれるより複雑なシナリオが使用される場合は、以下の同期関数を使用して、手動でスクリプトを記述することが必要になることがあります。

- WaitForObject
- WaitForProperty
- WaitForDisappearance
- WaitForChildDisappearance

AJAX モードでは、ブラウザがアイドル状態に類似した状態になるまで待機します。このことは、AJAX アプリケーションまたは AJAX コンポーネントを含むページに対して特に効果的です。AJAX モードを使用すると、同期関数を手動で記述する必要がなくなるため、スクリプト（オブジェクトの表示または非表示を待機したり、特定のプロパティ値を待機するなど）の作成処理が大幅に簡略化されます。また、この自動同期は、スクリプトを手動で適用しないで記録と再生を正常に行うための基礎となります。

トラブルシューティング


AJAX の非同期の特性のため、ブラウザが完全にアイドル状態になることはありません。このため、Silk Test Workbench でメソッド呼び出しの終了が認識されず、特定のタイムアウト時間が経過したあとで、タイムアウト エラーが発生することがまれにあります。この場合は、少なくとも、問題が発生する呼び出しに対して、同期モードを HTML に設定する必要があります。

 **注:** 使用するページ同期メソッドにかかわらず、Flash オブジェクトがサーバーからデータを取得し、計算を実行してデータをレンダリングするテストでは、手動でテストに同期メソッドを追加する必要があります。メソッドを追加しないと、Silk Test Workbench は、Flash オブジェクトが計算を完了するまで待機しません。たとえば、ビジュアルテストでは Delay を使用し、スクリプトでは `System.Threading.Thread.Sleep(milliseconds)` を使用します。

AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を **同期除外リスト** 設定で指定します。

URL を除外フィルタに追加するには、**オプション** ダイアログ ボックスの **同期除外リスト** で URL を指定します。

監視ツールを使用して、同期の問題により再生エラーが発生するかどうかを判断します。たとえば、FindBugs (<http://findbugs.sourceforge.net/>) を使用して、AJAX 呼び出しが再生に影響を及ぼしているかどうかを判断できます。次に、問題が発生するサービスを **同期除外リスト** に追加します。

 **注:** URL を除外すると、指定した URL を対象とする各呼び出しに対して同期が無効になります。その URL に対して必要な同期は、手動で呼び出す必要があります。たとえば、WaitForObject をスクリプトまたはビジュアルテストに手動で追加する必要がある場合があります。手動で数多くの呼び出しを追加することを避けるために、可能なかぎり、最上位の URL ではなく、具体的に対象を絞って URL を除外します。

ページ同期設定の構成

オプション ダイアログ ボックスでは、各ビジュアルテストまたはスクリプトのページ同期設定を個別に構成したり、すべてのテストおよびスクリプトに適用するグローバル オプションを設定したりできます。これらのオプションを構成するには、**ツール > オプション** を選択し、**再生** を展開し、**xBrowser** をクリックします。

ビジュアルテストの個別の設定を構成するには、ビジュアルテストを記録し、次に、グローバル再生値を上書きする **再生設定の設定** ステップを挿入します。設定には、**同期除外リスト**、**同期モード**、**同期タイムアウト** があります。

たとえば、ビジュアルテストでは、**同期モード** 再生設定を **HTML** に設定し、残りのステップでは必要に応じて **同期モード** を **AJAX** に戻します。

スクリプトの個別の設定を構成するには、以下のいずれかを呼び出します。

- `Agent.SetOption(Options.XBrowserSynchronizationMode, mode)`

`mode` には、HTML を使用する場合は 1、AJAX を使用する場合は 2 を入力します。

- `Agent.SetOption(Options.XBrowserSynchronizationExcludes, URL)`

除外するサービスまたは Web ページの URL または URL の一部を指定します。たとえば、`http://example.com/syncsample/timeService` と入力します。

- `Agent.SetOption(Options.XBrowserSynchronizationTimeout, time)`

`time` をミリ秒で指定します。

xBrowser における API 再生とネイティブ再生の比較

Silk Test Workbench では、Web アプリケーション用に API 再生とネイティブ再生がサポートされています。アプリケーションでプラグインまたは AJAX を使用している場合は、ユーザーの入力そのものを使用します。アプリケーションでプラグインまたは AJAX を使用していない場合は、API 再生を使用することをお勧めします。

ネイティブ再生には以下のような利点があります。

- ネイティブ再生では、マウス ポインタを要素上に移動し、対応する要素を押すことによって、エージェントはユーザー入力をエミュレートします。この結果、再生はほとんどのアプリケーションで変更なしで動作します。
- ネイティブ再生では、Flash や Java アプレットなどのプラグイン、および AJAX を使用するアプリケーションをサポートしていますが、高レベルの API 記録はサポートしていません。

API 再生には以下のような利点があります。

- API 再生では、Web ページが `onmouseover` や `onclick` などの DOM イベントによって直接実行されます。
- API 再生を使用するスクリプトでは、ブラウザをフォアグラウンドで実行する必要はありません。
- API 再生を使用するスクリプトでは、要素をクリックする前に、要素が表示されるようにスクロールする必要はありません。
- 一般的に、高レベルのユーザー入力は再生中にポップアップ ウィンドウやユーザー対話の影響を受けないため、API スクリプトの信頼性は高くなります。

- API 再生は、ネイティブ再生よりも高速です。

オプション ダイアログ ボックスを使用して、記録する関数の種類を構成したり、ユーザーの入力そのものを使用するかどうかを指定したりできます。

API 再生とネイティブ再生の関数の違い

DomElement クラスには、API 再生とネイティブ再生に対して異なる関数が備えられています。

以下の表に、API 再生とネイティブ再生で使用する関数を示します。

	API 再生	ネイティブ再生
マウス操作	DomClick	Click
	DomDoubleClick	DoubleClick
	DomMouseMove	MoveMouse
		PressMouse
		ReleaseMouse
キーボード操作	使用不可	TypeKeys
特殊な関数	Select	使用不可
	SetText	
	など	

マウス移動の詳細設定

マウス移動イベントを使用する Web アプリケーション、Win32 アプリケーション、および Windows Forms アプリケーションでマウス移動操作を記録するかどうかを指定します。たとえば、Apache Flex や Swing など、xBrowser テクノロジ ドメインの子ドメインのマウス移動イベントを記録することはできません。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般** をクリックします。
4. マウス移動操作を記録するには、**マウスの移動操作を記録する** リスト ボックスから **はい** を選択します。マウスの移動イベントを使用する Web ページをテストする場合は、**はい** を選択します。Silk Test Workbench では、スクリプトを短くするために、マウスが置かれた要素またはその親が変化するマウスの移動イベントのみが記録されます。
5. マウスの移動操作を記録する場合、MoveMouse 操作が記録される前に、どのくらいの間マウスが不動状態になければならないかを、**マウスの移動記録遅延** テキスト ボックスにミリ秒単位で指定します。デフォルト値は、200 に設定されています。マウスの移動操作は、この時間、マウスが静止している場合にのみ記録されます。遅延を短くすると、予期しないマウスの移動操作が増加します。遅延を長くすると、操作を記録するためにマウスを静止しておく必要があります。
6. **OK** をクリックします。

xBrowser のブラウザ構成の設定

いくつかのブラウザ設定は、テストを継続的に安定して実行するのに役立ちます。設定を変更しなくても Silk Test Workbench は動作しますが、ブラウザ設定を変更するにはいくつかの理由があります。

再生速度を向上させる 読み込みに時間を要する Web ページではなく、about:blank をホーム ページとして使用する

ブラウザの予期しない動作を回避する

- ポップアップ ウィンドウや警告ダイアログ ボックスを無効にする
- オート コンプリート機能を無効にする
- パスワード ウィザードを無効にする


ブラウザの誤動作を防止する 不要なサードパーティ製プラグインを無効にする

以下のセクションでは、対応するブラウザにおけるこれらの設定場所について説明します。

Internet Explorer

ブラウザ設定は、**ツール > インターネット オプション** にあります。以下の表に、調整できるオプションの一覧を示します。

タブ	オプション	設定	コメント
全般	ホーム ページ	about:blank に設定します。	新しいタブの起動時間を最小限に抑えます。
全般	タブ	<ul style="list-style-type: none"> • 複数のタブを閉じるときの警告を無効にします。 • 新しいタブを作成したとき、新しいタブに切り替えます。 	<ul style="list-style-type: none"> • 予期しないダイアログ ボックスが表示されないようにします。 • このようにしないと、新しいタブを開くリンクが正しく再生されない場合があります。
プライバシー	ポップアップ ブロック	ポップアップ ブロックを無効にします。	Web サイトで新しいウィンドウを開くことができることを確認します。
コンテンツ	オートコンプリート	完全にオフにします。	<ul style="list-style-type: none"> • 予期しないダイアログ ボックスが表示されないようにします。 • キー入力するときに予期しない動作を回避します。
プログラム	アドオンの管理	最低限必要なアドオンのみを有効にします。	<ul style="list-style-type: none"> • サードパーティ製アドオンにはバグが含まれていることがあります。 • Silk Test Workbench と互換性がない可能性があります。
詳細設定	設定	<ul style="list-style-type: none"> • Internet Explorer の更新について自動的に確認する を無効にします。 • スクリプトのデバッグを使用しない (Internet Explorer) を有効にします。 • スクリプトのデバッグを使用しない (その他) を有効にします。 • 自動クラッシュ回復機能を有効にする を無効にします。 • スクリプト エラーごとに通知を表示する を無効にします。 • すべての ...警告する 設定を無効にします。 	予期しないダイアログ ボックスが表示されないようにします。

 **注:** 100% 以外の拡大レベルを使用して Internet Explorer で Web アプリケーションを記録すると、期待通り機能しない可能性があります。Internet Explorer で Web アプリケーションに対する操作を記録する前に、拡大レベルを 100% に設定してください。


Mozilla Firefox

Mozilla Firefox のブラウザ設定を変更する必要はありません。Silk Test Workbench により、適切なコマンドラインパラメータが指定され、自動的に Mozilla Firefox が起動します。

 **注:** Web アプリケーションのテスト時に予期しない動作を避けるため、Mozilla Firefox の自動更新を無効にします。詳細については、『[Stop automatic updates](#)』を参照してください。

Google Chrome

Google Chrome のブラウザ設定を変更する必要はありません。Silk Test Workbench により、適切なコマンドラインパラメータが指定され、自動的に Google Chrome が起動します。

 **注:** Web アプリケーションのテスト時に予期しない動作を避けるため、Google Chrome の自動更新を無効にします。詳細については、『[Turning Off Auto Updates in Google Chrome](#)』を参照してください。

リモート デスクトップ ブラウザーの接続文字列

接続文字列は、テストに使用するリモート デスクトップ ブラウザーを指定します。リモート ブラウザーで Web アプリケーションをテストする場合、Silk Test Workbench はリモート ロケーションに接続するために接続文字列を使用します。接続文字列は、アプリケーション構成の主要な一部です。テストする Web アプリケーションを構成するときに、接続文字列は設定されます。接続文字列を変更するには、**アプリケーション構成の編集** ダイアログ ボックスを使用します。

リモート ブラウザーで Web アプリケーションをテストする場合、接続文字列にはリモート マシンの IP アドレスまたはホスト名を意味する host だけが含まれます (10.0.0.1 など)。正しいブラウザを選択するために、Silk Test Workbench はブラウザの種類と共に接続文字列を使用します。ブラウザの種類は、**アプリケーション構成の編集** ダイアログ ボックスでも指定できます。

ホスト名は、大文字と小文字は区別されません。

 **注:** リモート デスクトップ ブラウザーのテストは、リモート Microsoft Windows マシン上の Microsoft Edge およびリモート Mac 上の Apple Safari に対してのみサポートされます。

接続文字列の例

```
"host=10.0.0.1"
```

リモート Windows マシン上でのブラウザのテスト

Microsoft Edge、Google Chrome、Mozilla Firefox をリモート Windows マシン上でテストするには、リモート マシン上に Silk Test Workbench をインストールする必要があります。Microsoft Edge は、Microsoft Windows 10 マシン上でのみ実行できます。Silk Test Workbench は、リモート Windows マシン上の Mozilla Firefox 55 以前のテストをサポートします。

1. ブラウザーのテストを実行するローカル マシン上で、リモート Windows マシンをリモート ロケーションとして追加します。
詳細については、『[リモート ロケーションの編集](#)』を参照してください。
2. Silk Test Information Service がすでにリモート マシン上で管理者権限で実行されている場合は、Silk Test Information Service を無効化します。
 - a) リモート マシンに管理者としてログインします。
 - b) **コントロール パネル** (アイコン表示) を開きます。

- c) **管理ツール** をクリックします。
 - d) **サービス** をダブルクリックします。
 - e) Silk Test Information Service をダブルクリックします。
 - f) サービスが実行中の場合、**停止** をクリックして、サービスの状態が **停止** になるまで待機します。
 - g) **スタートアップの種類** を **無効** に変更します。
 - h) **OK** をクリックします。
3. Silk Test Information Service を標準の整合性レベル（管理者権限無し）で開始します。
 - a) ファイル エクスプローラーを開き、%OPEN_AGENT_HOME%/InfoService に移動します。
たとえば、C:\Program Files (x86)\Silk\SilkTest\ng\InfoService になります。
 - b) infoservice_start.bat をダブルクリックします。
 4. 以上で、リモート マシン上のブラウザーを **アプリケーションの選択** ダイアログ ボックスで選択できるようになります。



注: Silk Test Information Service が、サービス セッションではなくユーザー セッションで実行されていることを確認します。Silk Test Information Service は、UI 操作が有効でなければなりません。セッション 0 であるサービス セッションは、UI 操作が有効ではありません。

Mac 上の Google Chrome または Mozilla Firefox のテスト

リモート macOS マシン上の Google Chrome または Mozilla Firefox をテストするには、リモート マシン上に Silk Test Information Service をインストールする必要があります。詳細については、「[Silk Test Information Service を Mac にインストールする](#)」を参照してください。

1. Google Chrome または Mozilla Firefox をテストするローカル マシン上で、リモート macOS マシンをリモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
2. 以上で、リモート macOS マシン上の Google Chrome または Mozilla Firefox を **アプリケーションの選択** ダイアログ ボックスで選択できるようになります。
3. STW.EXE コマンド ラインを使って、macOS マシン上のリモート ブラウザーに対するスクリプトを実行するには、次のコマンドを使用します：

```
stw -d empty -browser <browsername> -connectionstring <connectionstring> -s <Scriptname>
```

例：

```
stw -d empty -browser Firefox -connectionstring "host=try-imag" -s BrowserScript
```

または

```
stw -d empty -verbose -browser GoogleChrome -connectionstring "host=try-imag" -s BrowserScript
```



注: Silk Test Information Service が、サービス セッションではなくユーザー セッションで実行されていることを確認します。Silk Test Information Service は、UI 操作が有効でなければなりません。セッション 0 であるサービス セッションは、UI 操作が有効ではありません。

Google Chrome 上でテストを実行する際の前提条件と制限事項についての情報は、「[Google Chrome を使用したテスト](#)」セクションにあるトピックを参照してください。Mozilla Firefox 上でテストを実行する際の前提条件と制限事項についての情報は、「[Mozilla Firefox を使用したテスト](#)」セクションにあるトピックを参照してください。

WebDriver ベースのブラウザーのケイパビリティの設定

WebDriver ベースのブラウザー上で Web アプリケーションをテストする場合は、ケイパビリティを設定してブラウザー セッションの設定をカスタマイズできます。

Silk Test Workbench では、次のブラウザーの種類に対する接続文字列で WebDriver ケイパビリティを指定できます。

- Google Chrome
- Mozilla Firefox

Mozilla Firefox 48 以降で利用可能なオプションとケイパビリティについての詳細は、『<https://github.com/mozilla/geckodriver>』を参照してください。Google Chrome で利用可能なオプションとケイパビリティについての詳細は、『[Capabilities & ChromeOptions](#)』を参照してください。

Silk Test Workbench でケイパビリティを設定するには：

1. ケイパビリティを変更する Web アプリケーションのプロジェクトを選択します。
2. プロジェクトの基本状態の接続文字列を編集します。

接続文字列は、次の方法で編集できます。

- **アプリケーション構成の編集** ダイアログ（カスタマイズしたブラウザーに対する操作を記録する場合など）
- スクリプト（カスタマイズしたブラウザーに対してテスト スクリプトを実行するだけの場合）

詳細については、「基本状態」を参照してください。

3. スクリプトを実行して、指定したオプションとケイパビリティでブラウザーを開始します。

例

Mozilla Firefox から実行可能ファイルを自動的にダウンロードするには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code - No linebreaks, this is a single line of code.
baseState.ConnectionString = "moz:firefoxOptions={¥"prefs¥":
{ ¥"browser.download.folderList¥": 2,
¥"browser.helperApps.neverAsk.saveToDisk¥": ¥"application/octet-stream
¥"} }";
```

Mozilla Firefox のダウンロード フォルダを指定するには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code
baseState.ConnectionString = "moz:firefoxOptions={¥"prefs¥":
{ ¥"browser.download.dir¥" : ¥"C:/Download¥"} }";
```

Mozilla Firefox のコマンド ライン引数を設定するには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code
baseState.ConnectionString = "moz:firefoxOptions={¥"args¥":[¥"--devtools
¥"]}";
```

プロキシ サーバーを使って Mozilla Firefox 上でテストを実行するには、次のコードを追加します。

```
' VB .NET code
baseState.ConnectionString = "moz:firefoxOptions={¥"prefs¥":
{ ¥"network.proxy.http¥": ¥"" + PROXY_HOST + "¥",
¥"network.proxy.http_port¥": " + PROXY_PORT + ", ¥"network.proxy.type
¥": 1 } }";
```

Google Chrome から実行可能ファイルを指定したフォルダに自動的にダウンロードするには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code - No linebreaks, this is a single line of code.
baseState.ConnectionString = "chromeOptions={¥"prefs¥":
{¥"profile.default_content_setting_values.automatic_downloads¥":1,
```

```
¥"download.default_directory¥":¥"c:/Download¥",  
¥"download.prompt_for_download¥":false}};"
```

Google Chrome のパスワード マネージャーにメッセージを表示させないようにするには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code - No linebreaks, this is a single line of code.  
baseState.ConnectionString = "chromeOptions={¥"args¥": [¥"--disable-save-  
password-bubble¥"], ¥"prefs¥": {¥"profile.password_manager_enabled  
¥":false, ¥"credentials_enable_service¥":false}};"
```

Mac 上の Apple Safari を使用したテスト

このセクションでは、Silk Test Workbench がインストールされている Windows マシンと接続し、Mac マシン上の Apple Safari をテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Mac 上の Apple Safari を使用したテストにおける前提条件

Mac 上の Apple Safari を使用してテストを行う前に、次の前提条件を満たしていることを確認してください。

- Mac がリモート ロケーションとして Silk Test Workbench がインストールされている Windows マシンに接続されている。詳細については、「リモート ロケーションの編集」を参照してください。
- Apple Safari 9 でテストする場合、Apple Safari 用の WebDriver 拡張である SafariDriver が Mac 上にインストールされている (SafariDriver は従来のクライアント/サーバーの関係を逆転し、WebSocket を使用して WebDriver クライアントと通信します)。Apple Safari 10.1 を使用する場合、Safari はドライバーの実装を組み込みで備えています。
- Java JDK が Mac 上にインストールされている。
- Information Service が Mac 上にインストールされている。Information Service に必要なファイルを取得するには、Silk Test インストーラを使用します。詳細については、「[Silk Test Information Service を Mac にインストールする](#)」を参照してください。
- Apple Safari 上でテストを実行するには、Information Service をインストールしたユーザーが Mac にログインしている必要があります。



ヒント: Micro Focus では、起動時に正しいユーザーで Mac に自動的にログインするように設定することを推奨します。詳細については、「[起動時に自動的にログインするように Mac を設定する](#)」を参照してください。

- Mac 上の Apple Safari に対して無人テストを実行する場合は、**システム環境設定** の **省エネルギー** ペインで、次の設定を行います。
 - **ディスプレイをオフにするまでの時間** を **しない** に設定します。
 - **ディスプレイがオフのときにコンピュータを自動でスリープさせない** チェック ボックスをオンにします。



注: これらの設定は、**Silk Test Configuration Assistant** を使用すると簡単に行えます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。

- Mac 上の Apple Safari に対して無人テストを実行するには、**スクリーンセーバー** を無効にします。

1. **システム環境設定 > デスクトップとスクリーンセーバ** を開きます。
2. **スクリーンセーバ** タブをクリックします。
3. **開始までの時間** を **開始しない** に設定します。



注: これらの設定は、**Silk Test Configuration Assistant** を使用すると簡単に行えます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。

- Apple Safari 10.1 を使用してテストする場合は、Safari 開発メニューを有効にします。**Safari > 環境設定** を選択して、**詳細** をクリックし、**メニューバーに '開発' メニューを表示** をオンにします。
- Apple Safari 10.1 を使用してテストする場合は、リモート オートメーションを有効にします。Safari 開発メニューから、**リモート オートメーションを許可** をオンにします。
- 初めて Apple Safari 10.1 でテストを実行する際に、パスワード入力が必要になる場合があります。

Apple Safari のテストの準備

Apple Safari 10.1 以降で Web アプリケーションをテストする場合、**Silk Test Configuration Assistant** を使うと、簡単に Apple Safari を設定できます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。または、「[Mac 上の Apple Safari を使用したテストにおける前提条件](#)」に書かれている要件を満たすよう、次の手順を実行します。

1. Apple Safari のリモート オートメーションを有効化するために、**開発** メニューを開き、**リモート オートメーションを許可** をオンにします。
開発 メニューは、デフォルトでは表示されません。このメニューを開くには：
 - a) **Safari** メニューから **環境設定** を選択します。
 - b) **環境設定** ウィンドウで、**詳細** タブを選択します。
 - c) **メニューバーに "開発" メニューを表示** チェック ボックスをオンにします。
 - d) **環境設定** ウィンドウを閉じます。
2. Apple Safari 上で最初にテストを実行すると、ブラウザ ウィンドウが自動テストによってリモート制御されていることを示すダイアログ ボックスが表示されます。**セッションを続ける** をクリックします。

Apple Safari と Selenium WebDriver についての詳細は、『<https://webkit.org/blog/6900/webdriver-support-in-safari-10/>』を参照してください。

Silk Test Information Service を Mac にインストールする



注: Information Service を Mac にインストールするには、Mac の管理者権限が必要です。

Mac 上の Apple Safari や、Mac に接続されている iOS や Android デバイス上のモバイル アプリケーションに対するテストを作成して実行するには、Mac に Silk Test Information Service (Information Service) をインストールしてから、**リモート ロケーション** ダイアログ ボックスを使用して、Silk Test Workbench をインストールした Windows マシンと Mac を接続する必要があります。

Information Service を Mac にインストールするには：

1. Java JDK が Mac 上にインストールされていることを確認します。
2. iOS デバイス上でモバイル アプリケーションをテストする場合は、Xcode が Mac 上にインストールされていることを確認します。
3. Information Service セットアップ ファイル (SilkTestInformationService<バージョン>-<ビルド番号>.pkg) にアクセスします。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードした場合は、Silk Test インストール ディレクトリ (C:\Program Files (x86)\Silk\SilkTest など) の macOS フォルダを開きます。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードしなかった場合は、[Micro Focus SupportLine](#) からセットアップ ファイルをダウンロードできます。
4. SilkTestInformationService<バージョン>-<ビルド番号>.pkg ファイルを Mac にコピーします。
5. SilkTestInformationService<バージョン>-<ビルド番号>.pkg を実行して、Information Service をインストールします。
6. インストール ウィザードの指示に従います。

7. パスワードを尋ねられた場合、現在サインインしている Mac ユーザーのパスワードを入力します。
8. Apple Safari が開き、SafariDriver を信頼するかどうかを尋ねるメッセージボックスが表示されたら、**信頼** をクリックします。


 **注:** リモート接続ではなく、直接 Mac にログインしている場合には、SafariDriver だけをインストールできます。

9. インストールを完了するために、現在の Mac ユーザーをログアウトします。Information Service が正しくインストールされていることを確認するには、Mac にログインします。

- 10 macOS Mojave (10.14) 以降の Mac に Information Service をインストールした場合、Mac にログインした後で Silk Test に対するオートメーションの承認を許可する必要があります。

許可を与える必要がある場合、Silk Test は要求の承認ダイアログを自動的に表示します。


- a) **OK** をクリックして、許可して情報ダイアログを閉じます。
- b) その後に表示されるすべての要求の承認ダイアログに対して、**OK** をクリックします。

 **重要:** これらの Silk Test に対する承認を許可しないと、Google Chrome 上の Web アプリケーションや iOS デバイスやシミュレーター上のモバイル アプリケーションを、この Mac でテストできなくなります。いずれかの承認ダイアログで、誤って **許可しない** をクリックしてしまった場合は、Mac 上でターミナルを開き、次のコマンドを実行してください。

```
sudo tccutil reset AppleEvents
```

その後、Mac を再起動して、再度承認ダイアログが表示されたら **OK** をクリックして許可します。

- 11 画面の右上にある Silk Test アイコンをクリックすると、利用可能なデバイスをブラウザーを確認できます。

 **ヒント:** Silk Test アイコンが表示されない場合は、Mac を再起動してください。

Apple Safari を使用したテストの制限事項

以下に、Mac 上の Apple Safari を使用してテストする際の既知の制限事項を一覧します。

- 次のクラス、インターフェイス、メソッド、プロパティは、Mac 上の Apple Safari を使用した Web アプリケーションのテストでは現時点ではサポートされません：
 - BrowserApplication クラス。
 - ClearCache メソッド
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetHorizontalScrollbar メソッド
 - GetNextCloseWindow メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - GetVerticalScrollbar メソッド
 - IsActive メソッド
 - Minimize メソッド
 - OpenContextMenu メソッド
 - OpenTab メソッド
 - Restore メソッド
 - SelectTab メソッド
 - SetActive メソッド
 - WindowState プロパティ

- BrowserWindow クラス。
 - AcceptAlert メソッド
 - DismissAlert メソッド
 - GetAlertText メソッド
 - IsAlertPresent メソッド
 - MouseMove メソッド
 - PressKeys メソッド
 - PressMouse メソッド
 - ReleaseKeys メソッド
 - ReleaseMouse メソッド
- IMoveable クラス。
 - GetFocus メソッド。
- Silk Test Workbench は、TypeKeys メソッドで **CMD** キーをサポートしません。
- Silk Test Workbench は、Apache Flex のテストをサポートしません。
- Silk Test Workbench は、Apple Safari 上の JavaScript ソースを使った iframe のテストをサポートしません。
- Apple Safari 上で HTTPS を使用したセキュアな Web アプリケーションをテストするには、必要なすべてのサーバー証明書が信頼済みであることを確認してください。
- Silk Test Workbench は、Apple Safari のネイティブ サポートは提供しません。内部 Apple Safari 機能をテストすることはできません。たとえば、テストで、ナビゲーションバーにテキストを追加して、現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。
- Silk Test Workbench は、Apple Safari に対する JavaScript ダイアログ API 関数をサポートしません。回避策として、このような関数が無視されるようにパッチを当てることができます。詳細については、「<https://groups.google.com/forum/#!topic/selenium-developer-activity/qsovJw93g9c>」を参照してください。
- Silk Test Workbench は、Apple Safari のタブをサポートしません。
- ビジュアル テストの再生速度が遅い場合は、**ツール > オプション > 再生 > 結果 > ビジュアル テスト** のオプション **コントロールのキャプチャ** を **いいえ** に設定します。 そうしないと、リッチな Web ページでは再生速度が著しく低下し、通信がタイムアウトすることもあります。
- 複数ウィンドウのアプリケーションをテストするには、Apple Safari のポップアップ ブロックを解除してください。解除する場合は、Apple Safari を起動し、**Safari 環境設定 > セキュリティ > ポップアップウィンドウを開かない** のチェックを外します。
- Silk Test Workbench は、パスワードを保存するダイアログ ボックスのテストをサポートしません。このダイアログ ボックスを表示しないようにする場合は、Apple Safari を起動し、**Safari 環境設定 > 自動入力** に移動して **ユーザー名とパスワード** チェック ボックスのチェックを外します。
- Silk Test Workbench は、Apple Safari では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、Content-Security-Policy HTTP ヘッダーを含んだ Web アプリケーションのテストをサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench はブラウザーの前に戻る操作をサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は TypeKeys メソッドでの Ctrl キーの使用をサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は Frame および IFrame 内では DOM 操作のみをサポートします。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は Frame および IFrame での移動操作をサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は記録時の直接スクロールをサポートしません。回避策として、executeJavaScript メソッドを使用できます。

複数の Apple Safari テストの同時実行

Apple Safari 上でテストを実行するには、Silk Test がインストールされた Windows マシンに接続された Mac が必要です。複数の Apple Safari テストを Apple Safari 上で実行する場合、これらのテストは同じ Mac 上で同時に実行されることになります。



注: Mac 上の Apple Safari に対して実行されるテストそれぞれが、Apple Safari のインスタンスを個々に開きます。数多くの Apple Safari のインスタンスが同時に実行すると、Mac のパフォーマンスが低下する可能性があります。

Silk Test Information Service を Mac からアンインストールする

Mac 上の Apple Safari に対するテストを実行する必要がなくなった場合など、次の手順で Silk Test Information Service を Mac からアンインストールすることができます。

1. `uninstallInfoService.sh` のような新しいシェル ファイルを作成します。
2. 新しいファイルに以下のコードを入力します。

```
#!/bin/sh

if launchctl list | grep com.borland.infoservice ; then
  launchctl unload /Library/LaunchAgents/com.borland.infoservice.plist
  echo "unloading Launch Daemon"
fi

if [ -d "/Applications/Silk" ]
then
  sudo rm -rf /Applications/Silk
fi

if [ -f "/Library/LaunchAgents/com.borland.infoservice.plist" ]
then
  sudo rm /Library/LaunchAgents/com.borland.infoservice.plist
fi

if [ -f "/usr/local/bin/ideviceinstaller" ]
then
  sudo rm /usr/local/bin/ideviceinstaller
fi

exit 0
```

3. コマンドラインで `chmod +x uninstallInfoService.sh` を実行し、シェル ファイルの実行可能にします。
4. コマンドラインからシェル ファイルを実行します。

Google Chrome を使用したテスト

このセクションでは、Google Chrome を使用してテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Silk Test Workbench は、Google Chrome 50 以降で操作を記録したり、テストを再生することをサポートします。また、Google Chrome 50 より前のバージョンでテストを再生したり、ロケーターを記録することをサポートします。

- Google Chrome 50 以降でテストを開始するときに、Google Chrome のインスタンスが実行していない場合、Silk Test Workbench は Google Chrome の新しいインスタンスを開始します。新しいブラウザーは、アドオン無しのキャッシュを空にした状態の一時プロファイルを使用します。
- Google Chrome 50 以降のインスタンス上でのテストを開始するときに、すでに実行中であれば、Silk Test Workbench はそのインスタンスが最初に開始されたときに使用されたコマンド ライン引数と同

じ引数で Google Chrome を再起動します。この再起動は、Silk Test Workbench オートメーション サポートを有効化するために必要です。

- Google Chrome 50 以降でテストするとき、Google Chrome インスタンスは、Open Agent のシャットダウン時または、Google Chrome 外のほかのアプリケーションのテストを開始するときに閉じられます。



ヒント: Google Chrome 50 以降を使用して既存のテスト スクリプトを実行する場合は、基本状態を使用して、URL へ移動するコマンドをテスト スクリプトに追加することを、Micro Focus はお勧めします。

例 1

Google Chrome 50 以降の実行中のインスタンスが、コマンド「C:/Program Files (x86)/Google/Chrome/Application/chrome.exe www.borland.com」で最初に起動されていた場合、Google Chrome は再起動後に *www.borland.com* を開きます。

例 2

Google Chrome 50 以降の実行中のインスタンスが、コマンド「C:/Program Files (x86)/Google/Chrome/Application/chrome.exe」で最初に起動されていた場合、Google Chrome は再起動後に *about:blank* を開きます。

Google Chrome を使用したテスト再生の前提条件


コマンドラインパラメータ

Google Chrome を使用してテストの再生やロケーターの記録を行うと、次のコマンドで Google Chrome が起動されます。

```
%LOCALAPPDATA%\%Google%\Chrome%\Application%\chrome.exe
--enable-logging
--log-level=1
--disable-web-security
--disable-hang-monitor
--disable-prompt-on-repost
--dom-automation
--full-memory-crash-report
--no-default-browser-check
--no-first-run
--homepage=about:blank
--disable-web-resources
--disable-preconnect
--enable-logging
--log-level=1
--safebrowsing-disable-auto-update
--test-type=ui
--noerrdialogs
--metrics-recording-only
--allow-file-access-from-files
--disable-tab-closeable-state-watcher
--allow-file-access
--disable-sync
--testing-channel=NamedTestingInterface:st_42
```

ウィザードを使用してアプリケーションに追加する場合は、これらのコマンドラインパラメータは、基本状態に自動的に追加されます。テストを開始したときに、適切なコマンドラインパラメータなしで Google Chrome のインスタンスがすでに実行されている場合、Silk Test Workbench は Google Chrome

を終了して、コマンドラインパラメータを使用してブラウザを再起動しようとしています。ブラウザを再起動できない場合は、エラーメッセージが表示されます。

 **注:** クロスドメインのドキュメントを記録または再生する場合は、コマンドラインパラメータ `disable-web-security` が必要です。

 **注:** ローカルファイルシステムに保存された Web アプリケーションをテストするには、Google Chrome で `chrome://extensions` に移動し、**Silk Test Chrome Extension** の **ファイルの URL へのアクセスを許可する** チェックボックスをオンにします。


Google Chrome 拡張のテスト

Google Chrome 拡張 (アドオン) を Silk Test Workbench を使ってテストするには、以下の 2 つの手段のいずれかを使用できます。

Google Chrome の起動時に .crx ファイルとして拡張をインストールする

.crx ファイルとしてインストールされた Google Chrome 拡張をテストするには、基本状態に次のコマンドラインを追加します。

```
chrome.exe --load-extension=C:/myExtension/myExtension.crx
```

 **注:** Google Chrome には単一の拡張のみを .crx ファイルとしてインストールできます。Google Chrome に複数の拡張をインストールするには、複数の .crx ファイルをカンマ区切りで指定します。例：

```
chrome.exe --load-extension=C:/myExtension/myExtension.crx,C:/myExtension2/myExtension2.crx
```

ブラウザのコマンドライン引数指定方法の詳細については、「[基本状態の変更](#)」を参照してください。

プロファイルに拡張を追加する

Google Chrome ユーザー データ ディレクトリに拡張を追加し、そのプロファイルをテストに使用します。詳細については、「[ユーザー データ ディレクトリを使用した Google Chrome のテスト](#)」を参照してください。

ユーザー データ ディレクトリを使用した Google Chrome のテスト


ホームページ、使用するツールバーの設定や保存したパスワード、ブックマークなど、Google Chrome で行ったすべての変更は、ユーザー データ ディレクトリと呼ばれる特別なフォルダに格納されます。

Silk Test Workbench を使用して、テスト対象アプリケーションの基本状態にユーザー データ ディレクトリへのパスを指定することによって、Google Chrome ユーザー データ ディレクトリをテストできます。次のコマンドラインには、プロファイルへのパスが含まれています。

```
chrome.exe "--user-data-dir=C:/Users/MyUser/AppData/Local/Google/Chrome/User Data"
```

サンプル Web アプリケーションのプロファイル ディレクトリを設定するには、次のコードを使用できません。

```
' VB code
Dim baseState = New BrowserBaseState(BrowserType.GoogleChrome, "demo.borland.com/InsuranceWebExtJS")
Dim myProfileDir = "--user-data-dir=D:¥¥temp¥¥SilkTest --profile-directory=Profile1"
baseState.CommandLineArguments = myProfileDir
baseState.Execute()
```

 **注:** Google Chrome が Silk Test Workbench によって起動されるときには、空のユーザー データ ディレクトリが使用されます。これにより、クリーンな状態でテストが開始されることとなります。

Google Chrome を使用したテストの制限事項

以下のリストに、ローカル Windows マシン上で Google Chrome を使用した場合の、テストの再生とロケータの記録の既知の制限事項をリストします。

- Silk Test は、Google Chrome を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Google Chrome ではサポートされていません。
- Silk Test Workbench は、HTTP 基本認証ダイアログのテストの記録をサポートしません。
- Silk Test は、Google Chrome のネイティブ サポートは提供しません。内部 Google Chrome 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーションバーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。
- Silk Test Workbench は Google Chrome を使用して Web アプリケーションをテストする際の列挙子をサポートしていません。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test は、Google Chrome メニューを使用して Google Chrome の印刷ダイアログ ボックスが開かれたことは認識しません。Google Chrome でダイアログ ボックスを開く動作を追加してテストするには、TypeKeys メソッドを使用して **Ctrl+Shift+P** を送信する必要があります。Internet Explorer はこのショートカットを認識しません。したがって、最初に Internet Explorer でテストを記録してから、手動で **Ctrl+Shift+P** を押す操作をテストに追加する必要があります。
- Google Chrome の複数ウィンドウの同時テストは、最初の Google Chrome ウィンドウから AUT 自身が追加のウィンドウを開いた場合にのみサポートされます。追加の Google Chrome ウィンドウを手動で開いた場合、Silk Test Workbench はこれらの Google Chrome ウィンドウ上の要素を解決できません。たとえば、記録中に AUT のリンクやボタンをクリックして開かれた Google Chrome ウィンドウ上の要素は Silk Test Workbench は解決できますが、記録中に **CTRL+N** を押して開かれた Google Chrome ウィンドウ上の要素は Silk Test Workbench は解決できません。
- ビジュアル テストの再生速度が遅い場合は、**ツール > オプション > 再生 > 結果 > ビジュアル テスト** のオプション **コントロールのキャプチャ** を **いいえ** に設定します。そうしないと、リッチな Web ページでは再生速度が著しく低下し、通信がタイムアウトすることもあります。
- Google Chrome 49 以前を使用する場合、Internet Explorer を使用してテストを再生する場合、executeJavaScript をテストするために次のコードを使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("function foo() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("foo();");
```

Google Chrome 上でテストを再生する場合、スクリプトはクロージャで実行され、グローバル コンテキスト (window) では実行されません。すべては関数内で実行されます。上記のサンプル コードの最初の ExecuteJavaScript 呼び出しは、Google Chrome では機能しません。これは、関数 foo が ExecuteJavaScript 呼び出しが存続する間だけ有効であるためです。

Google Chrome 上で同じテストを再生する場合は、次のような関数式を使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo = function() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo();");
```

前のサンプル コードは Silk4J では機能します。他の Silk Test クライアントに対するコードも同様です。詳細については、Silk Test クライアントのヘルプにある ExecuteJavaScript メソッドのドキュメントを参照してください。

- Google Chrome のユーザー データ ディレクトリをグループ ポリシーを使って設定していると、Google Chrome 上での並列テストは機能しません。この問題を回避するには、レジストリー キー HKEY_LOCAL_MACHINE¥Software¥Policies¥Google¥Chrome¥UserDataDir または HKEY_CURRENT_USER¥Software¥Policies¥Google¥Chrome¥ を削除するよう、管理者に依頼してください。

macOS 上の Google Chrome を使用したテストの制限事項

以下のリストに、macOS 上で Google Chrome を使用した場合の、テストの再生とロケーターの記録の既知の制限事項をリストします。

- Silk Test Workbench は、TypeKeys メソッドで **CMD** キーをサポートしません。
- Silk Test は、Google Chrome を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Google Chrome ではサポートされていません。
- Silk Test Workbench は、HTTP 基本認証ダイアログのテストの記録をサポートしません。
- Silk Test は、Google Chrome のネイティブ サポートは提供しません。内部 Google Chrome 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーションバーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- 複数の Google Chrome ウィンドウ上での同時テストは、macOS 上ではサポートされません。
- 既に開いている Google Chrome ウィンドウへのアタッチは、macOS 上ではサポートされません。
- Google Chrome 49 以前を使用する場合、Internet Explorer を使用してテストを再生する場合、executeJavaScript をテストするために次のコードを使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("function foo() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("foo();");
```

Google Chrome 上でテストを再生する場合、スクリプトはクロージャで実行され、グローバル コンテキスト (window) では実行されません。すべては関数内で実行されます。上記のサンプル コードの最初の ExecuteJavaScript 呼び出しは、Google Chrome では機能しません。これは、関数 foo が ExecuteJavaScript 呼び出しが存続する間だけ有効であるためです。

Google Chrome 上で同じテストを再生する場合は、次のような関数式を使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo = function() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo();");
```

前のサンプル コードは Silk4J では機能します。他の Silk Test クライアントに対するコードも同様です。詳細については、Silk Test クライアントのヘルプにある ExecuteJavaScript メソッドのドキュメントを参照してください。


Mozilla Firefox を使用したテスト

このセクションでは、Mozilla Firefox を使用してテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Silk Test Workbench は、Mozilla Firefox 上での操作の記録とテストの再生をサポートします。

- Mozilla Firefox でテストを開始するときに、Mozilla Firefox のインスタンスが実行していない場合、Silk Test Workbench は Mozilla Firefox の新しいインスタンスを開始します。新しいブラウザーは、アドオン無しのキャッシュを空にした状態の一時プロファイルを使用します。
- Mozilla Firefox のインスタンス上でのテストを開始するときに、すでに実行中であれば、Silk Test Workbench はそのインスタンスが最初に開始されたときに使用されたコマンド ライン引数と同じ引数で Mozilla Firefox を再起動します。この再起動は、Silk Test Workbench オートメーション サポートを有効化するために必要です。

- この Mozilla Firefox のインスタンスは、Open Agent のシャットダウン時または、Mozilla Firefox 外のほかのアプリケーションのテストを開始するときに閉じられます。

 **ヒント:** Mozilla Firefox を使用して既存のテスト スクリプトを実行する場合は、基本状態を使用して、URL へ移動するコマンドをテスト スクリプトに追加することを、Micro Focus はお勧めします。

Mozilla Firefox 52 以降で Silk Test Workbench を使用して記録する場合、Mozilla Firefox は外部リンクを新しいウィンドウではなく新しいタブで開きます。外部リンクを新しいウィンドウで開く場合は、**詳細設定** オプションの **Firefox のシングル ウィンドウ モードを使用する** オプションを無効にします。

例 1

Mozilla Firefox の実行中のインスタンスが、コマンド「C:/program files/Mozilla/firefox.exe www.borland.com」で最初に起動されていた場合、Mozilla Firefox は再起動後に *www.borland.com* を開きます。

例 2

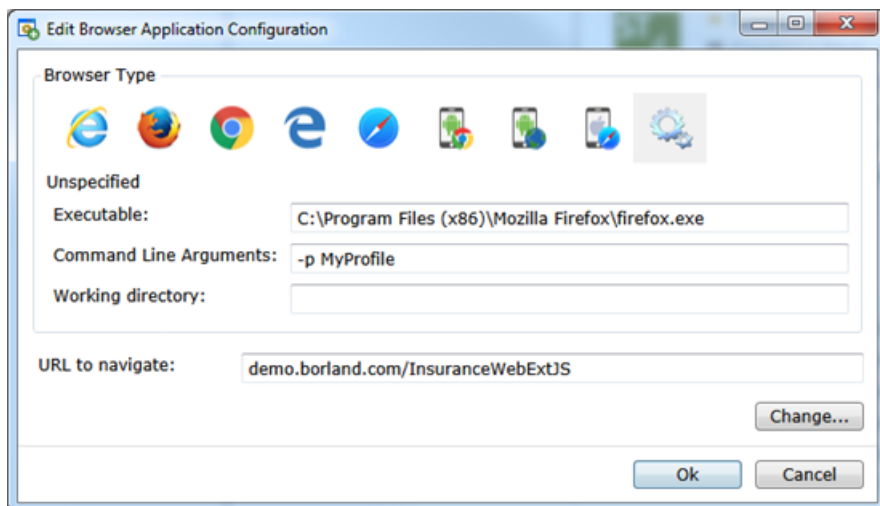
Mozilla Firefox の実行中のインスタンスが、コマンド「C:/program files/Mozilla/firefox.exe」で最初に起動されていた場合、Mozilla Firefox は再起動後に *about:blank* を開きます。

プロファイルを使用した Mozilla Firefox のテスト

ホームページ、使用するツールバーの設定や保存したパスワード、ブックマークなど、Mozilla Firefox で行ったすべての変更は、プロファイルと呼ばれる特別なフォルダに格納されます。

Mozilla Firefox プロファイル进行测试するには、プロファイルの名前、またはプロファイルへのパスのどちらかをカスタム ブラウザー コマンド ライン引数として **アプリケーション構成の編集** ダイアログ ボックスで指定します。

1. Mozilla Firefox プロファイル进行测试するプロジェクトを選択します。
2. メニューから **アプリケーション構成の編集** をクリックします。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
3. **編集** をクリックして、既存の **ブラウザー : Firefox** アプリケーション構成を編集します。
4. ブラウザーの種類から **カスタム** を選択します。
5. **実行可能ファイル** フィールドに、Mozilla Firefox 実行可能ファイルへのパスを入力します。たとえば、C:\Program Files (x86)\Mozilla Firefox\firefox.exe など。
6. **コマンド ライン引数** フィールドに、Firefox プロファイル名または Firefox プロファイルへのパスを入力します。たとえば、-p myProfile または -profile C:/Temp など。




プロファイルの名前の使用 次のコマンド ライン引数では、プロファイル名を指定しています。


```
-p myProfile
```

プロファイル マネージャ で名前をつけてプロファイルを設定できます。**プロファイル マネージャ** を起動するには、コマンド ウィンドウで `firefox.exe -P` を入力します。

プロファイルへのパスの使用 次のコマンド ライン引数では、プロファイルへのパスを指定しています。

```
-profile C:/<path to profile folder>
```

 **注:** Silk Test Workbench が Mozilla Firefox を起動する場合、空のプロファイルが使用されます。これにより、クリーンな状態でテストが開始されることとなります。ユーザーが手動で Mozilla Firefox を起動した場合は、デフォルトのプロファイルが使用されます。

 **注:** プロファイルはテストマシンにデプロイする必要があり、メモリ消費量も大きいいため、プロファイルのテスト時に問題が発生する可能性があります。いくつかのブラウザーの設定を変更するだけであれば、プロファイルの代わりにケイパビリティを使用できます。詳細については、「[WebDriver ベースのブラウザーのケイパビリティの設定](#)」を参照してください。

Mozilla Firefox 拡張機能のテスト

Silk Test Workbench を使って Mozilla Firefox 拡張機能（アドオン）をテストするには、Mozilla Firefox プロファイルに拡張機能を追加し、そのプロファイルをテストに使用します。詳細については、「[プロファイルを使用した Mozilla Firefox のテスト](#)」を参照してください。

Mozilla Firefox を使用したテストの制限事項

Silk Test Workbench を使用して Mozilla Firefox 上の Web アプリケーションをテストする際の既知の制限事項を以下に示します。

- Silk Test Workbench は Mozilla Firefox 47、48、49、50、51 をサポートしません。ただし、Silk Test Workbench は Mozilla Firefox 47.0.1 および 47.0.2 をサポートします。
- Mozilla Firefox の複数ウィンドウの同時テストは、最初の Mozilla Firefox ウィンドウから AUT 自身が追加のウィンドウを開いた場合にのみサポートされます。追加の Mozilla Firefox ウィンドウを手動で開いた場合、Silk Test Workbench はこれらの Mozilla Firefox ウィンドウ上の要素を解決できません。たとえば、記録中に AUT のリンクやボタンをクリックして開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できますが、記録中に **CTRL+N** を押して開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できません。
- Silk Test Workbench は `window.showmodalDialog` コマンドを使って表示されるウィンドウであるモーダルブラウザー ウィンドウのテストをサポートしません。これらのモーダルブラウザー ウィンドウは、公式に廃止されており、Google Chrome 37 以降では無効化されています。さらに、Mozilla Firefox の今後のバージョンではサポートされる計画はありません。低レベルな操作を使ってこの問題を回避できます。たとえば、オブジェクトの座標を使用したネイティブなクリックやテキスト フィールドを `typeKeys` を使って入力する方法が利用できます。
- Silk Test Workbench は、Mozilla Firefox を使用した Silverlight のテストをサポートしません。
- Silk Test Workbench は Mozilla Firefox の **バージョン情報** ダイアログなどの一部のブラウザー ダイアログのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox を使用した `about:*` ページのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox の **印刷** ボタンのクリックの記録をサポートしません。再生時にこのボタンをクリックする場合は、座標によるデスクトップのクリックをテスト スクリプトに手動で追加することにより行えます。例：

```
'VB .NET code  
_desktop.Click(printButton.GetRect(true).Center)
```

- Silk Test は、Mozilla Firefox のネイティブ サポートは提供しません。内部 Mozilla Firefox 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーション バーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して

Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。

- Silk Test Workbench は Mozilla Firefox の JavaScript 警告ボックスのロケータの記録をサポートしません。Mozilla Firefox バージョン 58 以前を使用する場合、Javascript 警告ダイアログを処理するために、次のメソッドを使用できます。
 - AcceptAlert
 - DismissAlert
 - GetAlertText
 - IsAlertPresent



注: Mozilla Firefox 59 以降を使用する場合、これらのメソッドは使用できません。

- Silk Test Workbench は、Mozilla Firefox 52 以降では Java アプレットをサポートしません。Silk Test Workbench は、Mozilla Firefox 47.0.1 以前では Java アプレットをサポートします。ただし、次のような制限事項があります。
 - Silk Test Workbench は ロケーター //AppletContainer をサポートしません。
 - アプレットがモーダル ダイアログを開くとき、//BrowserApplication//BrowserWindow/JDialog[@caption='Information']//JButton[@caption='OK'] のようなロケーターが機能しない場合があります。代わりに、//JDialog[@caption='Information']//JButton[@caption='OK'] のようなロケーターは使用できます。
- Silk Test Workbench は、Mozilla Firefox では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test は、Mozilla Firefox を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Mozilla Firefox ではサポートされていません。
- Mozilla Firefox 52 以降を使用する場合、次のメソッドはサポートされません。
 - PressKeys
 - ReleaseKeys
- Mozilla Firefox 52 以前を使用する場合、BrowserWindow クラスの SetViewportSize メソッドはサポートされません。
- Mozilla Firefox 52 以降を使用する場合、次のネイティブ再生はサポートされません。
 - ダブルクリック。
 - マウス ボタンの右と中央のクリック。
- Mozilla Firefox 52 以降を使用する場合、DomClick メソッドは、警告を開くコントロールではサポートされません。
- Mozilla Firefox 55 を使用する場合、ファイルのアップロードは機能しません。詳細については、『[File upload no longer works with geckodriver 0.18.0 and Firefox 55](#)』を参照してください。

macOS 上の Mozilla Firefox を使用したテストの制限事項

Mozilla Firefox を使用して macOS 上の Web アプリケーションをテストする際の既知の制限事項を以下に示します。


- Silk Test Workbench は、macOS 上の Mozilla Firefox 54 以降を使用してテストされました。
- Mozilla Firefox の複数ウィンドウの同時テストは、最初の Mozilla Firefox ウィンドウから AUT 自身が追加のウィンドウを開いた場合にのみサポートされます。追加の Mozilla Firefox ウィンドウを手動で開いた場合、Silk Test Workbench はこれらの Mozilla Firefox ウィンドウ上の要素を解決できません。たとえば、記録中に AUT のリンクやボタンをクリックして開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できますが、記録中に **CTRL+N** を押して開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できません。
- Silk Test Workbench は window.showmodalDialog コマンドを使って表示されるウィンドウであるモーダル ブラウザー ウィンドウのテストをサポートしません。これらのモーダル ブラウザー ウィン

ドウは、公式に廃止されており、Google Chrome 37 以降では無効化されています。さらに、Mozilla Firefox の今後のバージョンではサポートされる計画はありません。低レベルな操作を使ってこの問題を回避できます。たとえば、オブジェクトの座標を使用したネイティブなクリックやテキスト フィールドを typeKeys を使って入力する方法が利用できます。

- Silk Test Workbench は、Mozilla Firefox を使用した Silverlight のテストをサポートしません。
- Silk Test Workbench は Mozilla Firefox の **バージョン情報** ダイアログなどの一部のブラウザー ダイアログのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox を使用した about:* ページのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox の **印刷** ボタンのクリックの記録をサポートしません。再生時にこのボタンをクリックする場合は、座標によるデスクトップのクリックをテスト スクリプトに手動で追加することにより行えます。例：

```
'VB .NET code
_desktop.Click(printButton.GetRect(true).Center)
```

- Silk Test は、Mozilla Firefox のネイティブ サポートは提供しません。内部 Mozilla Firefox 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーション バーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。
- Silk Test Workbench は Mozilla Firefox の JavaScript 警告ボックスのロケータの記録をサポートしません。Mozilla Firefox バージョン 58 以前を使用する場合、Javascript 警告ダイアログを処理するために、次のメソッドを使用できます。
 - AcceptAlert
 - DismissAlert
 - GetAlertText
 - IsAlertPresent

 **注:** Mozilla Firefox 59 以降を使用する場合、これらのメソッドは使用できません。

- Silk Test Workbench は、macOS 上の Mozilla Firefox では Java アプレットをサポートしません。
- Silk Test Workbench は、Mozilla Firefox では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test は、Mozilla Firefox を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Mozilla Firefox ではサポートされていません。
- 次のメソッドはサポートされません。
 - PressKeys
 - ReleaseKeys
- 次のネイティブ再生はサポートされません。
 - ダブルクリック。
 - マウス ボタンの右と中央のクリック。
- DomClick メソッドは、警告を開くコントロールではサポートされません。
- Mozilla Firefox 55 を使用する場合、ファイルのアップロードは機能しません。詳細については、『[File upload no longer works with geckodriver 0.18.0 and Firefox 55](#)』を参照してください。

Microsoft Edge を使用したテスト

このセクションでは、Microsoft Edge を使用してテストすることによって、クロスブラウザー テスト セットを拡張する方法について説明します。

Microsoft Edge を使用したテストの制限事項

以下に、Microsoft Edge を使用してテストする際の既知の制限事項を一覧します。

- 次のクラス、インターフェイス、メソッド、プロパティは、Microsoft Edge 上の Web アプリケーションのテストでは現時点ではサポートされません：
 - BrowserApplication クラス。
 - ClearCache メソッド
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetHorizontalScrollbar メソッド
 - GetNextCloseWindow メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - GetVerticalScrollbar メソッド
 - IsActive メソッド
 - Minimize メソッド
 - OpenContextMenu メソッド
 - OpenTab メソッド
 - Restore メソッド
 - SelectTab メソッド
 - SetActive メソッド
 - WindowState メソッド
 - BrowserWindow クラスの次のメソッドは、Microsoft Edge のビルド 38.14393 (Microsoft Windows 10 Anniversary Update のバージョンの Microsoft Edge) より前のバージョンではサポートされません。
 - PressKeys メソッド
 - ReleaseKeys メソッド
- Silk Test Workbench は、Microsoft Edge に対する操作を記録するとき、ブラウザを自動的に最前面に表示しません。
- Microsoft Edge を使用してテストすると、BrowserApplication の矩形領域は絶対値ではありません。
- Silk Test Workbench は、Apache Flex のテストをサポートしません。
- Silk Test Workbench は、Microsoft Edge のネイティブ サポートは提供しません。内部 Microsoft Edge 機能をテストすることはできません。たとえば、テストで、ナビゲーションバーにテキストを追加して、現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。
- Silk Test Workbench は、Microsoft Edge の警告ダイアログなどのダイアログ ボックスをサポートしません。
- イメージ クリックは、Microsoft Edge Threshold 2 (ビルド 25.10586) 以降に対してのみサポートされます。Microsoft Edge の以前のバージョン上で Web アプリケーションをテストする場合は、イメージ検証のみ使用できます。
- Silk Test Workbench は、Microsoft Edge のタブをサポートしません。タブはウィンドウとして解決されます。
- ビジュアル テストの再生速度が遅い場合は、**ツール > オプション > 再生 > 結果 > ビジュアル テスト** のオプション **コントロールのキャプチャ** を **いいえ** に設定します。 そうしないと、リッチな Web ページでは再生速度が著しく低下し、通信がタイムアウトすることもあります。
- Microsoft Edge 上の Web アプリケーションをテストするとき、Silk Test Workbench は http-equiv 属性などのメタタグを検索できません。たとえば、Silk Test Workbench は次のメタタグを検索できません：

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

- Microsoft Edge を使用する場合、Silk Test Workbench は DOM 要素の `currentStyle` 属性の直接読み取りをサポートしません。DomElement クラスの `GetCssStyle` メソッドを使用して、指定したスタイル名の算出 CSS スタイルを取得できます。
- Microsoft Edge 上の Web アプリケーションとのやり取りを開始するとき、Silk Test Workbench は Microsoft Edge のすべての開いているインスタンスを閉じ、新しくブラウザを開始します。新しいブラウザは、アドオン無しのキャッシュを空にした状態の一時プロファイルを使用します。この Microsoft Edge のインスタンスは、Open Agent のシャットダウン時または、Microsoft Edge 外のほかのアプリケーションのテストを開始するときに閉じられます。
- Microsoft Edge を使用する場合、Silk Test Workbench は操作やロケータの記録中に `textContent` 属性を認識しません。しかし、オブジェクト マップで `textContent` 属性を使用して、Microsoft Edge 上でテストを再生するときに使用できます。
- Silk Test Workbench は、Microsoft Edge では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、IMoveable クラスの `GetFocus` メソッドをサポートしません。
- Silk Test Workbench は、Content-Security-Policy HTTP ヘッダーを含んだ Web アプリケーションのテストをサポートしません。

レスポンス Web デザインのテスト

レスポンス Web デザインに基づいて構築されたデスクトップ Web アプリケーションは、アプリケーションを表示する画面や Web ブラウザーのサイズに応じて、その外観が変化します。このようなテストでは、再生するウィンドウのサイズを適切に選択できないと、その安定性に多大な影響を与える可能性があります。

Silk Test Workbench では、次のタイミングでブラウザ ウィンドウの正確なサイズを指定することができます。

- Web アプリケーションに対して新しいプロジェクトを作成する時点
- 新しいビジュアル テストや .NET スクリプトを作成する時点
- 新しいテストを記録する時点
- 既存のテストに操作を記録する時点
- Web アプリケーションに対するテストを再生する時点

次の設定を使用して、ブラウザ ウィンドウのサイズを指定することができます。

- **ブラウザ サイズ** リストには、あらかじめ定義されたブラウザ ウィンドウのサイズとカスタマイズしたサイズの両方が表示され、ここからテストで使用するサイズを選択できます。
- **向き** リストでは、ブラウザ ウィンドウの向きが横向きか縦向きかを選択できます。
- **ブラウザ サイズの編集** をクリックすると、**ブラウザ サイズ** リストにカスタム ブラウザー サイズを追加したり、リストに表示するブラウザ サイズを選択することができます。
 - 新しいカスタム ブラウザー サイズをリストに追加するには、**ブラウザ サイズの追加** をクリックします。
 - リストにサイズを表示しないようにするには、対応するチェック ボックスをオフにします。
 - 特定の Web アプリケーションのビジュアル ブレークポイントを **ブラウザ サイズ** リストに追加するには、**ビジュアルブレークポイントの検出** をクリックします。

コマンド ラインや Silk Central から Web アプリケーションに対するテストを再生する場合は、`silktest.browserViewportSize` 環境変数を設定してブラウザ ビューポートのサイズを指定できます。**ブラウザ サイズ** リストで定義したブラウザの名前や特定のサイズを指定することができます。

Imports SilkTest.ntf.Mobile をテスト スクリプトで定義すると、テストスクリプトで直接ブラウザのサイズを変更できます：

```
' VB .NET code
With .BrowserWindow("BrowserWindow")
    .SetViewportName("Google Pixel 2")
```

```
.SetViewportSize(800, 300)
End With
```

例：自動再生するブラウザー サイズに名前を使用して指定する場合

ブラウザー サイズ リストの SVGA エントリを使用して、ブラウザー サイズを SVGA (800, 600) に設定するサンプルコードを以下に示します。

```
SET silktest.browserViewportSize=name=SVGA;orientation=landscape
STW.exe -script MyScript
```

次のコマンドを使用して、特定のブラウザーとビューポート サイズを指定することもできます。

```
STW -dsn silktest -browser InternetExplorer -script MyScript -
viewportname SVGA
```

例：自動再生するブラウザー サイズに幅と高さを使用して指定する場合

`width` と `height` パラメータを使用して、ブラウザー サイズを SVGA (800, 600) に設定するサンプルコードを以下に示します。

```
SET
silktest.browserViewportSize=width=800;height=600;orientation=landscape
STW.exe -script MyScript
```

次のコマンドを使用して、特定のブラウザーとビューポート サイズを指定することもできます。

```
STW -dsn silktest -browser InternetExplorer -script MyScript -
viewportwidth 800 -vieportheight 600
```

ビジュアルブレイクポイントの検出

レスポンシブ Web アプリケーションのビジュアルブレイクポイントを検出する前に、Mozilla Firefox 56 以降または Google Chrome 60 以降を Silk Test Workbench を実行しているマシン上にインストールしてください。

レスポンシブ Web デザイン技術を使用して実装する多くの Web アプリケーションは、表示されるブラウザーやデバイスのサイズに応じて、そのレイアウトを変更します。レイアウトの変更が発生する特定の解像度を、ビジュアルブレイクポイントと言います。

Silk Test Workbench では、ビジュアルブレイクポイントを検出し、それを使用して特定のサイズに記録ウィンドウのサイズを変更することで、このようなアプリケーションのテストをサポートします。

Silk Test Workbench では、次のタイミングでブラウザー ウィンドウの正確なサイズを指定することができます。

- Web アプリケーションに対して新しいプロジェクトを作成する時点
- 新しいビジュアルテストや .NET スクリプトを作成する時点
- 新しいテストを記録する時点
- 既存のテストに操作を記録する時点
- Web アプリケーションに対するテストを再生する時点

Web アプリケーションのビジュアルブレイクポイントを検出し、対応する解像度を **ブラウザー サイズ** リストに表示させるには、次の手順を実行します。

1. **ブラウザー サイズの編集** をクリックします。 **ブラウザー サイズの編集** ダイアログ ボックスが表示されます。

2. **ビジュアルブレイクポイントの検出** をクリックします。Web アプリケーションの URL がアプリケーション構成や基本状態で指定されていない場合は、**ビジュアルブレイクポイント検出 URL** ダイアログが表示されます。
3. Web アプリケーションの URL がアプリケーション構成や基本状態で指定されていない場合は、**ビジュアルブレイクポイント検出 URL** ダイアログが表示されます。URL をテキスト フィールドに入力し、**OK** をクリックします。Silk Test Workbench は Web アプリケーションのすべてのビジュアルブレイクポイントを検出し、**ブラウザー サイズ** リストに追加します。
4. **OK** をクリックして **ブラウザー サイズの編集** ダイアログを閉じます。

以上により、テストするブラウザー ウィンドウまたはモバイル デバイスのサイズに合わせて、ビジュアルブレイクポイントを選択できるようになります。

追加のブラウザーのバージョンでのテスト

このトピックでは、Silk Test Workbench のバージョンに自動的に含まれない WebDriver ベースのブラウザーの追加のバージョンでテストを行う方法について説明します。



注: Silk Test Workbench は、ブラウザーの最新のバージョンを自動的にサポートします。ただし、ブラウザーのベンダーが Silk Test Workbench のアップデートを必要とする変更を行った場合、このトピックに記述した手順を行っても、新しいブラウザーのバージョンでテストを行うことができない場合もあります。

このトピックで説明されている機能は、次のブラウザーに対してサポートされます。

- Google Chrome
- Microsoft Edge
- Mozilla Firefox

1. テストするブラウザーのバージョンに対して適切なドライバーをダウンロードします。
 - Google Chrome の場合、追加の ChromeDriver のバージョンは [Downloads - ChromeDriver](#) からダウンロードできます。
 - Mozilla Firefox の場合、追加の geckodriver のバージョンは [Releases - mozilla/geckodriver](#) からダウンロードできます。
 - Microsoft Edge の場合、追加の Microsoft WebDriver のバージョンは [WebDriver - Microsoft Edge](#) からダウンロードできます。
2. Silk Test Workbench インストール フォルダーの、¥ng¥WebDrivers¥ フォルダーに移動します。
3. 使用する新しいブラウザーのバージョンのオペレーティング システムに対応するフォルダーを開きます。
 - Microsoft Windows の場合、Windows フォルダーを開きます。
 - macOS の場合、osx64 フォルダーを開きます。
4. ブラウザーに対して適切なフォルダーを開きます。
 - Google Chrome の場合、Chrome を開きます。
 - Mozilla Firefox の場合、Gecko を開きます。
 - Microsoft Edge の場合、Edge を開きます。
5. 新しいドライバーのバージョン用に新しいフォルダを作成します。
たとえば、ドライバーが ChromeDriver 2.26 であれば、2.26 という新しいフォルダーを作成します。
6. ダウンロードしたドライバーを新しいフォルダーで展開します。
7. Silk Test Workbench インストール フォルダーで、¥ng¥WebDrivers¥Common¥Config¥ フォルダーに移動します。
8. ブラウザーに対して適切なフォルダーを開きます。
 - Google Chrome の場合、Chrome を開きます。

- Mozilla Firefox の場合、Gecko を開きます。
 - Microsoft Edge の場合、Edge を開きます。
9. プロパティ ファイルをテキスト エディターで開きます。
たとえば、Google Chrome の場合、Chrome.properties を開きます。
- 10 新しいブラウザのバージョンと新しいドライバーのバージョンを次のように追加します。

```
<browser version>=<driver version>
```

たとえば、Google Chrome 53 でテストする場合、ChromeDriver 2.26 が必要であるため、Chrome.properties ファイルに次の行を追加する必要があります。

```
53=2.26
```

- 11 プロパティ ファイルを保存します。

クロス ブラウザ テスト:よくある質問

このセクションでは、さまざまなブラウザ上で Web アプリケーションをテストするときに発生することがある質問を示します。

再生中にダイアログが認識されない

スクリプトを記録するときに、Silk Test Workbench はいくつかのウィンドウを Dialog として認識します。スクリプトをクロス ブラウザ スクリプトとして使用する場合は、ブラウザによっては Dialog が認識されないため、Dialog を Window に置き換える必要があります。

たとえば、スクリプトに以下の行があるとします。

```
/BrowserApplication//Dialog//PushButton[@caption='OK']
```

クロス ブラウザ テストを可能にするには、次のように行を書き換えます。

```
/BrowserApplication//Window//PushButton[@caption='OK']
```

DomClick(x, y) が Click(x, y) のように動作しない

アプリケーションで onclick イベントを使用しており、座標を必要とする場合、DomClick メソッドは動作しません。代わりに、Click を使用します。

FileInputField.DomClick() でダイアログが開かない

代わりに、Click を使用します。

テストの開始時にブラウザ ウィンドウを最大化する方法

テストの開始時など、テスト スクリプトからブラウザを最大化するために、BrowserApplication クラスの Maximize メソッドを使用できます。

ブラウザを最大化するには、テスト スクリプトに次のコードを追加します。

```
'VB code
With _desktop.BrowserApplication("demo_borland_com")

    .Maximize()

End With
```

ブラウザをスクロールさせる方法

VB .NET の場合、Silk Test Workbench では、次のようにして再生中にブラウザの表示領域にコントロールをスクロールさせることができます。

ExecuteJavaScript メソッド (DomElement) ScrollIntoView メソッドを使用して、特定の DOM 要素をブラウザ ウィンドウの表示領域にスクロールできます。

ExecuteJavaScript メソッド (BrowserWindow) ExecuteJavaScript メソッドを使用して、範囲を指定してページ全体を上下にスクロールさせることができます。

使用例

次のコマンドは、1 ページ下にスクロールさせます。

```
'VB .NET code  
browserWindow.ExecuteJavaScript("window.scrollTo(0, window.innerHeight)")
```

次のコマンドは、1 00 ピクセル下にスクロールさせます。

```
'VB .NET code  
browserWindow.ExecuteJavaScript("window.scrollTo(0, 100)")
```

次のコマンドは、1 00 ピクセル上にスクロールさせます。

```
'VB .NET code  
browserWindow.ExecuteJavaScript("window.scrollTo(0, -100)")
```

現在使用しているブラウザの確認方法

BrowserApplication クラスには、ブラウザの種類を返すプロパティ "browsertype" があります。このプロパティをロケーターに追加することで、どのブラウザに一致させるかを定義できます。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

使用例

ブラウザの種類を取得するには、次のコードをロケーターに入力します。

```
browserApplication.GetProperty("browsertype")
```

また、BrowserWindow には、現在のウィンドウのユーザー エージェント文字列を返すメソッド GetUserAgent があります。

要素のテキストに使用されるフォント タイプの確認方法

属性名を「:」で区切ると、DOM 要素の currentStyle 属性のすべての属性にアクセスできます。

Internet Explorer 8 以前 wDomElement.GetProperty("currentStyle:fontName")

Internet Explorer 9 またはそれ以降および Mozilla Firefox などの他のすべてのブラウザ wDomElement.GetProperty("currentStyle:font-name")

innerText をカスタム クラス属性として構成したが、ロケーターで使えない

ロケーター文字列に使用する属性には最大長があります。InnerText は長くなりすぎる傾向があり、ロケーターで使えない場合があります。可能な場合は、textContent を代わりに使用してください。

xBrowser API で公開されていない機能が必要な場合の対処方法

ExecuteJavaScript() を使用して、JavaScript コードを Web アプリケーションから直接実行できます。この方法は、ほとんどすべての問題の回避策となります。

Link.Select で、Internet Explorer で新しく開いたウィンドウにフォーカスが設定されない

この制限は、ブラウザの構成設定を変更することで修正できます。新しく開いたウィンドウが常にアクティブ化されるようにオプションを設定します。

アプリケーションのログ出力に正しくないタイムスタンプが含まれる

この方法によって、同期に関して予期しない結果が発生する場合があります。この問題を回避するには、HTML 同期モードを指定します。

新しいページに移動したあと、スクリプトがハングする

この問題は、AJAX アプリケーションによりブラウザがビジー（サーバー プッシュ/ActiveX コンポーネントの接続が開いている）のままになっている場合に、発生することがあります。HTML 同期モードを設定してください。他のトラブルシューティングのヒントについては、「xBrowser のページ同期」のトピックを参照してください。

正しくないロケーターが記録されている

マウスを要素上に移動したときに、要素の属性が変更することがあります。Silk Test Workbench によってこのシナリオの追跡が試行されますが、失敗することがあります。影響を受ける属性を特定し、それが Silk Test Workbench で無視されるように構成してください。

Internet Explorer で要素を囲む四角形の位置が正しくない

- 拡大率が 100% に設定されていることを確認します。このようにしないと、四角形が正しく配置されません。
- ブラウザ ウィンドウの上に通知バーが表示されていないことを確認します。Silk Test Workbench では、通知バーを処理できません。

マウス移動設定がオンになっているにもかかわらず、すべての操作が記録されない理由

多くの無用な MoveMouse 操作がスクリプトに影響を及ぼさないように、Silk Test Workbench では以下の操作が行われます。

- マウスが一定時間静止している場合にのみ、MoveMouse 操作が記録されます。
- マウスを要素上に移動したあとで操作が行われていることが確認された場合にのみ、MoveMouse 操作が記録されます。場合によっては、スクリプトに手動操作を追加することが必要となることがあります。
- Silk Test Workbench は、Web アプリケーション、Win32 アプリケーション、および Windows Forms アプリケーションに対してのみ、マウス移動の記録をサポートします。Silk Test Workbench は、Apache Flex や Swing など、xBrowser テクノロジ ドメインの子テクノロジ ドメインのマウス移動を記録することはできません。

textContent、innerText、および innerHtml の違い

- textContents は、書式設定のみを目的とする要素およびその子要素に含まれるすべてのテキストです。
- innerText は、要素およびその子要素に含まれるすべてのテキストを返します。


- innerHtml は、要素に含まれるすべてのテキスト（html タグも含む）を返します。

以下の html コードについて検討します。

```
<div id="mylinks">
  This is my <b>link collection</b>:
  <ul>
    <li><a href="www.borland.com">Bye bye <b>Borland</b> </a></li>
    <li><a href="www.microfocus.com">Welcome to <b>Micro Focus</b></a></li>
  </ul>
</div>
```

以下の表に、返されるプロパティの詳細を示します。

コード	返される値
browser.DomElement("//div[@id='mylinks']").GetProperty("textContent")	This is my link collection:
browser.DomElement("//div[@id='mylinks']").GetProperty("innerText")	This is my link collection:Bye bye Borland Welcome to Micro Focus
browser.DomElement("//div[@id='mylinks']").GetProperty("innerHTML")	This is my link collection: Bye bye Borland Welcome to Micro Focus

 **注:** Silk Test 13.5 以降で、要素の textContents プロパティを介して取得されるテキストの空白はサポートするブラウザすべてで一貫してトリムされます。ブラウザのバージョンによっては、この空白の処理方法が Silk Test 13.5 以前の Silk Test バージョンで異なる場合があります。以前の動作を有効にしたい場合、OPT_COMPATIBILITY オプションを 13.5.0 より低いバージョンに設定してください。たとえば、オプションを Silk Test 13.0 に設定するには、スクリプトで次のように入力します：

```
'VB .NET code
Agent.SetOption("OPT_COMPATIBILITY", "13.0.0")
```

クロス ブラウザー スクリプトの作成時に必要な処置

クロス ブラウザー スクリプトを作成する場合は、以下の 1 つまたは複数の問題に遭遇する場合があります。

- クロス ブラウザー テスト用にスクリプトを記録する場合、Google Chrome、Mozilla Firefox、または Microsoft Edge を使用することを Micro Focus では推奨しています。Internet Explorer を使って Silk Test Workbench で記録したスクリプトは、ほかのブラウザで記録したスクリプトと若干異なる場合があります。
- 属性値が異なる。たとえば、Internet Explorer の色が "# FF0000" として、Mozilla Firefox の色が "rgb(255,0,0)" として返されます。
- 属性名が異なる。たとえば、Internet Explorer 8 以前のバージョンのフォント サイズ属性は "fontSize" でしたが、Internet Explorer 9 以降および Mozilla Firefox などの他のすべてのブラウザでは "font-size" となります。
- 一部のフレームワークで異なる DOM ツリーがレンダリングされることがある

安定したクロス ブラウザ テストを実現するために最適なロケーター

組み込みロケーター生成プログラムでは、安定したロケーターの作成が試みられます。ただし、情報を使用できない場合、高品質のロケーターを生成することは困難です。この場合、ロケーター生成プログラム

では、階層形式の情報およびインデックスが使用されます。その結果、直接的な記録／再生には適していても、安定した日常的な実行には適さない脆弱なロケーターが生成されます。さらに、クロス ブラウザ テストでは、いくつかの AJAX フレームワークで異なるブラウザに対して異なる DOM 階層がレンダリングされることがあります。

この問題を回避するには、アプリケーションの UI 要素にカスタム ID を使用します。

ロケーターでクラスとスタイルの属性が使用されない理由

これらの属性は AJAX アプリケーションで頻繁に変更され、ロケーターの安定性が損なわれることがあり、ため、無視リストに含まれています。ただし、多くの場合、これらの属性を使用してオブジェクトを識別できるため、アプリケーションで使用することに意味がある場合があります。

Internet Explorer 10 で Click の記録が異なる理由

Internet Explorer 10 の DomElement で Click を記録し、DomElement が Click の後で破棄された場合、記録動作が予期したとおりにならないことがあります。別の DomElement が最初の DomElement の下にある場合、Silk Test では、1 つの Click が記録されるのではなく、Click、MouseMove、および ReleaseMouse が記録されます。

予期しない記録動作を回避する方法は、テスト対象のアプリケーションによって異なります。通常は、記録されたスクリプトから不要な MouseMove イベントと ReleaseMouse イベントを削除すれば十分です。

ハンドル無効エラーが表示される理由

このトピックでは、Silk Test Workbench に「このオブジェクトのハンドルは無効になりました。」というエラー メッセージが表示された場合の対処法について説明します。

このメッセージは、たとえば WaitForProperty などのメソッドを呼び出したオブジェクトが何らかの理由で消失していることを示しています。たとえば、Web アプリケーションでメソッドを呼び出しているときに、何らかの理由でブラウザが新しいページに移動した場合、以前のページのすべてのオブジェクトは自動的に無効になります。

Web アプリケーションのテストでは、組み込みの同期がこの問題の原因の場合があります。たとえば、テスト対象のアプリケーションにショッピング カートが含まれていて、このショッピング カートに品物を追加したとします。ユーザーは次のページが読み込まれ、ショッピング カートのステータスが品物がある状態に変わるまで待機しています。品物を追加するという操作からの戻り時間が短すぎた場合、最初のページのショッピング カートはステータスが変化するまで待機しますが、その間も新しいページは読み込まれています。したがって、最初のページのショッピング カートは無効になります。この動作によって、ハンドル無効エラーが発生します。

この問題を回避するには、2 番目のページでのみ有効なオブジェクトが表示されるまで待機してから、ショッピング カートのステータスを確認するようにしてください。このオブジェクトが有効になるとすぐに、ショッピング カートのステータスを確認できるようになり、2 番目のページで正しく検証されるようになります。

スクリプトからのブラウザの起動

テストの開始時に再生するブラウザを選択するのではなく、再生時にテスト スクリプトから特定のブラウザを起動することが必要になる場合があります。

BrowserBaseState クラスの使用

テスト スクリプトからブラウザを起動するために、BrowserBaseState クラスを使用すると、Executable プロパティで指定したブラウザが実行され、テストの準備ができています。さらに、基本状態として Uri プロパティで指定した URL に移動され、ブラウザが最前面に表示されます。

次のサンプル コードは、BrowserBaseState を使用して Internet Explorer を起動します。

```
' VB .NET code
Dim MyBrowserBaseState As New BrowserBaseState(BrowserType.InternetExplorer, "http://
www.borland.com")
MyBrowserBaseState.Execute()
```

ブラウザの複数インスタンスの使用

複数のブラウザ ウィンドウまたはタブが開かれている場合、Silk Test Workbench は各ブラウザ ウィンドウまたはタブを、ユニークなロケーターを持つ個別のオブジェクトとして処理します。ロケーターは、WebBrowser、WebBrowser[1]、WebBrowser[2] のように、インデックス付きで表されます。

非表示入力フィールドの検索

非表示入力フィールドは、タグに type="hidden" を指定した HTML フィールドです。Find で非表示入力フィールドを検索できるようにするために、OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS オプションを使用できます。このオプションのデフォルト値は、TRUE です。

```
'VB .NET code
Agent.SetOption("OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS", true)

// C# code
Agent.SetOption("OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS", true);
```

Web アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Web アプリケーションがサポートする属性は次のとおりです。

- caption (次のワイルドカードをサポート：? および *)
- すべての DOM 属性 (次のワイルドカードをサポート：? および *)



注：各ブラウザによって、空のスペースの処理に違いがあります。この結果、「textContent」および「innerText」属性は正規化されています。空のスペースのあとに別の空のスペースが続く場合、空のスペースはスキップされるか、または 1 文字の空白で置き換えられます。空のスペースとは、検出されたスペース、キャリッジ リターン、改行、タブのことです。また、このような値に一致するものも正規化されます。例：

```
<a>abc
abc</a>
```

以下のロケーターを使用します。

```
//A[@innerText='abc abc']
```

Web アプリケーションのカスタム属性

HTML は、安定した識別子を表すことができる一般的な属性 ID を定義します。定義により、ID は文書内の要素を一意的に識別します。特定の ID を持つ要素は文書内で 1 つだけ存在します。

ただし、多くの場合 (特に AJAX アプリケーションでは)、ID は HTML 要素に関連付けられたサーバー ハンドラを動的に識別するために使用されます。つまり、Web 文書の作成のたびに ID は変わるようになります。このような場合、ID は安定した識別子ではなく、Web アプリケーションの UI コントロールを識別するのに適しません。

Web アプリケーションの場合、より確実にするには、Silk Test Workbench に UI コントロールの情報を公開するためだけに使用されるカスタム HTML 属性を新たに導入することです。

カスタム HTML 属性はブラウザーは無視するため、AUT の動作は変わりません。ブラウザーの DOM を通じてアクセスすることができます。Silk Test Workbench では、このような属性を (属性がコントロールクラスのカスタム 属性であっても) 識別時のデフォルト属性として使用するように設定することができます。特定のテクノロジドメインのデフォルト識別属性としてカスタム属性を設定するには、**ツール > オプション > 記録** をクリックして、テクノロジドメインを選択します。

アプリケーション開発者は、Web 要素にさらに HTML 属性を追加することが必要です。

元の HTML コード :

```
<A HREF="http://abc.com/control=4543772788784322..." <IMG  
src="http://abc.com/xxx.gif" width=16 height=16> </A>
```

新しいカスタム HTML 属性 *AUTOMATION_ID* を持つ HTML コード :

```
<A HREF="http://abc.com/control=4543772788784322..."  
AUTOMATION_ID = "AID_Login" <IMG src="http://abc.com/xxx.gif"  
width=16 height=16> </A>
```

カスタム属性を設定すると、Silk Test Workbench は、できる限りカスタム属性を使用して、一意のロケーターを構成しようとします。Web ロケーターは次のようになります。

```
...//DomLink[@AUTOMATION_ID='AID_Login']
```

例 : 変化する ID

変化する ID の 1 例は、Google Widget Toolkit (GWT) で、ID は Web 文書の作成のたびに変化する動的な値を保持します :

```
ID = 'gwt-uid-<nnn>'
```

この場合、<nnn> が頻繁に変化します。

xBrowser クラス リファレンス

xBrowser アプリケーションを設定すると、Silk Test Workbench は標準の xBrowser コントロールのテストのサポートを組み込みで提供します。

Microsoft Windows 10 上のテストの制限事項

以下に、Microsoft Windows 10 (Windows 10) 上でテストする際の既知の制限事項を一覧します。

- Silk Test は、Windows 10 上のユニバーサル Windows プラットフォーム (UWP) アプリのテストをサポートしません。

サポートする属性の種類

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。必要に応じて、以下のいずれかの方法を使用して属性の種類を変更できます。

- 他の属性の種類と値を手動で入力する。
- **推奨属性リスト** の値を変更して、デフォルトの属性の種類に対して別の設定を指定する。

Apache Flex アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Flex アプリケーションがサポートする属性は次のとおりです。

- automationName
- caption (automationName と同様)
- automationClassName (FlexButton など)
- className (実装クラスの完全修飾名。 mx.controls.Button など)
- automationIndex (FlexAutomation のビューでのコントロールのインデックス。 index:1 など)
- index (automationIndex と同様。ただし、接頭辞はなし。 1 など)
- id (コントロールの ID)
- windowId (id と同様)
- label (コントロールのラベル)
- すべての動的ロケーター属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

動的ロケーター属性の詳細については、「動的ロケーター属性」を参照してください。

Java AWT/Swing アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Java AWT/Swing でサポートされる属性には以下のものがあります。

- caption
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。 caption のないコントロールの場合、自動的に属性 **priorlabel** がロケーターに使用されます。コントロールの **priorlabel** 値 (テキスト入力フィールドなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。
- name
- accessibleName
- *Swing* のみ : すべてのカスタム オブジェクトの定義属性は、ウィジェットに putClientProperty("propertyName", "propertyValue") で設定されます。




注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Java SWT アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Java SWT がサポートする属性は次のとおりです。

- caption
- すべてのカスタム オブジェクト定義属性


 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

SAP アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジ ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

SAP がサポートする属性は次のとおりです。


- automationId
- caption

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Silverlight コントロールを識別するためのロケータ属性

Silverlight コントロールでサポートされているロケータ属性は次のとおりです。

- automationId
- caption
- className
- name
- すべての動的ロケータ属性

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。


Silverlight スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、*name*、または任意の動的ロケータ属性を指定できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下ようになります：

```
// SLButton[@automationId="okButton"]
```

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
automationId	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナ上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	<pre>// SLButton[@automationId="okButton"]</pre>
caption	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合	<pre>//SLButton[@caption="Ok"]</pre>

属性の種類	説明	例
	合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	
className	Silverlight コントロールの .NET 単純クラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 Silverlight コントロールから派生したカスタム コントロールを識別するのに役立ちます。	// SLButton[@className='MyCustomButton']
name	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	//SLButton[@name="okButton"]

 **注目:** XAML コードの *name* 属性は、ローケータ属性 *name* ではなく、ローケータ属性 *automationId* にマップされます。

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して Silverlight コントロールのローケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* が固有の場合は Silk Test Workbench がローケータを作成する際に使用されます。


以下の表は、アプリケーション開発者がテキスト「Ok」を持つ Silverlight ボタンをアプリケーションの XAML コードに定義する方法を示しています。

オブジェクトの XAML コード	Silk Test からオブジェクトを検索するためのローケータ
<Button>Ok</Button>	//SLButton[@caption="Ok"]
<Button Name="okButton">Ok</Button>	//SLButton[@automationId="okButton"]
<Button AutomationProperties.AutomationId="okButton">Ok</Button>	//SLButton[@automationId="okButton"]
<Button AutomationProperties.Name="okButton">Ok</Button>	//SLButton[@name="okButton"]

UI オートメーションを使ってコントロールを識別するためのローケータ属性

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールに対して、次のローケータ属性がサポートされています。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ローケータ属性

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ローケータ属性は、ワイルドカード ? および * をサポートしています。

動的ローケータ属性の詳細については、「動的ローケータ属性」を参照してください。

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールを識別するには、スクリプトで *automationId*、*caption*、*className*、*name*、またはすべての動的ローケータ属性を使用できます。 *automationId* はアプリケーション開発者が設定します。たと

例えば、*automationId* を持つロケータは、以下のようになります： //
UIButton[@automationId="okButton"]

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
automationId	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	// UIButton[@automationId="okButton"]
caption	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//UIButton[@caption="Ok"]
className	UI オートメーション コントロールのクラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 UI オートメーション コントロールから派生したカスタム コントロールを識別するのに役立ちます。	// UIButton[@className='MyCustomButton']
name	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	//UIButton[@name="okButton"]

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性を、この表に示した順番に使用して UI オートメーション コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* を使って固有に識別できれば、Silk Test Workbench によるロケータの作成に使用されます。


ユーザーの AUT の UI オートメーション コントロールに対して使用できるその他のカスタム属性を確認するには、**プロパティの検証** ダイアログ ボックスを使用できます。使用するときは、記録する UI オートメーション コントロール上にマウス カーソルを移動し、**Ctrl + Alt** をクリックします。これで、コントロールに対して使用できるプロパティを確認できます。たとえば、アプリケーションによっては、属性 value が有効な場合もあります。

Rumba コントロールを識別するためのロケータ属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。サポートされている属性は次のとおりです。

- caption** コントロールが表示するテキスト。
- priorlabel** フォームの入力フィールドには通常入力の目的を説明するラベルがあるため、**priorlabel** の目的は隣接するラベル フィールド **RumbaLabel** のテキストによってテキスト入力フィールド **RumbaTextField** を識別することです。テキスト フィールドの同じ行の直前にラベルがない場合、または右側のラベルが左側のラベルよりテキスト フィールドに近い場合、テキスト フィールドの右側にあるラベルが使用されます。
- StartRow** この属性は記録されていませんが、手動でロケータに追加することができます。**StartRow** を使用して、この行で始まるテキスト入力フィールド、**RumbaTextField** を識別します。
- StartColumn** この属性は記録されていませんが、手動でロケータに追加することができます。**StartColumn** を使用して、この列で始まるテキスト入力フィールド、**RumbaTextField** を識別します。

すべての動的ロケーター属性。 動的ロケーター属性の詳細については、「動的ロケーター属性」を参照してください。


 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Web アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Web アプリケーションがサポートする属性は次のとおりです。

- caption (次のワイルドカードをサポート : ? および *)
- すべての DOM 属性 (次のワイルドカードをサポート : ? および *)

 **注:** 各ブラウザによって、空のスペースの処理に違いがあります。この結果、「textContent」および「innerText」属性は正規化されています。空のスペースのあとに別の空のスペースが続く場合、空のスペースはスキップされるか、または 1 文字の空白で置き換えられます。空のスペースとは、検出されたスペース、キャリッジ リターン、改行、タブのことです。また、このような値に一致するものも正規化されます。例 :

```
<a>abc  
abc</a>
```

以下のロケーターを使用します。


```
//A[@innerText='abc abc']
```

Windows Forms アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Windows Forms アプリケーションがサポートする属性は次のとおりです。

- automationid
- caption
- windowid
- priorlabel (caption のないコントロールの場合、自動的に priorlabel が caption として使用されます。caption のあるコントロールの場合、caption を使う方が簡単な場合があります。)


 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Windows Presentation Foundation (WPF) アプリケーションの属性

WPF アプリケーションがサポートする属性は次のとおりです。

- automationId
- caption

- *className*
- *name*
- すべての動的ロケータ属性。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

オブジェクト解決

WPF スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、あるいは *name* を指定できます。アプリケーション中の要素に指定された *name* が利用可能な場合、ロケータの *automationId* 属性として使用されます。この結果、多くのオブジェクトは、この属性のみを使用して一意に識別できます。たとえば、*automationId* を持つロケータは、以下のようになります://

```
WPFButton[@automationId='okButton']"
```

automationId や他の属性を定義した場合、再生中に *automationId* だけが使用されます。*automationId* が定義されていない場合には、コンポーネントを解決するのに *name* が使用されます。*name* も *automationId* もどちらも定義されていない場合には、*caption* 値が使用されます。*caption* が定義されていない場合は、*className* が使用されます。*automationId* は非常に役立つプロパティであるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト アプリケーションの開発者によって提供された ID	//WPFButton[@automationId='okButton']"
<i>name</i>	コントロールの名前。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に名前を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この名前を使用します。	//WPFButton[@name='okButton']"
<i>caption</i>	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//WPFButton[@automationId='Ok']"
<i>className</i>	WPF の .NET 単純クラス名 (名前空間なし)。クラス名属性を使用すると、Silk Test Workbench が解決する標準 WPF コントロールから派生したカスタム コントロ	//WPFButton[@className='MyCustomButton']"

属性の種類	説明	例
	ールを識別するのに役立ちます。	

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して WPF コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、Silk Test Workbench がロケータを作成する際には *automationId* が使用されま

す。以下の例では、アプリケーション開発者がアプリケーションの WPF ボタンに対して *name* と *automationId* を XAML コードに定義する方法を示します。

```
<Button Name="okButton" AutomationProperties.AutomationId="okButton"
Click="okButton_Click">Ok</Button>
```

Windows API ベースのクライアント/サーバー アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Windows API ベースのクライアント/サーバー アプリケーションがサポートする属性は次のとおりです。

- *caption*
- *windowid*
- *priorlabel* : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。 *caption* のないコントロールの場合、自動的に属性 ***priorlabel*** がロケータに使用されます。コントロールの ***priorlabel*** 値 (テキスト ボックスなど) には、コントロールの左側または上にある最も近いラベルの *caption* が使用されます。



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性

再生中にコントロールを識別するために、事前に定義されたロケータ属性のセット (*caption* や *automationId* など。テクノロジー ドメインに依存します) をロケータに使用できます。しかし、動的プロパティを含む、コントロールのすべての属性をロケータ属性として使用することもできます。特定のコントロールで使用可能なプロパティのリストを取得するには、*GetPropertyList* メソッドを使用します。返されたプロパティはすべて、ロケータを使用してコントロールを識別するのに使用できます。



注: 特定のプロパティの実際の値を取得するには、*GetProperty* メソッドを使用します。この値はロケータで使用できます。

例

Silverlight アプリケーションのダイアログ ボックスにあるボタンを識別する場合、以下のように入力します。

```
browser.Find("//SLButton[@IsKeyboardFocused=true]")
```

または

```
Dim button = dialog.SLButton("@IsKeyboardFocused=true")
```

これが機能するのは、Silk Test Workbench により Silverlight ボタン コントロールの IsDefault というプロパティが公開されるためです。

例

Silverlight アプリケーションのフォント サイズ 12 のボタンを識別する場合、以下のように入力します。

```
Dim button = browser.Find("//SLButton[@FontSize=12]")
```

または

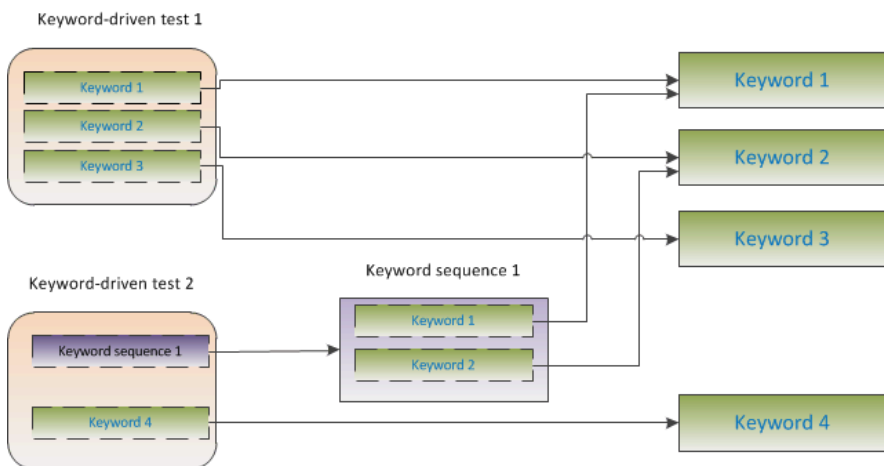
```
Dim button = browser.SLButton("@FontSize=12")
```

これが機能するのは、テスト対象アプリケーションの基になるコントロール (この場合、Silverlight ボタン) が FontSize というプロパティを持つためです。

キーワード駆動テスト

キーワード駆動テストは、テスト開発からテスト設計を分離するソフトウェアテスト手法です。このため、テスト自動化プロセスにビジネスアナリストなどの専門家を含めることができます。Silk Central と Silk Test はキーワード駆動テストをサポートしており、Silk Test のキーワードの形式での共有資産として構成されるメンテナンス可能な自動化フレームワークを自動化エンジニアが開発することによって、自動化エンジニアとビジネスアナリスト間で密接な共同作業を行うことができます。その後、これらのキーワードは、Silk Test で新しいキーワード駆動テストを作成したり、Silk Central で既存の手動テスト資産を自動テストに変換したり、新しいキーワード駆動テストを作成するために、ビジネスアナリストが使用することができます。

- キーワード駆動テストは、実行可能なキーワードのコレクションです。キーワード駆動テストは、他のテストと同様に再生することができます。
- キーワードシーケンスは、他のキーワードを組み合わせたものです。キーワードシーケンスは、頻繁に使用するキーワードの組み合わせを1つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。
- キーワードは、テストオブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。



キーワード駆動テストの作成には、次の2つのフェーズがあります。

1. テストの設計
2. キーワードの実装

キーワード駆動テストで利用可能な記録/再生コントロールの完全な一覧については、「キーワード駆動テストクラスリファレンス」を参照してください。

キーワード駆動テストの利点

キーワード駆動テスト手法を使用する利点を次に示します。

- キーワード駆動テストを使用すると、テスト自動化とテストケースのデザインが分離され、うまく分業できるようになり、キーワードを実装するテストエンジニアとテストケースをデザインする専門家が共同作業できます。
- テスト対象アプリケーションにアクセスすることなく、初期の段階からテストを開発でき、後からキーワードを実装できます。
- プログラムの知識がなくてもテストを開発できます。

- キーワード駆動テストは、長期的に見るとメンテナンス コストを低減できます。キーワードのメンテナンスが必要で、これらのキーワードを使用するすべてのキーワード駆動テストは自動的に更新されます。
- テストケースが簡潔です。
- 技術者でなくてもテスト ケースが読みやすく、理解しやすくなります。
- テスト ケースの変更が簡単です。
- 既存のキーワードを再利用して新しいテストを再利用できます。これにより、より広範囲なテスト カバレッジを実現しやすくなります。
- キーワード実装の内部的な複雑性を、キーワード駆動テストを作成または実行するユーザーに対して隠蔽できます。

キーワード

キーワードは、テスト オブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。Silk Test Workbench でのキーワードは、ビジュアル テストまたはメソッド名の前に Keyword 属性を持つメソッドです。キーワードは、キーワード資産として保存されます。

キーワード駆動テストの作成中にキーワードやキーワード シーケンスを定義し、後でそれらをテスト メソッド、またはビジュアル テストとして実装できます。既存のテスト メソッドまたはビジュアル テストに Keyword 属性を付けて、キーワードとしてマークすることもできます。キーワードは次のように定義されます：

```
'VB .NET code
<Keyword("keyword_name")>
```

キーワード シーケンスは、他のキーワードを組み合わせたものです。キーワード シーケンスは、頻繁に使用するキーワードの組み合わせを 1 つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。

キーワードまたはキーワード・シーケンスは、合計 20 の入力および出力パラメーターを持つことができます。キーワードを実装するテスト メソッドやビジュアル テストのパラメータは、キーワードのパラメータです。キーワードのパラメータに違う名前を指定するために、次を使用できます：

```
'VB .NET code
Argument("parameter_name")
```

例

キーワードとしてマークされたテスト メソッドは、次のようになります。

```
'VB .NET code
<Keyword("Login")>
Public Sub Login()
    ... // method implementation
End Sub
```

または

```
'VB .NET code
<Keyword("Login", Description:="Logs in with the given name and
password.")>
Public Sub Login(<Argument("UserName")> username As String,
<Argument("Password")> password As String)
    ... // method implementation
End Sub
```

このキーワードは、指定したユーザー名とパスワードを使ってテスト対象アプリケーションにログインします。



注: ヘルプ トピックを PDF で参照している場合、このサンプルコードは、実際のスクリプトでは許されない場所で改行されてしまっている場合があります。スクリプトでこのサンプルコードを使用する場合は、これらの改行を削除してください。

- Keyword 属性のキーワード名パラメータは、省略可能です。メソッドの名前とは異なる名前を指定する場合に、キーワード名パラメータを使用できます。パラメータが指定されていない場合、メソッドの名前がキーワード名として使用されます。
- Argument 属性も省略可能です。メソッドをキーワードとしてマークすると、自動的にすべての引数がキーワードの引数として使用されます。例のように、`userName` を `UserName` にしたい場合など、キーワードの引数とは異なる名前を指定する場合に、Argument 属性を使用できます。

Silk Test Workbench でキーワード駆動テストを作成する

キーワード駆動テスト エディター を使って、新しいキーワードと既存のキーワードを新しいキーワード駆動テストに結合できます。新しいキーワードは、後のステップでビジュアル テストまたは VB .NET スクリプトのメソッドとして実装する必要があります。

1. **ファイル** > **新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。

2. 資産の種類のリストから **キーワード駆動テスト** を選択します。

3. 新しい資産の名前を **資産名** フィールドに入力します。

4. 新しい資産を追加したいプロジェクトを選択します。

デフォルトでは、プロジェクトがアクティブであれば、そのアクティブなプロジェクトに新しい資産が作成されます。



注: Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

5. **OK** をクリックします。 **キーワード駆動テスト エディター** が開きます。

6. 次のアクションのいずれかを実行します。

- 新しいキーワードを追加する場合は、 **新しいキーワード** フィールドにキーワードの名前を入力します。
- 既存のキーワードを追加する場合は、 リストを展開して追加するキーワードを選択します。

7. **Enter** を押します。

8. 実行するすべてのキーワードを追加するまで、上記の 2 つの手順を繰り返します。

9. **保存** をクリックします。

続いて、キーワードを実装します。すべてのキーワードが実装されている場合は、テストを実行します。

Silk Test Workbench でのキーワード駆動テストの記録

単一のキーワードを記録する場合は、「[キーワードの記録](#)」を参照してください。

キーワード駆動テストを記録するには：

1. **ファイル** > **新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。

2. 資産の種類のリストから **キーワード駆動テスト** を選択します。


3. 新しい資産の名前を **資産名** フィールドに入力します。

4. 新しい資産を追加したいプロジェクトを選択します。

デフォルトでは、プロジェクトがアクティブであれば、そのアクティブなプロジェクトに新しい資産が作成されます。



注: Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

5. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
6. **OK** をクリックします。
7. **記録対象** ダイアログ ボックスで、次のアクションを実行します。
 - a) **資産の種類** フィールドで、キーワード駆動テストを .NET スクリプトとして保存するか、ビジュアルテストとして保存するかを選択します。
 - b) 省略可能: キーワード駆動テストを .NET スクリプトとして保存するように選択した場合、**資産名** フィールドに .NET スクリプトの名前を指定します。
デフォルトの名前は、最後に使用した .NET スクリプトの名前です。
 - c) **OK** をクリックします。
8. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザーの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能: あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能: ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能: **ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
9. 開いているダイアログに応じて、次のいずれかを実行します。
 - **アプリケーションの選択** ダイアログ ボックスで、**OK** をクリックします。
 - **ブラウザーの選択** ダイアログ ボックスで、**記録** をクリックします。
10. テスト対象アプリケーションで、最初のキーワードに含める操作を実行します。
記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
11. キーワードの名前を指定するには、**記録中** ウィンドウでキーワードの名前の上にマウスカーソルを動かして、**編集** をクリックします。
 **注:** Silk Test Workbench は、キーワード駆動テストの開始にアプリケーションの開始 キーワードを自動的に追加します。このキーワードで、アプリケーションの基本状態が実行され、テストを正しく再生できるようになります。基本状態についての詳細は、「基本状態」を参照してください。
12. キーワードの名前を **キーワードの名前** フィールドに入力します。
13. **OK** をクリックします。
14. 次のキーワードの操作を記録するには、**新しいキーワードの名前** フィールドに新しいキーワードの名前を入力し、**追加** をクリックします。Silk Test Workbench は、新しいキーワードに新しい操作を記録します。
15. キーワード駆動テスト全体を記録すまで、新しいキーワードを作成し、キーワードに対する操作を記録します。
16. **停止** をクリックします。 **記録完了** ダイアログ ボックスが開きます。

Silk Test Workbench は、すべての記録したキーワードを含む新しいキーワード駆動テストを作成します。

Silk Test Workbench でのキーワード駆動テストの基本状態の設定

Silk Test Workbench でキーワード駆動テストを実行すると、キーワード駆動テストは基本状態のキーワードを呼び出すことにより、Silk Test Workbench は AUT を基本状態から開始します。

キーワード駆動テストの記録時に、Silk Test Workbench は基本状態のキーワード (IsBaseState プロパティが *True* に設定されているキーワード) を現在のプロジェクトから検索します。

- 基本状態のキーワードが現在のプロジェクトに存在した場合、Silk Test Workbench は、キーワード駆動テストの最初のキーワードとして、このキーワードを挿入します。
- 基本状態のキーワードがプロジェクトに存在しない場合、Silk Test Workbench は、アプリケーションの開始 という名前の新しい基本状態のキーワードを作成し、キーワード駆動テストの最初のキーワードとして挿入します。

.NET メソッドとして実装されたキーワードを手動で基本状態のキーワードとしてマークするには、IsBaseState プロパティを Keyword 属性に追加し、プロパティの値を *True* に設定します。

```
<Keyword( "Start application" , IsBaseState:= True )>  
Public Sub Start_application()  
End Sub
```

ビジュアルテストとして実装されたキーワードを手動で基本状態のキーワードとしてマークするには、ビジュアルテストの <<開始>> ステップのプロパティを開き、**キーワード** セクションの **基本状態** プロパティを **はい** に設定します。

キーワード駆動テストのアプリケーション構成は、現在のプロジェクトに保存されます。アプリケーション構成を管理するには、Silk Test Workbench のメニューから **ツール** > **アプリケーション構成の編集** を選択します。これによって、**プロジェクトの選択** ダイアログ ボックスが開き、アクセス可能なすべてのプロジェクトが表示されます。プロジェクトを選択し、**編集** をクリックしてアプリケーション構成を編集します。

基本状態を含むキーワード駆動テストを実行すると、そのキーワード駆動テストが定義されているプロジェクトのアプリケーション構成が使用されます。

Silk Test Workbench でのキーワードの実装

キーワードを実装する前に、キーワード駆動テストの一部としてキーワードを定義します。

キーワード駆動テストで再利用するためにキーワードを実装するには：

1. 実装するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、実装するキーワードの左側に表示される **キーワードの実装** をクリックします。 **キーワードの実装** ダイアログ ボックスが開きます。
3. 新しいキーワードの資産の種類を選択します。
 - .NET メソッドとしてキーワードを実装するには、**.NET スクリプト** をクリックします。既存の .NET スクリプトに新しいキーワードを追加したり、**資産名** フィールドに名前を入力して、キーワードに対して新しい .NET スクリプトを作成できます。
 - キーワードシーケンスとしてキーワードを実装するには、**キーワード** をクリックします。中間レベルのキーワードは、他のキーワードのシーケンスであるため、記録することはできません。
 - ビジュアルテストとしてキーワードを実装するには、**ビジュアルテスト** をクリックします。
4. キーワードの名前を **資産名** フィールドに入力します。
5. 省略可能：キーワードを記録するには、**記録の開始** チェック ボックスをオンにします。

6. **OK** をクリックします。
7. **記録** をクリックします。

記録の詳細については、「[キーワードの記録](#)」を参照してください。

Silk Test Workbench でのキーワードの記録

完全に新しいキーワードに対してではなく、キーワード駆動テストに既に存在するキーワードに対しては、操作の記録のみを行えます。新しいキーワード駆動テストを記録する場合は、「[キーワード駆動テストの記録](#)」を参照してください。

新しいキーワードの操作を記録するには：

1. 記録するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、実装するキーワードの左側に表示される **キーワードの実装** をクリックします。 **キーワードの実装** ダイアログ ボックスが開きます。
3. 新しいキーワードの資産の種類を選択します。
 - .NET メソッドとしてキーワードを実装するには、**.NET スクリプト** をクリックします。既存の .NET スクリプトに新しいキーワードを追加したり、**資産名** フィールドに名前を入力して、キーワードに対して新しい .NET スクリプトを作成できます。
 - キーワード シーケンスとしてキーワードを実装するには、**キーワード** をクリックします。中間レベルのキーワードは、他のキーワードのシーケンスであるため、記録することはできません。
 - ビジュアルテストとしてキーワードを実装するには、**ビジュアルテスト** をクリックします。
4. **記録の開始** チェック ボックスをチェックします。
5. **OK** をクリックします。
6. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
7. テスト対象アプリケーションで、テストする操作を実行します。
記録中に利用可能な操作についての詳細は、「[記録中に利用可能な操作](#)」を参照してください。
8. **停止** をクリックします。 **記録完了** ダイアログ ボックスが開きます。

記録した操作は、定義したビジュアルテストまたは .NET スクリプトに表示されます。

ビジュアルテストをキーワードとして指定

既存のビジュアルテストをキーワードとして指定して、キーワード駆動テストで使用することができます。

1. キーワードとして指定するビジュアルテストを開きます。
2. **タスク** ペインで **<<開始>>** ステップをクリックします。
3. **プロパティ** ペインを開きます。
4. **キーワード** を **はい** に設定します。

これで、ビジュアルテストをキーワード駆動テストでキーワードとして使用できるようになります。

スクリプトのテストメソッドをキーワードとして指定

VB .NET スクリプトの既存のテストメソッドをキーワードとして指定して、キーワード駆動テストのメソッドを再利用します。

1. キーワードとして指定するテストメソッドを含む VB .NET スクリプトを開きます。
2. <Keyword()> をテストメソッドの直前に追加します。
デフォルトでは、キーワード名はテストメソッドの名前です。
3. 省略可能 : <Keyword("KeywordName")> をテストメソッドの直前に追加すると、キーワードに他の名前を設定できます。

これで、テストメソッドをキーワード駆動テストでキーワードとして使用できるようになります。

使用例

テストメソッド Login を *Login* という名前で新しいキーワードとして指定するには、テストメソッドの直前に以下を入力します。

```
'VB .NET code  
<Keyword("Login")>
```

2つの入力パラメータ UserName と PassWord を持つテストメソッド Login を *Login* という名前で新しいキーワードとして指定するには、以下を入力します。

```
// VB .NET code  
<Keyword("Login", Description="Logs in with the given name and  
password.")>  
Public Sub Login(Argument("UserName") As String, Argument("PassWord")  
As String)  
... //  
End Sub
```




注: ヘルプ トピックを PDF で参照している場合、このサンプルコードは、実際のスクリプトでは許されない場所で改行されてしまっている場合があります。スクリプトでこのサンプルコードを使用する場合は、これらの改行を削除してください。

キーワード間でのデータの受け渡し

キーワード間でパラメータを受け渡し、ビジュアルテストまたは .NET スクリプトとして実装されているキーワードから値を戻すことができます。ビジュアルテストのグローバル変数の値をキーワードに戻すことはできません。これらの変数は、ビジュアルテストの実行エンジンによって管理されており、.NET スクリプトやキーワードからはアクセスできないためです。


1. あるキーワードから他のキーワードにデータを渡したい場合 :
 - a) 出力パラメータをデータを渡したいキーワードに追加します。
 - b) 同じパラメータを入力パラメータとしてデータを渡されたいキーワードに追加します。
2. ビジュアルテストまたは .NET スクリプトとして実装されているキーワードからデータを戻したい場合 :
 - a) 出力パラメータをデータを戻したいキーワードに追加します。
 - b) データを戻されたいビジュアルテストまたは .NET スクリプトで、戻されたデータを格納するローカル変数を作成します。
 - c) ビジュアルテストまたは .NET スクリプトでローカル変数を使用します。

キーワード駆動テストの編集

 **注:** Silk Test Workbench では、Silk Test Workbench にあるキーワード駆動テストの編集と実行を行うことができ、また、Silk Central に格納されているキーワード駆動テストを実行することができます。Silk Central に格納されているキーワード駆動テストを編集するには、**キーワード駆動テストエディター** でキーワード駆動テストを開き、**編集** をクリックします。

キーワード駆動テストを編集するには：

1. **キーワード駆動テスト エディター** でキーワードを開きます。
 - a) **ウィンドウ > アセットブラウザ** をクリックします。**アセットブラウザ** が開きます。
 - b) **キーワード駆動テスト** をクリックします。
 - c) 編集するキーワード駆動テストをダブルクリックします。
2. 新しいキーワードをキーワード駆動テストに追加するには：
 - a) **新しいキーワード** フィールドをクリックします。
 - b) 新しいキーワードの名前を入力します。
 - c) Enter を押します。
3. 既存のキーワードを編集するには、キーワードの左側にある **キーワードを開く** をクリックします。

 **注:** Silk Central は、Silk Central で作成したすべてのキーワードの所有権を持ちます。このことは、このようなキーワードに対して行う変更は Silk Test Workbench でではなく、Silk Central で保存されることを意味します。

4. キーワードをキーワード駆動テストにコピーするには：

a) キーワードを選択します。



ヒント: 行番号の列で **Ctrl + クリック** または **Shift + クリック** を使用すると、複数のキーワードを選択できます。

b) **Ctrl + C** を押します。

c) キーワードを挿入する上の行を選択します。

d) **Ctrl + V** を押します。

5. キーワード駆動テストのほかの場所にキーワードを移動するには、キーワードをクリックして他の場所にドラッグするか、または次の手順を実行します。

a) キーワードを選択します。



ヒント: 行番号の列で **Ctrl + クリック** または **Shift + クリック** を使用すると、複数のキーワードを選択できます。

b) **Ctrl + X** を押します。

c) キーワードを挿入する上の行を選択します。

d) **Ctrl + V** を押します。





6. キーワード駆動テストからキーワードを削除するには、キーワードの左側にある **キーワードの削除** をクリックします。



キーワードは、**キーワード** ウィンドウでまだ利用可能なので、いつでもキーワード駆動テストに再度追加することができます。

7. 変更を保存するには、**新しいバージョンとして保存** をクリックします。

Silk Central でテストのキーワードを管理する

キーワード ページでは、選択したキーワード駆動テストのキーワードを管理できます。次のアクションを実行できます。

タスク	ステップ
テストまたはキーワード シーケンスを Silk Test で開く	Silk Test で開く をクリックして、選択したテストまたはキーワード シーケンスを Silk Test で開きます。
キーワードの追加	<ol style="list-style-type: none"> 1. キーワード リストの一番下にある 新しいキーワード をクリックするか、キーワードを右クリックして、コンテキスト メニューから キーワードを上挿入 を選択します。  注: キーワードの使用状況に基づいて、Silk Test にキーワードを推薦させることができます。推薦させるかどうかを切り替えるには、コンテキスト メニューの レコメンドの有効化 または レコメンドの無効化 を使用します。詳細については、「<i>Silk Test Workbench</i> が推薦するキーワード」を参照してください。 2. 利用可能なキーワードのリストからキーワードを選択するか、新しいキーワードを作成します。 3. 保存 をクリックします。 <p>または、右側にある すべてのキーワード ペインから、既存のキーワードをダブルクリックするか、ドラッグ&ドロップします。</p> <p> ヒント: Ctrl + クリック を使用すると、複数のキーワードを選択できます。ドロップするとき、キーワードは選択した順番に並び替えられます。</p>
キーワードの削除	削除するキーワードの アクション 列で  をクリックします。 保存 をクリックします。
キーワードの順序の変更	キーワードを移動したい位置にドラッグ&ドロップします。 保存 をクリックします。
キーワード シーケンス (他のキーワードから構成されるキーワード) の作成	<ol style="list-style-type: none"> 1. キーワード リストから結合したいキーワードを選択します。行番号の列で Ctrl + クリック または Shift + クリック を使用すると、複数のキーワードを選択できます。 2. 選択範囲を右クリックして、結合 をクリックします。 3. 新しいキーワード シーケンスの 名前 と 説明 を指定します。
キーワード シーケンスからキーワードの抽出	キーワード シーケンスを右クリックし、 キーワードの抽出 を選択します。これによって、元のキーワード シーケンスがそれに含まれるキーワードによって置換されますが、ライブラリからは削除されません。 保存 をクリックします。
テストまたはキーワード シーケンスへのキーワードのコピーと貼り付け	<ol style="list-style-type: none"> 1. キーワード リストからコピーしたいキーワードを選択します。行番号の列で Ctrl + クリック または Shift + クリック を使用すると、複数のキーワードを選択できます。 2. Ctrl + C を押すと選択項目がコピーされます。キーワードを移動する場合には、Ctrl + X を押します。 3. キーワードをコピーするテストまたはキーワード シーケンスを開き、キーワードを挿入する上の行を選択します。 4. Ctrl + V を押します。 <p> ヒント: 選択したキーワードを Excel に貼り付けて編集し、それをコピーしてテストまたはキーワード シーケンスに貼り付けることもできます。</p>
キーワード シーケンスのパラメータの定義	<ol style="list-style-type: none"> 1. キーワード リスト上の パラメータ をクリックします。パラメータ ダイアログ ボックスが表示されます。 2. パラメータの追加 をクリックします。

タスク	ステップ
	<ol style="list-style-type: none"> 3. 新しいパラメータに対して 名前 を指定します。パラメータが出力パラメータ (入力値を必要とするのではなく、値を戻します) の場合、出力 チェックボックスをチェックします。 4. OK をクリックします。 5. 保存 をクリックします。 <p> 注: キーワードまたはキーワード シーケンスは、入力パラメータと出力パラメータを合わせて 20 個まで持つことができます。</p>
ドラフト キーワードの編集	<ol style="list-style-type: none"> 1. 編集するドラフト キーワードのアクション 列で、 をクリックします。 2. キーワードの グループ を選択するか、新しいグループを指定します。 3. 説明 にキーワードの説明を入力します。この情報は、キーワードを実装するエンジニアにとって有益な情報です。 4. OK をクリックします。 5. 省略可能 : パラメータ フィールドをクリックして、キーワードのパラメータを追加します。キーワードが Silk Test で実装されると、これらのパラメータは、生成されたコード スタブに現れます。 6. 保存 をクリックします。
キーワードの検索	<p>キーワード ウィンドウ の検索フィールドを使用して、特定のキーワードを検索します。文字を入力するとリストが更新され、その文字に一致する既存のキーワードが動的に表示されます。検索のヒント :</p> <ul style="list-style-type: none"> • 検索では大文字小文字は区別されません。doAction を入力すると、doaction や DOAction が表示されます。 • 大文字のみを入力すると、いわゆる キャメルケース検索が実行されます。ECD を入力すると、Enter Car Details、Enter Contact Details、EnterContactDetails が表示されます。 • キーワード名とグループ名が検索対象になります。test を入力すると、test を含むすべてのキーワードと、グループ名に test を含むグループのすべてのキーワードが表示されます。 • ? は、0 または 1 文字と一致します。user?test を入力すると、userTest や usersTest が表示されます。 • * は、0 または n 文字と一致します。my*keyword を入力すると、myKeyword や myNewKeyword や my_other_keyword が表示されます。 • <文字列>. はグループ名のみを検索します。group. を入力すると、グループ名に group を含むグループのすべてのキーワードが表示されます。 • .<文字列> はキーワード名のみを検索します。.keyword を入力すると、keyword を含むすべてのキーワードが表示されます。 • <string>.<string> は特定のグループのキーワードを検索します。group.word を入力すると、myGroup グループの myKeyword が表示されます。 • 引用符を使用して完全一致のみを検索します。'Keyword' を入力すると、Keyword や MyKeyword は表示されますが、keyword は表示されません。

Silk Test Workbench のキーワード レcommend 機能

キーワード駆動テスト エディター でキーワード駆動テストにキーワードを追加するとき、Silk Test Workbench は、そのテストの次のキーワードとして使用する可能性のある既存のキーワードを推薦します。推薦するキーワードはキーワード リストの上位にリストされ、Silk Test Workbench がどの程度そのキーワードを推薦しているかが、棒グラフの塗りつぶした長さによって示されます。

Silk Test Workbench は、次のアルゴリズムに基づいてキーワードを推薦します。


- キーワード駆動テストまたはキーワード シーケンスに最初のキーワードを追加する場合、Silk Test Workbench は他のキーワード駆動テストまたはキーワード シーケンスの最初のキーワードとして使用されているのと同様なキーワードを検索します。最も頻繁に使用されるキーワードがより上位に推薦されます。
- 既に他のキーワードを含んでいるキーワード駆動テストまたはキーワード シーケンスに、さらにキーワードを追加する場合、Silk Test Workbench は次のようにしてキーワードを推薦します。
 - 新しいキーワードを追加するキーワード駆動テストまたはキーワード シーケンスの位置の前にキーワードがある場合、Silk Test Workbench は前のキーワード群と、すべてのほかのキーワード駆動テストとキーワード シーケンスのキーワードの組み合わせとを比較し、最も頻繁に使用されているキーワードの組み合わせに続いて現れるキーワードを推薦します。
 - キーワード駆動テストまたはキーワード シーケンスの位置の前にキーワードがないが、現在の位置の後にキーワードがある場合、Silk Test Workbench は後のキーワード群と、すべてのほかのキーワード駆動テストとキーワード シーケンスのキーワードの組み合わせとを比較し、最も頻繁に使用されているキーワードの組み合わせの前に現れるキーワードを推薦します。
- さらに、Silk Test Workbench は見つかったキーワードの類似性も考慮します。たとえば、2つのキーワードの名前とグループの両方が一致する場合、Silk Test Workbench は名前だけが一致するキーワードよりもこれらをより上位に推薦します。
- Silk Central との接続が確立すると、現在のプロジェクトに対応するキーワード ライブラリに属したキーワード駆動テストに含まれるキーワードも考慮されます。

キーワードでのパラメータの使用

キーワードまたはキーワード シーケンスは、入力パラメータと出力パラメータを合わせて 20 個まで持つことができます。このトピックでは、VB .NET スクリプトとして実装されたキーワードで作業している場合に、Silk Test Workbench を使用してこれらのパラメータを処理する方法について説明します。

キーワード駆動テスト エディター で、キーワードまたはキーワード シーケンスに対して定義されたパラメータを表示し、パラメータの値を編集できます。

キーワード ウィンドウで、キーワードまたはキーワード シーケンス上にマウス カーソルを動かすと、キーワードまたはキーワード シーケンスに割り当てられているパラメータを確認できます。

 **注:** キーワードがビジュアルテストとして実装されている場合、<<開始>> ステップのプロパティ ウィンドウの **入力パラメータ** と **出力パラメータ** セクションで入出力パラメータを指定できます。

単純なキーワードの入力パラメータ

他のテスト メソッドに対するものと同じ方法で、VB .NET スクリプトとして実装された キーワードの入力パラメータを定義して使用することができます。

次のサンプル コードでは、2つの入力パラメータ `userName` と `password` を持つキーワード `SetUserDetails` を定義する方法を説明します。

```
' VB .NET code
<Keyword>
public Sub SetUserDetails(userName As String, password As String)
```

```
...
End Sub
```

単純なキーワードの出力パラメータ

VB .NET スクリプトとして実装されたキーワードの戻り値または 1 つ以上の出力パラメータを定義することができます。また、戻り値と 1 つ以上の出力パラメータを組み合わせて使用することもできます。

次のサンプルコードでは、文字列を返すキーワード `GetText` を定義する方法を説明します。

```
' VB .NET code
<Keyword>
public Function GetText As String
    Return "text"
End Function
```

次のサンプルコードでは、2 つの出力パラメータ `userName` と `password` を持つキーワード `GetUserDetails` を定義する方法を説明します。

```
' VB .NET code
<Keyword>
public Sub GetUserDetails(ByRef userName As String, ByRef password As String)
    userName="name"
    password="password"
End Sub
```

キーワードシーケンスのパラメータ

キーワードシーケンスのパラメータは、**パラメータ** ダイアログ ボックスで定義し、編集できます。このダイアログ ボックスは、**キーワードシーケンスエディター** の **パラメータ** をクリックして開くことができます。

例：パラメータを取るキーワード

このトピックでは、パラメータを取るキーワードを使用する方法の例を紹介します。キーワードまたはキーワードシーケンスは、入力パラメータと出力パラメータを合わせて 20 個まで持つことができます。

最初の手順として、使用するキーワードを含んだキーワード駆動テストを作成します。これは、キーワード駆動テスト全体を記録するか、新しいキーワード駆動テストを作成してからキーワード駆動テストエディターでキーワードを追加することによって行います。

この例では、キーワード駆動テストは次のキーワードを含んでいます。

アプリケーションの開始 これは、AUT を開始して基本状態を設定する標準キーワードです。

Login これは、ユーザー名とパスワードで識別される特定のユーザーで AUT にログインするキーワードです。

GetCurrentUser このキーワードは、AUT に現在ログインしているユーザーの名前を返します。

AssertEquals このキーワードは、2 つの値を比較します。

Logout これは、AUT からユーザーをログアウトするキーワードです。

次のステップでは、パラメータをキーワードに追加します。これを行うには、キーワードのテストスクリプトを開き、メソッドにパラメータを追加します。

入力パラメータ `UserName` と `Password` をキーワード `Login` に追加するには、

```
' VB .NET code
<Keyword>
public Sub Login()
```



```
...  
End Sub
```

を、次のように変更します。

```
' VB .NET code  
<Keyword>  
public Sub Login(UserName As String, Password As String)  
...  
End Sub
```

出力パラメータ `UserName` をキーワード `GetCurrentUser` に追加するには、

```
' VB .NET code  
<Keyword>  
public Sub GetCurrentUser()  
...  
End Sub
```











を、次のように変更します。

```
' VB .NET code  
<Keyword>  
public Sub GetCurrentUser(ByRef currentUser As String)  
...  
End Sub
```

キーワード駆動テスト エディター 上のキーワード駆動テストは、次のようになります。

		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<i>currentUser ←</i>	
4	 	AssertEquals	<i>Expected</i>	<i>Actual</i>
5	 	Logout		

これで、**キーワード駆動テスト エディター** 上で入力パラメータの実際の値を指定できます。キーワード `GetCurrentUser` の出力パラメータ `UserName` の値を取得するには、`#{current user}` のように、変数を指定します。その後のキーワードで、変数を指定して格納された値を渡すことができます。


		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<code>#{current user}</code>	
4	 	AssertEquals	John Smith	<code>#{current user}</code>
5	 	Logout		

キーワードのキーワード シーケンスへの結合

キーワード駆動テスト エディター を使って、複数のキーワード駆動テストで順番に実行したいキーワードを結合してキーワード シーケンスを作成できます。

1. 結合するキーワードが含まれているキーワード駆動テストを開きます。
2. キーワード駆動テスト エディター で、Ctrl キーを押しながら結合したいキーワードをクリックします。
3. 選択範囲を右クリックして、**結合** をクリックします。**キーワードの結合** ダイアログ ボックスが開きます。
4. **名前** フィールドに、新しいキーワード シーケンスの名前を入力します。
5. 省略可能：**説明** フィールドに、新しいキーワード シーケンスの説明を入力します。
6. **結合** をクリックします。


新しいキーワード シーケンスが開き、**キーワード** ウィンドウにも表示されます。キーワード駆動テストでキーワード シーケンスを使用できます。

 **注:** 他のキーワードと同じように、キーワード シーケンス自身を実行することはできませんが、キーワード駆動テストの一部として実行することができます。

変数を指定したキーワード駆動テストの再生

キーワード駆動テスト用の変数の値を設定する前に、キーワード駆動テストを作成する必要があります。

1. 変数を指定して実行するキーワード駆動テストを開きます。
2. キーワード駆動テスト エディター で、**グローバル変数** をクリックします。**グローバル変数** ダイアログ ボックスが開きます。
3. **変数名** 列で、最初の空のフィールドに新しい変数の名前を入力します。
4. 変数の値を **値** 列に入力します。











 **注:** 値フィールドを空のままにすると、誤ったテスト結果が生じる可能性があります。

5. テストの実行に使用するすべての変数を指定するまで、前のステップを繰り返します。
6. **OK** をクリックします。
7. 実行するキーワード駆動テストを開きます。
8. キーワード駆動テスト エディター で、新しい変数を使用するパラメーターを編集します。

次のように記述します。

```
${variable name}
```

たとえば、次のキーワード駆動テストでは、`${current user}` パラメーターはグローバル変数を使用します。

		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<code>\${current user}</code>	
4	 	AssertEquals	John Smith	<code>\${current user}</code>
5	 	Logout		

キーワード駆動テストが Silk Test Workbench から実行されるときはいつでも、変数が使用されます。

Silk Test Workbench と Silk Central の統合

Silk Test Workbench と Silk Central を統合することによって、技術者と非技術者のユーザー間で共同作業を行えます。

Silk Test Workbench と Silk Central が統合され、Silk Central に存在するライブラリとアクティブな Silk Test Workbench プロジェクトが同じ名前であれば、**キーワード** ビュー (**表示** > **キーワード** から開く) には、アクティブな Silk Test Workbench プロジェクトで定義されたキーワードに加えて、Silk Central ライブラリのすべてのキーワードが表示されます。



注: Silk Central の接続情報は、すべての Silk Test Workbench ユーザーに別々に保存されるため、Silk Central のキーワードおよびキーワード シーケンスで作業するすべての Silk Test Workbench ユーザーは、Silk Test Workbench を Silk Central と統合する必要があります。

Silk Test Workbench と Silk Central を統合すると、次のメリットがあります。

- テスト管理と実行を Silk Central で処理できる
- キーワードが Silk Central データベースに格納され (ライブラリのアップロード)、Silk Central のすべてのプロジェクトで利用できる
- 手動テストを Silk Central で直接自動化し、作成したキーワード駆動テストを Silk Central から Silk Test Workbench で実行できる



注: Silk Test Workbench では、Silk Test Workbench にあるキーワード駆動テストの編集と実行を行うことができ、また、Silk Central に格納されているキーワード駆動テストを実行することができます。Silk Central に格納されているキーワード駆動テストを編集するには、**キーワード駆動テストエディター** でキーワード駆動テストを開き、**編集** をクリックします。

1. メニューから、**ツール** > **オプション** を選択します。 **オプション** ダイアログ ボックスが開きます。
2. **オプション** ツリーで、**Silk Central** を選択します。
3. **URL** フィールドに、Silk Central サーバーの URL を入力します。

たとえば、Silk Central のサーバー名が *sctm-server* で、Silk Central のポート番号が 13450 の場合は、`http://sctm-server:13450` と入力します。

4. 認証用 Web サービス トークンを指定します。

Web サービス トークンは、Silk Central の **ユーザー設定** ページから生成できます。このページは、Silk Central のメニューに表示されているユーザー名をクリックするとアクセスできます。



注: Silk Central ユーザー名とパスワードを使って認証を行う場合は、**認証** リストから **ユーザー名とパスワード** を選択します。ただし、セキュリティ上の理由から、ユーザー名とパスワードをネットワーク越しに送信するのではなく、Web サービス トークンを使用することを Micro Focus では推奨しています。


5. 有効なユーザー名とパスワードを、それぞれのフィールドに入力します。
6. **検証** をクリックして、Silk Test Workbench が指定したユーザーで Silk Central サーバーにアクセスできるかどうか確認します。
7. **OK** をクリックします。

Silk Central からキーワードを開く

Silk Central から Silk Test Workbench でキーワードを開くには、Silk Central でキーワードの名前の左にあるアイコンをクリックします。これにより、キーワードを実装する資産が Silk Test Workbench で開きます。

Silk Test Workbench のインスタンスが実行されていない場合は、ユーザー名とパスワードを入力するか、Windows 認証を使用してログインする必要があります。Silk Test Workbench は DSN を作成し、**ログイン** ダイアログでそれを自動的に選択します。キーワードが Silk Central にアップロードされたときのデータベースについての接続情報が含まれています。

Silk Test Workbench のインスタンスが実行されており、キーワードが Silk Central にアップロードされたときのデータベースと接続されている場合、キーワードはこのインスタンスで開かれます。


 **注:** キーワード ライブラリが Silk Test Workbench 16.0 から Silk Central にアップロードされたものである場合、Silk Test Workbench でキーワードを開く前にライブラリを再アップロードする必要があります。ライブラリを再アップロードしない場合、キーワードを使用する際に予期しない動作が発生する可能性があります。

Silk Test Workbench での Silk Central キーワードの実装

Silk Central キーワードを実装する前に、Silk Central でキーワード駆動テストの一部としてキーワードを定義します。

Silk Test Workbench で Silk Central キーワードを実装するには :

1. Silk Test Workbench のプロジェクトをキーワード駆動テストを含んだ Silk Central のキーワード ライブラリと同じ名前で作成します。
2. Silk Central のキーワード ライブラリにタイプが割り当てられていない場合、**ツール > キーワード ライブラリのアップロード** をクリックして、ライブラリ タイプを設定します。
3. 省略可能 : Silk Central の特定のキーワードを Silk Test Workbench で実装するには、ライブラリの **キーワード** タブを Silk Central で開き、キーワードの **アクション** 列から **Silk Test** で**実装** をクリックします。
4. Silk Test Workbench で、**表示 > キーワード** をクリックします。
5. **キーワード** ウィンドウで、キーワード駆動テストをダブル クリックします。
キーワード ウィンドウで Silk Central による変更を更新するには、**更新** をクリックします。
6. ツールバーで、**操作の記録** をクリックします。
7. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能 : あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能 : ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能 : **ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
8. **記録** をクリックします。
記録の詳細については、「**キーワードの記録**」を参照してください。
9. 最初の未実装のキーワードの操作を記録します。
- 10 現在のキーワードに対するすべての操作を記録し終わったら、**次のキーワード** をクリックします。
- 11 **記録中** ウィンドウでキーワードを切り替えるには、**前のキーワード** と **次のキーワード** をクリックします。
- 12 **停止** をクリックします。 **記録完了** ダイアログ ボックスが開きます。

 **注:** Silk Central のキーワード駆動テストは、Silk Test Workbench では読み取り専用であるため、キーワードを削除したり、キーワードの順番を変更することはできません。

Silk Central へのキーワード ライブラリのアップロード

Silk Central で作業するためには、有効な Silk Central の場所が設定されている必要があります。詳細については、「[Silk Test Workbench と Silk Central の統合](#)」を参照してください。

Silk Central で自動テストを自動化するために、Silk Test Workbench プロジェクトで実装したキーワードをキーワード ライブラリとして Silk Central にアップロードできます。そして、キーワードを使用して手動テストを自動化できます。

1. メニューで、**ツール > キーワード ライブラリのアップロード** を選択します。現在の Silk Test Workbench データベースにあるすべてのプロジェクトが表示されます。
2. Silk Central にアップロードするキーワードを含むプロジェクトを選択します。



注: プロジェクトと同じ名前のキーワード ライブラリが Silk Central 存在していることを確認します。

3. **OK** をクリックします。
4. 省略可能：キーワード ライブラリに対する変更の説明を指定します。
5. 省略可能：**設定** をクリックして、Silk Central への接続を設定します。
6. 省略可能：接続した Silk Central インスタンスで利用可能なライブラリを確認するには、リンクをクリックします。
7. **アップロード** をクリックします。



注意: Silk Central のキーワード ライブラリが既に他の自動化ツール、つまり他の Silk Test クライアントに割り当てられている場合、キーワード ライブラリのタイプを変更するかどうか確認されます。タイプの変更を行う場合にのみ、ライブラリはアップロードされます。

Silk Test Workbench は、プロジェクトで実装されたすべてのキーワードからキーワード ライブラリを作成します。その後、Silk Test Workbench は library.zip という名前で、キーワード ライブラリをプロジェクトの出力フォルダに保存します。ライブラリの整合性が検証され、Silk Central の既存のテストが影響を受ける変更が **キーワード ライブラリを Silk Central にアップロード** ダイアログ ボックスに一覧されます。最後に、Silk Test Workbench はライブラリを Silk Central にアップロードします。これで、Silk Central でキーワードを使用できるようになります。キーワード ライブラリに含まれるキーワードを使用する Silk Central のキーワード駆動テストは、現在のキーワードの実装を自動的に使用します。

現在の Silk Test Workbench データベースが Access データベースの場合、アップロードしたキーワードは、Access データベースにアクセスしたマシン上でのみ実行することができます。Silk Central にアップロードされたキーワード ライブラリには、キーワードの実装は含まれず、キーワードがこの Silk Test Workbench データベースで利用可能であるという情報のみ含まれます。

コマンド ラインから Silk Central へのキーワード ライブラリの更新

Java ベースのコマンド ラインから外部キーワード ライブラリを Silk Central にアップロードすることで、Silk Central とキーワード駆動テストを Jenkins などの継続的インテグレーション ビルド システムに統合することができます。

Java ベースのコマンド ラインからキーワード ライブラリを Silk Central にアップロードするには：

1. Silk Central で、**ヘルプ > ツール** を選択し、**Java キーワード ライブラリ ツール** をダウンロードします。
2. ダウンロードした jar ファイルに含まれるコマンド ライン ツールを次の引数で呼び出します。

- java

- -jar com.borland.silk.keyworddriven.jar
- -upload
- 更新または存在しない場合は作成される Silk Central のライブラリの ライブラリ名。
- アップロードするライブラリ パッケージ (zip アーカイブ) の パッケージ名。
- Silk Central フロントエンド サーバーのホスト名:ポート。
- Silk Central ユーザーの Web サービス トークン。認証に必要です。Web サービス トークンは、Silk Central の **ユーザー設定** ページから生成できます。このページは、Silk Central のメニューに表示されているユーザー名をクリックするとアクセスできます。



注: セキュリティ上の理由から、ユーザー名とパスワードをネットワーク越しに送信するのではなく、Web サービス トークンを使用することを Micro Focus では推奨しています。

- Silk Central ユーザーの ユーザー名。Web サービス トークンを認証に使用する場合は必要ありません。
- Silk Central ユーザーの パスワード。Web サービス トークンを認証に使用する場合は必要ありません。
- ライブラリに適用された変更を説明する 更新情報 (引用符で囲む)。
- テストまたはキーワード シーケンスで使用されているキーワードの削除を許可するための [-allowUsedKeywordDeletion] フラグ (省略可能)。デフォルトでは、使用中のキーワードを削除しようとするとエラーが発生します。

Java 9 以降を使ってライブラリを Silk Central にアップロードするコマンド ラインは次のようになります。

```
java --add-modules=java.activation,java.xml.ws -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```

使用例

認証に Web サービス トークンを使い、Java 8 以前を使ってライブラリを Silk Central にアップロードするコマンド ラインは次のようになります。

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scToken "Build xy:
Implemented missing keywords"
```

認証にユーザー名とパスワードを使い、Java 8 以前を使って上記と同じライブラリをアップロードするコマンド ラインは次のようになります。

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```

Java 9 以降を使った場合は、それぞれ次のようになります。

```
java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scToken "Build xy:
Implemented missing keywords"
```

```
java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```



注: Java 9 以降を使ってキーワード駆動ライブラリをアップロードする場合は、実行サーバー上で JAVA_HOME が定義され、対応するバージョンの JDK を指していることを確認してください。

キーワードの検索

キーワード ウィンドウの検索フィールドを使用して、特定のキーワードを検索します。文字を入力するとリストが更新され、その文字に一致する既存のキーワードが動的に表示されます。検索のヒント：

- 検索では大文字小文字は区別されません。doAction を入力すると、doaction や DOAction が表示されます。
- 大文字のみを入力すると、いわゆる キヤメルケース検索が実行されます。ECD を入力すると、Enter Car Details、Enter Contact Details、EnterContactDetails が表示されます。
- キーワード名とグループ名が検索対象になります。test を入力すると、test を含むすべてのキーワードと、グループ名に test を含むグループのすべてのキーワードが表示されます。
- ? は、0 または 1 文字と一致します。user?test を入力すると、userTest や usersTest が表示されます。
- * は、0 または n 文字と一致します。my*keyword を入力すると、myKeyword や myNewKeyword や my_other_keyword が表示されます。
- <文字列>. はグループ名のみを検索します。group. を入力すると、グループ名に group を含むグループのすべてのキーワードが表示されます。
- .<文字列> はキーワード名のみを検索します。.keyword を入力すると、keyword を含むすべてのキーワードが表示されます。
- <string>.<string> は特定のグループのキーワードを検索します。group.word を入力すると、myGroup グループの myKeyword が表示されます。
- 引用符を使用して完全一致のみを検索します。'Keyword' を入力すると、Keyword や MyKeyword は表示されますが、keyword は表示されません。

キーワードのフィルタリング

現在のプロジェクトの特定のキーワードを検索するために、**キーワード** ウィンドウに表示されているキーワードをフィルタすることができます。Silk Central との統合が設定されている場合、結果には Silk Central に関連付けられたキーワードも含まれます。

1. メニューで **表示 > キーワード** をクリックします。**キーワード** ウィンドウが開きます。
2. **キーワード** ウィンドウで、検索フィールドに検索するキーワードの名前を入力します。**キーワード** ウィンドウに、現在のプロジェクト内の指定した名前を持つすべてのキーワードが表示されます。
3. 省略可能：キーワードを編集するには、**キーワード** ウィンドウでキーワード上にマウスカーソルを移動して、**実装へ移動** をクリックします。

キーワードのグループ化

キーワードをライブラリで構造化するために、グループ化することができます。

このトピックでは、キーワードを特定のグループに追加する方法について説明します。このグループ名は Silk Central でも使用され、キーワードはグループ名に従って分類されます。

キーワードを特定のグループに追加するには：

1. キーワードの実装を開きます。
 - a) キーワードを実装しているプロジェクトをアクティブ化します。
 - b) **キーワード** ウィンドウを開きます。
 - c) **キーワード** ウィンドウで、キーワードを選択します。
 - d) **実装へ移動** をクリックします。

2. キーワードが VB .NET スクリプトで実装されている場合は、キーワード グループをスクリプトの最初に追加します。

たとえば、キーワードを Calculator グループ追加するには、次のように入力します。

```
<KeywordGroup("Calculator")>
```

3. キーワードがビジュアル テストで実装されている場合は、キーワード グループをビジュアル テストのプロパティに追加します。

a) ビジュアル テストの <<開始>> ステップを選択します。

b) **プロパティ** ペインを開きます。

c) **キーワード グループ** プロパティに、グループの名前を入力します。

キーワード ウィンドウで表示されるキーワード名にグループが含まれるようになります。たとえば、*Addition* キーワードが *Calculator* グループに属している場合は、*Calculator.Addition* として表示されます。

コマンドライン

Silk Test Workbench には、テストを開いて実行するためのコマンドラインプログラムが 2 つ含まれています。

SilkTest.Exe コマンドライン

SilkTest.exe コマンドラインプログラムを使用すると、ログインダイアログボックスをスキップして Silk Test Workbench を開くことができます。これは ¥Silk¥Silk Test¥ng¥gui¥ にあります。

パラメータ

パラメータの長い名前	パラメータの短い名前	説明
userid または user	-u	ログインに使用するユーザー名。
pass または password	-p	ログインユーザーのパスワード。
dsn	-d	使用する DSN。

例

```
SilkTest.exe -u admin -p mypassword -d mydsn
```

STW.EXE コマンドライン

Silk Test Workbench には、バッチスクリプト実行の 1 つ以上の .NET スクリプトまたはビジュアルテストを実行するために使用できるコマンドラインプログラムが含まれています。コマンドラインはエラー情報を返し、オプションで結果の出力ファイルを作成します。

コマンドラインプログラムの名前は stw.exe であり、次の場所にあります : ¥Silk¥Silk Test¥ng¥gui¥.


コマンドラインの構文は次のとおりです :


```
STW -d DSN_Name -u User_Name -p User_Password -s Script_Name
```

このセクションのトピックでは、必須パラメータとオプションパラメータを指定してプログラムを実行する方法について説明します。



STW.EXE のパラメータ


コマンドラインプログラムには、必須パラメータとオプションパラメータの両方があります。パラメータは、長い形式と短い形式のどちらかの名前を使用して指定できます。一部のオプションパラメータでは、環境の変更、スクリプトに変数を渡したり、出力ファイルの作成などを指定できます。

 **注:** パラメータ名、スクリプト、スクリプトパラメータでは、大文字と小文字が区別されません。ただし、スクリプトパラメータの値は、指定どおりにスクリプトに渡されます。

 **注:** パラメータ名の前には、文字 - の代わりに文字 / を使用できます。

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
dsn	-d	はい	データベースへの接続に使用する DSN の名前。このパラメータは必須ですが、空にすると <i>SilkTest</i> が使用されます。
username	-u	はい	接続に使用されるユーザー名。空にすると <i>Admin</i> が使用されます。
password	-p	いいえ	接続に使用されるパスワード名。パスワードが <i>Nothing</i> の場合はオプション (お勧めできません)。
project	-r	いいえ	再生するスクリプトを含むプロジェクト。デフォルトでは、プロジェクトのすべてのスクリプトが実行されます。固有のスクリプトを指定するには、 <code>-script</code> パラメータを使用します。スクリプトを指定せず、スクリプトで入力パラメータが必要である場合は、デフォルト値が使用されます。 .NET スクリプトが先に実行され、ビジュアルテストが次に実行されます。 プロジェクトが存在しない場合は、エラーメッセージが表示されます。  重要: プロジェクトパラメータが指定されていない場合、 <i>Common</i> プロジェクトがデフォルトで使用されます。たとえば、「 <code>stw -d silktest -u admin -p admin -s Script1 -r OtherProject -s Script2</code> 」のように指定した場合は、 <i>Script1</i> は <i>Common</i> プロジェクトで実行され、 <i>Script2</i> は <i>OtherProject</i> という名前のプロジェクトで実行されます。
environment	-e	いいえ	スクリプトに使用する実行環境の名前。このパラメータを含めない場合は、デフォルトの再生環境が使用されます。 再生環境が存在しない場合、スクリプトは実行されません。  重要: 使用する場合は、このパラメータの次に <code>project</code> と <code>script</code> パラメータを指定する必要があります。
script	-s	はい	再生する .NET スクリプト、ビジュアルテスト、またはキーワード駆動テストの名前。 .NET スクリプト、ビジュアルテスト、またはキーワード駆動テストが存在しない場合、エラーメッセージが表示されます。
variable	-m	いいえ	このパラメータを使用すると、.NET スクリプト、ビジュアルテスト、キーワード駆動テストのパラメータを指定できます。たとえば、このパラメータを使用して、 <i>Silk Central</i> からキーワード駆動テストにグローバルプロパティを渡すことができます。 指定すると、資産で定義されたグローバルプロパティは上書きされます。このパラメータを使用する場合は、 <code>script</code> パラメータの次に指定する必要があります。 名前と値のペアは、二重引用符で囲む必要があります。変数に割り当てる値を表すには、文字 <code>=</code> を使用します。ブール値パラメータは、 <i>True</i> または <i>False</i> として指定する必要があります。

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
file	-f	いいえ	<p>パラメータは任意の順序で指定できますが、名前と数はスクリプトのパラメータと正確に一致する必要があります。渡す名前が多すぎたり、少なかったり、無効だったりすると、エラーメッセージが表示されてスクリプトは実行されません。</p> <p>このオプションでは、コマンドラインパラメータをテキストファイルで指定し、コマンドラインプログラムに渡すことができます。再入力したくない長い引数がある場合にこれを使用できます。</p> <p>使用法の詳細については、「入力ファイル」を参照してください。</p>
outputfile	-o	いいえ	<p>実行されたスクリプトに関する情報を含むファイルの名前を指定するには、このパラメータを使用します。実行されて、正常に完了したか再生エラーを生成したスクリプトのみが出力ファイルに表示されます。</p> <p>パスや名前にスペースが含まれる場合は、出力ファイルの名前を引用符で囲む必要があります。</p> <p>使用法の詳細については、「出力ファイル」を参照してください。</p>
browser	-b	いいえ	<p>ブラウザー アプリケーション構成を持つスクリプトが実行されるブラウザーを指定するには、このパラメータを使用します。パラメータには次の値を指定できます (文字小文字は区別されません)。</p> <ul style="list-style-type: none"> • InternetExplorer • Firefox • GoogleChrome • Edge • AndroidBrowser • Safari • None <p>AndroidBrowser または Safari を使用する場合には、mobiledevice パラメータを指定します。</p> <p> 重要: 使用する場合は、このパラメータの次に project と script パラメータを指定する必要があります。</p>
commandlineargs	-cla	いいえ	<p>スクリプトの基本状態に渡すコマンドライン引数を指定する場合には、このパラメータを使用します。コマンドライン引数を使用する任意の基本状態に対して、このパラメータを使用できます。エージェントがコマンドライン引数を使用する場合、この設定が渡されますが、それ以外の場合は、無視されます。</p> <p>コマンドライン引数の指定には、<i>silktest.commandLineArgs</i> 環境変数を使用することもできます。STW.exe のパラメータと環境変数の両方でコマンドライン引数が指定された場合は、STW.exe のパラメータが使用されます。</p> <p> 重要: 使用する場合は、このパラメータの次に script パラメータを指定する必要があります。</p>
viewportwidth	-vw	いいえ	<p>ブラウザー ビューポートの幅をピクセルで指定する場合には、このパラメータを使用します。viewportwidth パラメータ</p>

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
viewportheight	-vh	いいえ	と viewportheight パラメータは同時に指定する必要があります。いずれか一方のパラメータだけを指定すると、エラーになります。viewportname パラメータを、このパラメータと同時に指定することはできません。
viewportname	-vn	いいえ	ブラウザ ビューポートの高さをピクセルで指定する場合には、このパラメータを使用します。viewportheight パラメータと viewportwidth パラメータは同時に指定する必要があります。いずれか一方のパラメータだけを指定すると、エラーになります。viewportname パラメータを、このパラメータと同時に指定することはできません。
orientation	-on	いいえ	UI であらかじめ定義したカスタム ブラウザー サイズのリストの名前を使用して、ビューポートの幅と高さを指定する場合には、このパラメータを使用します。このリストは、 ブラウザーの選択 ダイアログ、 アプリケーションの選択 ダイアログの Web タブ、 ブラウザー アプリケーション構成の編集 ダイアログで、ブラウザーを選択するときに表示されます。viewportwidth パラメータと viewportheight パラメータを、このパラメータと同時に指定することはできません。
mobiledevice	-md	いいえ	orientation パラメータは、viewportname が指定されたときだけ有効になります。その値として、 <i>portrait</i> または <i>landscape</i> を指定できます。UI の 向き の設定に該当します。
append	-a	いいえ	スクリプトを実行するモバイル デバイスの名前。この名前は、 ブラウザーの選択 ダイアログ ボックスにも表示されます。 この値をリセットするには、二重引用符で囲んだ空の文字列 ("") を渡します。  重要: 使用する場合は、このパラメータの次に project と script パラメータを指定する必要があります。
increment	-i	いいえ	結果を追加するには、このパラメータを使用します。 オプション ダイアログ ボックスに表示されるものと同じ設定です。
verbose	-v	いいえ	結果を増分するには、このパラメータを使用します。 オプション ダイアログ ボックスに表示されるものと同じ設定です。
help または ?	-h	いいえ	このパラメータでは、スクリプトの実行中にユーザーに出力が表示されます。コマンド ラインの任意の場所に表示されることがあります。 DSN、ユーザー名、パスワードは常に表示されます。出力ファイルを使用する場合は、それが表示されます。スクリプト名が表示される場所では、プロジェクト名が表示され、名前が括弧で囲まれて後に続きます。スクリプトの名前、プロジェクト、スクリプト ステータス、合計検証数、合格した検証数、失敗した検証数を含む行は、実行されたスクリプトごとに表示されます。
resultdir	-rd	いいえ	コマンドラインプログラムの使用可能パラメータを表示するには、「STW -?」と入力します。
			実行したスクリプトのすべての結果ファイルを保存するディレクトリを指定するには、このパラメータを使用します。 次のファイルがこのディレクトリに保存されます。 <ul style="list-style-type: none">XML 結果ファイル

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
connectionstring	-c	いいえ	<ul style="list-style-type: none"> XML 結果ファイルに対応する XSLT ファイル エラー画面 (実行中にエラーが発生した場合) リモート ロケーションに接続されているブラウザーやモバイル デバイスをテストするには、このパラメータを使用します。connectionstring パラメータを使用してテストを実行するとき、モバイル デバイスとパラメータの間で connectionstring は出力ファイルに書き出されます。

例 1

この例では、Internet Explorer でブラウザー スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -script MyScript
```

例 2

この例では、Internet Explorer と Google Chrome の両方でブラウザー スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -script MyScript -browser GoogleChrome -script MyScript
```

例 3

この例では、まず、Android ブラウザーでスクリプトを実行し、それからスクリプトのアプリケーション構成で設定されたブラウザーで実行するコマンドを示します。

```
STW -dsn silktest -browser AndroidBrowser -mobiledevice Nexus_4 -script MyScript -browser None -mobiledevice "" -script MyScript
```

例 4

この例では、モバイル デバイス上の Chrome for Android で Common プロジェクトのすべてのスクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser GoogleChrome -mobiledevice Nexus_4 -project Common
```

例 5

この例では、Mac に接続されたモバイル デバイス上の Chrome for Android で Common プロジェクトのすべてのスクリプトを実行するコマンドを示します。

```
STW -dsn silktest -connectionstring "platformName=Android;deviceName=Moto G;host=http://MyMac - GoogleChrome" -project Common
```

例 6

この例では、ブラウザー ビューポート サイズ 800x600 の Internet Explorer でブラウザー スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -viewportwidth 800 -vieportheight 600 -script MyScript
```

例 7

この例では、ブラウザー ビューポート サイズ SVGA (800, 600) の Internet Explorer でブラウザー スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -viewportname SVGA -script MyScript
```

例 8

この例では、カスタム プロファイルを指定した Mozilla Firefox でブラウザー スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -username admin -password admin -browser Firefox -commandLineArgs "-p MyProfile" -project Common -script BrowserTest
```

入カファイル

-file オプションを使用すると、テキスト ファイルでコマンド ライン パラメーターを指定して、コマンド ライン プログラムに送信することができます。

ファイルはコマンド ラインの任意の場所で指定できますが、指定した順番に処理され、その前後のコマンド ライン オプションに影響することがあります。また、複数の -file コマンドを指定（後にファイル名を指定）することも、ファイル内で -f パラメーターを指定することもできます。パラメーターは複数行で指定できます。

行をコメントアウトするには、# 文字を使用します。

ファイル内に別の **-file** パラメーターが含まれている場合は、再帰の使用について確認されます。つまり、ファイル内の -file パラメーターが、同じ処理セット内の任意の場所で参照されているのと同じファイルを示すことがあってはなりません。

ファイルの有効な形式は、UTF-8、UTF-16、または Unicode です。

例 1 : FileA.txt

```
# Execute my script
-script MyScript
# Change the environment and run again
-environment MyEnvironment
-script MyScript
```

例 2 : FileB.txt

```
# Include contents of FileA.txt, twice
-file FileA.txt

# Reset the playback environment
-environment

# Include it again
-file FileA.txt
```

出力ファイル

outputfile パラメータを使用すると、コマンドラインは指定ファイルを作成して内容を上書きしようとします。ファイルにアクセスできない場合はエラーメッセージが表示されますが、実行は進みます。これが発生する可能性がある理由は次のとおりです。

- パスが存在しない。
- 別のプロセスによってロックされているか使用されている。
- そのフォルダのアクセス権がユーザーにない。

出力ファイル形式

作成されるファイルはテキストファイルであり、各行は実行されたスクリプトを表します。各行には多くのフィールドが含まれ、フィールドはタブ文字 (0x9) で区切られます。これは TXT 形式であり、Microsoft Office Excel で使用できます。

出力ファイルのフィールド

フィールド	説明
スクリプト名	.NET またはビジュアル テスト スクリプトのどちらかになります。
プロジェクト	スクリプトが存在するプロジェクト。コマンドラインにプロジェクトを指定しない場合は、ここに「Common」と表示されます。
実行環境	スクリプトの実行に使用された実行環境の名前。デフォルトの実行環境が使用された場合、このフィールドは空です。
合計検証数	スクリプトによって実行された検証の総数。
合格した検証数	合格した検証の数。
失敗した検証数	失敗した検証の数。
スクリプト ステータス	スクリプトのステータス。
再生エラー	再生エラーメッセージが含まれます。再生エラーが検出されない場合は空です。
ブラウザー	スクリプトの実行に提供されたブラウザーの名前。
接続文字列	モバイル デバイスまたはリモート ロケーションへの接続文字列。
スクリプト パラメータ	スクリプトがパラメータを取る場合は、パラメータがフィールドのペアでここにリストされます。最初のフィールドがパラメータ名、次のフィールドが値です。

エラー数

コマンドライン プログラムは ERRORLEVEL をオペレーティングシステムに返し、呼び出し側プログラムはこれを分析して、実行時に発生したことを判断できます。次の表ではエラーについて説明します。

コード	説明	スクリプトのステータスとして出力ファイルで返されるかどうか
0	エラーはありません。すべてのスクリプトは正常に実行されました。	はい
1	致命的なエラー。コマンドライン プログラムをまったく実行できません。言語リソース ファイルがないことが理由の 1 つと考えられます。	いいえ

コード	説明	スクリプトのステータスとして出力ファイルで返されるかどうか
2	Silk Test COM オブジェクトを作成できませんでした。COM オブジェクトが登録されたことを確認してください。Silk Test.exe を実行すると COM オブジェクトが登録されることに注意してください。	いいえ
3	DSN、ユーザー名、パスワードのいずれかが正しくありません。詳細については、「 Silk Central から STW.EXE を実行するための前提条件 」を参照してください。	いいえ
4	スクリプト名が指定されていません。	いいえ
5	指定プロジェクトが見つかりませんでした。	いいえ
6	指定スクリプトが見つかりませんでした。	いいえ
7	指定実行環境が見つかりませんでした。	いいえ
8	スクリプトのパラメータが無効です。指定したスクリプトのパラメータが少なすぎるか多すぎるか、パラメータが見つかりませんでした。	いいえ
9	スクリプトの実行が完了しませんでした。最後に達する前に停止した可能性があります。再生エラーという結果にはなっていません。	はい
10	スクリプトで再生エラーが発生しました。	はい
11	スクリプトが実行できなかったか、スクリプトの検証がエラーになりました。	はい
12	複数の問題が発生しました。たとえば、スクリプトがエラーになり、スクリプト再生エラーが検出されました。	いいえ
13	-file オプションのためにコマンド ラインを処理できなかった場合、これが返されます。たとえば、ファイルが存在しないか、ファイルの再帰がある場合です。	いいえ
14	コマンド ライン ウィンドウの停止またはシャットダウンを行う方法はさまざまです。Ctrl+C、Ctrl+Break、Windows からのログアウト、Windows のシャットダウンのいずれかにより、コマンド ラインは閉じるか停止します。このような操作は捕えられて、このエラー コードが返されます。ウィンドウを閉じると、スクリプトの停止、製品のログアウト、SilkTest.EXE のアンロードも行われます。	いいえ
15	viewportwidth または viewportheight のいずれか一方だけが指定されました。	いいえ
16	viewportname が viewportwidth と viewportheight と同時に指定されました。	いいえ

バッチ ファイルを使用して ERRORLEVEL を確認する

バッチ ファイルで ERRORLEVEL を確認する方法は多数あります。次の例で 1 つの方法について説明します。

```
@ECHO OFF
SETLOCAL

REM -----
REM Define the error codes that can be returned by STW.EXE
REM -----
SET eNoError=0
SET eFatalError=2
SET eUnableToCreateComObject=2
SET eLoginFailed=3
SET eNoScriptSpecified=4
SET eProjectNotFound=5
```



```

SET eScriptNotFound=6
SET eRunEnvNotFound=7
SET eInvalidScriptParameters=8
SET eScriptDidNotComplete=9
SET eScriptPlaybackError=10
SET eScriptFailed=11
SET eMultipleProblems=12
SET eCommandCouldNotBeProcessed=13
SET ePrograTerminatedByUser=14
SET eInvalidViewportSize=15
SET eInvalidViewportParameters=16

REM -----
REM Run STW.EXE
REM -----
@ECHO ON
STW.EXE -username Admin -dsn STW-Scratch -script ScriptNotFound
@ECHO.
@ECHO OFF

REM -----
REM Anything greater than 12 is unknown, can be caused if the command
REM processor cannot find STW.EXE.
REM -----
IF %ERRORLEVEL% GTR %eMultipleProblems% (
    ECHO STW.EXE returned an unknown return code %ERRORLEVEL%
    GOTO END
)

REM -----
REM Check the specific error codes here.
REM -----
IF %ERRORLEVEL% EQU %eMultipleProblems% (
    ECHO eMultipleProblems
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptFailed% (
    ECHO eScriptFailed
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptPlaybackError% (
    ECHO eScriptPlaybackError
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptDidNotComplete% (
    ECHO eScriptDidNotComplete
    GOTO END
)

IF %ERRORLEVEL% EQU %eInvalidScriptParameters% (
    ECHO eInvalidScriptParameters
    GOTO END
)

IF %ERRORLEVEL% EQU %eRunEnvNotFound% (
    ECHO eRunEnvNotFound

```

```

    GOTO END
)
IF %ERRORLEVEL% EQU %eScriptNotFound% (
    ECHO eScriptNotFound
    GOTO END
)
IF %ERRORLEVEL% EQU %eProjectNotFound% (
    ECHO eProjectNotFound
    GOTO END
)
IF %ERRORLEVEL% EQU %eNoScriptSpecified% (
    ECHO eNoScriptSpecified
    GOTO END
)
IF %ERRORLEVEL% EQU %eLoginFailed% (
    ECHO eLoginFailed
    GOTO END
)
IF %ERRORLEVEL% EQU %eUnableToCreateComObject% (
    ECHO eUnableToCreateComObject
    GOTO END
)
IF %ERRORLEVEL% EQU %eFatalError% (
    ECHO eFatalError
    GOTO END
)
IF %ERRORLEVEL% EQU %eNoError% (
    ECHO eNoError
    GOTO END
)
IF %ERRORLEVEL% EQU %eCommandCouldNotBeProcessed% (
    ECHO eCommandCouldNotBeProcessed
    GOTO END
)
IF %ERRORLEVEL% EQU %ePrograTerminatedByUser% (
    ECHO ePrograTerminatedByUser
    GOTO END
)
IF %ERRORLEVEL% EQU %eInvalidViewportSize% (
    ECHO eInvalidViewportSize
    GOTO END
)
IF %ERRORLEVEL% EQU %eInvalidViewportParameters% (
    ECHO eInvalidViewportParameters
    GOTO END
)
:END

```

Silk Central から STW.EXE を実行するための前提条件


Silk Central 実行サーバーから STW.EXE を実行し、Silk Test Workbench データベースに正しく接続するには、次の前提条件を満たす必要があります。

1. Silk Central 実行サーバーがサービスではなく、プロセスとして実行されている。
2. Silk Test Workbench データベースへの接続にデータベース認証が使用されている。Silk Test Workbench では、Silk Test Workbench SQL Server データベースとの接続に Windows 認証とデータベース認証を使用できます。
3. データベース サーバー インスタンスが静的ポートを使用するように構成されている。

これらの前提条件が満たされていない場合、Silk Test Workbench 実行サーバーから STW.EXE を実行しようとするときに Silk Central は次のエラーで終了する場合があります。
エラー 3 : DSN、ユーザー名、パスワードのいずれかが正しくありません。

STW.EXE の使用例

コマンドラインでは、STW.EXE の 1 回の実行で任意の数のスクリプトを実行することがサポートされます。コマンドラインでは、入力ファイルと出力ファイル、およびスクリプトごとにプロジェクトと実行環境を変えることもサポートされます。

 **注:** このほとんどの例には、読みやすくするため、-d、-u、-p の各パラメータが含まれていません。

一般的なヘルプ

```
STW -help
```

単一プロジェクト、すべてのスクリプト

```
STW -project ProjectName
```

単一プロジェクト、単一スクリプト

```
STW -project ProjectName -script ScriptName
```

単一スクリプト、Common プロジェクト

```
STW -s scriptname
```

コマンドラインでは、SilkTest を DSN として、Admin をパスワードがないユーザー名として使用します。

名前にスペースが含まれるスクリプト

```
STW -script "Script with spaces"
```

Common プロジェクトの特定スクリプト

```
STW -script Script1 -script Script2
```

Common プロジェクトのすべてのスクリプト


```
STW
```

さまざまなプロジェクトの特定スクリプト

この例では2つの別々のスクリプトを実行します。別々のプロジェクトでは同じスクリプト名を使用できるので、この2つは2つの固有なスクリプトを表します。

```
STW -project Project1 -script Script1 -project Project2 -script Script1
```

Common プロジェクトとユーザー プロジェクトの特定スクリプト

 **注:** プロジェクト名を指定すると、それに続くすべてのスクリプトはそのプロジェクトに関連します。

```
STW -project Project1 -script Script1 -script Script2 -project Common -script Script1 -script Script2
```

入力ファイルの使用

```
STW -file FileA.txt
```

FileA.txt 内に以下が含まれる場合

```
# Execute my script
-script MyScript
# Change the environment and run again
-environment MyEnvironment
-script MyScript
```

環境

```
STW -environment PlaybackEnvironment -s ScriptName
```

複数の実行環境

この例では、ユーザーが定義した実行環境とデフォルトの実行環境で同一スクリプトを実行します。

```
STW -environment MyRunEnv -script Script1 -environment -script Script1
```

-environment の2回目の使用では名前を指定しません。環境はデフォルトにリセットされます。

変数

パラメータを必要とする .NET スクリプト、ビジュアルテスト、またはキーワード駆動テストをコマンドラインから実行する場合、*variable* パラメータを使用して資産のパラメータを指定できます。指定すると、資産で定義されたグローバルプロパティは上書きされます。

```
STW -script ScriptWithParameters -variable "var1=1234" "var2=Hello" "var3=True"
```

詳細モード

```
STW -u Admin -verbose -d MyDsn -s MyScript
```

以下は、詳細モードの出力例です。

```
STW.exe
Copyright (c) 1996-2011, Micro Focus IP Development Limited. All Rights Reserved

DSN                : STW-Scratch
Username           : admin
Password          :
Script(s)         : MyScript [Project3]

Append result     : False
Increment result  : False
Output file      : MyOutput.txt
Verbose Mode     : ON
```

```
Searching executable assets...
.NET Script ' MyScript' [Project3] found
Executing MyScript [Project3] ...
Script: MyScript [Project3], Status: Failed, Verifications: Total 5, Passed 0, Failed 5
```

出力ファイル

```
STW -outputfile C:¥Temp¥MyFile.txt
```

結果ディレクトリの指定

```
STW -s scriptname -resultdir C:¥Temp¥TestResults
```


または

```
STW -s scriptname -rd C:¥Temp¥TestResults
```

コマンドラインからの資産のインポートとエクスポート

Silk Test Workbench のインポートユーティリティ (STWImport.exe) およびエクスポートユーティリティ (STWExport.exe) をコマンドラインプロンプトから直接実行して、データベースの資産をインポートまたはエクスポートできます。

再利用可能な構成ファイルを作成して、繰り返し使用するパラメータ値を保存することもできます。

 **注:** Silk Test Workbench 18.5 以降を使用している場合、18.5 より前のバージョンの Silk Test Workbench でエクスポートした資産をインポートできません。このような資産をインポートする場合は、**StwConvertExportedXML** プログラムを使用してください。詳細については、「[資産のインポート](#)」を参照してください。



資産のインポートのコマンドラインパラメータ

STWImport.exe コマンドラインユーティリティを使用して、コマンドラインから資産をインポートします。インポートパラメータは全部をコマンドラインから指定することも、または特に記載のないかぎりは構成ファイルに指定することもできます。


ユーティリティは、¥Silk¥Silk Test¥ng¥gui¥ にあります。

コマンドラインインポートパラメータ


構文	値	必須かどうか	説明
Username (-u)	ユーザー定義	はい	接続に使用されるユーザー名。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されません。Windows 認証を使用する場合は、ユーザー名は省略します。
Password (-p)	ユーザー定義	はい	接続に使用されるパスワード名。このパラメータはパスワードが Nothing 以外の場合に必要です。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されません。Windows 認証を使用する場合は、パスワードは省略します。

構文	値	必須かどうか	説明
DSN (-d)	ユーザー定義	はい	データベースへの接続に使用する DSN の名前。このパラメータは、「SilkTest」以外の DSN 名の場合に必要です。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。
ImportFile (-if)	ユーザー定義		インポートする ZIP ファイルのファイル名または場所。
Execute (-ex)	Yes No (下記の注意を参照)	はい	インポート操作をコミットします。デフォルト値は、「No」です。
Filename (-f)	ユーザー定義	いいえ	<p>コマンドラインパラメータを格納する INI 構成ファイルです。コマンドラインでのみ指定できます。</p> <p> 注: INI ファイルパラメータはコマンドラインパラメータを上書きしますが、例外があります。INI ファイルで指定された Projects (-prj) および System (-s) パラメータの値は、コマンドライン上に指定された同じパラメータを上書きしません。それぞれの値が両方とも使われます。</p>
AllVersions (-av)	Yes No (下記の注意を参照)	いいえ	資産のすべてのバージョンをインポートします。デフォルト値は、「No」です。
ReportResult (-rr)	ユーザー定義	いいえ	作成される要約レポート ファイルの名前と場所が含まれています。デフォルトの場所は現在のディレクトリです。
SilentMode (-sm)	Yes No (下記の注意を参照)	いいえ	コンソール出力の表示を抑止します。
Projectmapping (-pm)	ユーザー定義	いいえ	インポート ファイルの元のプロジェクトの代わりに使用する対象のプロジェクトを指定します。縦線文字 () で区切って複数のマッピングを指定できます。
Projects (-prj)	資産のノード	いいえ	<p>インポートする単一のプロジェクト資産またはプロジェクト資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべての資産がインポートされます。ノードのグループは縦線文字 () で区切られます。形式は、<code>{ProjectName} . {AssetType} . {AssetName} . {Ver#}</code> です。例：</p> <p style="text-align: center;">-prj ProjectA.Script.SampleScript.1</p> <p>縦線文字で区切る文字列は、二重引用符で囲む必要があります。例：</p> <p style="text-align: center;">-prj "Common ProjectB.Result"</p> <p> 注: 資産の種類を指定する際は、単数形を使用してください。たとえば、「results」ではなく、「result」を使用します。資産名が 2 語で構成される場合 (visual test) は、スペースを削除してください (visualtest)。ただし、テストスクリプトだけは例外です。テストスクリプトを指定する際は、「script」という単語のみを使用してください。</p>

構文	値	必須かどうか	説明
System (-s)	資産のノード	いいえ	<p>インポートする単一のシステム資産またはシステム資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべてのシステム資産がインポートされます。ノードのグループは縦線文字 () で区切られます。形式は、{SystemName}・{SystemType}・{SystemAssetName} です。例： -s Options.Record.RecA</p> <p>縦線文字で区切る文字列は、二重引用符で囲む必要があります。</p>
ProjectAssetResolve-Conflict (-parc)	s a p	いいえ	<p>データベースにすでに存在しているプロジェクト資産について、競合の解決方法を指定します。</p> <ul style="list-style-type: none"> • s を指定すると、インポートが中止されます。 • a を指定すると、インポート時に資産にバージョンが追加されます。 • p を指定した場合、競合する資産それぞれに対して、ユーザーに操作を選択するよう促すプロンプトを表示し、SilentMode パラメータを上書きします。デフォルトは p です。
SystemAssetResolve-Conflict (-sarc)	s p	いいえ	<p>データベースにすでに存在しているシステム資産について、競合の解決方法を指定します。</p> <ul style="list-style-type: none"> • s を指定すると、インポートが中止されます。 • p を指定した場合、競合する資産それぞれに対して、ユーザーに操作を選択するよう促すプロンプトを表示し、SilentMode パラメータを上書きします。デフォルトは「p」です。

 **注:** コマンドラインパラメータのデフォルト値を指定するには、コマンドでパラメータの指定を省略します。デフォルト以外の値を指定するには、値なしでパラメータだけを含めます。たとえば、資産のすべてのバージョンをインポートするには、インポートコマンドで AllVersions (-av) パラメータを値「Yes」なしで指定します。資産の最新バージョンだけをインポートするには、単にこのパラメータを省略するとデフォルト値の「No」が適用されます。

インポート構成ファイルでパラメータ値を指定するとき、パラメータまたはパラメータ値を省略すると、デフォルト値が使用されます。デフォルト以外の値の場合は、値を明示的に指定する必要があります。たとえば、資産のすべてのバージョンをインポートするには、AllVersions (-av) パラメータを値の「Yes」と共に指定します。

 **注:** Silk Test Workbench 18.5 以降を使用している場合、18.5 より前のバージョンの Silk Test Workbench でエクスポートした資産をインポートできません。このような資産をインポートする場合は、**StwConvertExportedXML** プログラムを使用してください。詳細については、「[資産のインポート](#)」を参照してください。

例

```
C:/>STWImport.exe -u Admin -p admin -d MyDBName -mt sf -mn "c:¥MySTAssetFiles" -prj ProjectA -pm "ProjectA=ProjectB" -rr "c:¥STImport.txt" -ex
```

上記の例では、MySTAssetFiles ディレクトリに含まれる ProjectA の資産ファイルの最新バージョンが、MYDBName データベースの ProjectB にインポートされ、要約レポートを含むファイルが作成されます。

構成ファイルの例

```
C:/>STWImport.exe -u Admin -p admin -d MyDBName -f "C:¥STImport.ini"
```

上記の例には、コマンドライン上で指定する必要のあるインポートパラメータだけが含まれています。その他のすべてのパラメータは構成ファイルに保存されます。

コマンドラインインポート構成ファイルの作成


設定を永続的に保存するには、コマンドラインインポート構成ファイルを作成します。インポートパラメータの大部分は、このファイルに保存できます。Username、Password、DSN、Filenameなどのいくつかのパラメータは、コマンドライン上で指定する必要があります。

1. オプションのインポートパラメータをすべて含めるための .INI ファイル (たとえば STImport.ini) を作成します。
2. ファイルを必要な場所に保存します。
3. 以下のインポートパラメータのリストをコピーして、構成ファイルに貼り付けます。

```
[SilkTestImport]
Execute=
ImportFile=
AllVersions=
ReportResult=
SilentMode=
Projects=
System=
ProjectAssetResolveConflict=
SystemAssetResolveConflict=
```

4. 各インポートパラメータの値を設定します。値は引用符で囲む必要はありません。値が設定されない場合、またはファイルでこのパラメータが省略された場合は、デフォルト値が使用されます。パラメータ値では大文字と小文字は区別されません。
5. ファイルを保存します。
6. コマンドラインからファイルにアクセスするには、Filename (-f) パラメータとファイルの場所を指定します。例：

```
C:¥>STWImport.exe -u Admin -p admin -d YourDBName -f "C:¥STImport.ini"
```



 **注:** 構成ファイルの見出し [SilkTestImport] は、ファイルで最初に指定されている必要がありますが、パラメータの順番は変更できます。パラメータをコメントアウトする必要がある場合、パラメータ名の前に円記号 (¥) を入力します。構成ファイルに指定されたパラメータは、コマンドライン上に指定されている同じパラメータを上書きしますが、例外が 1 つあります。構成ファイル内とコマンドライン上の両方に Projects または System パラメータが指定されている場合、それぞれの値が使用されます。

資産のエクスポートのコマンドラインパラメータ


STWExport.exe コマンドラインユーティリティを使用して、コマンドラインから資産をエクスポートします。エクスポートパラメータは全部をコマンドラインから指定することも、または特に記載のないかぎりは構成ファイルに指定することもできます。

ユーティリティは、¥Silk¥Silk Test¥ng¥gui¥ にあります。

コマンドライン エクスポート パラメータ

構文	値	必須かどうか	説明
Username (-u)	ユーザー定義	はい	接続に使用されるユーザー名。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。Windows 認証を使用する場合は、ユーザー名は省略します。
Password (-p)	ユーザー定義	はい	接続に使用されるパスワード名。このパラメータはパスワードが Nothing 以外の場合に必要です。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。Windows 認証を使用する場合は、パスワードは省略します。
DSN (-d)	ユーザー定義	はい	データベースへの接続に使用する DSN の名前。このパラメータは、「SilkTest」以外の DSN 名の場合に必要です。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。
ExportFile (-ef)	ユーザー定義	はい	エクスポートする ZIP ファイルのファイル名または場所。
Execute (-ex)	Yes No (下記の注意を参照)	はい	エクスポート操作をコミットします。デフォルト値は、「No」です。
Filename (-f)	ユーザー定義	いいえ	<p>コマンドラインパラメータを格納する INI 構成ファイルです。コマンドラインでのみ指定できます。</p> <p> 注: INI ファイルパラメータはコマンドラインパラメータを上書きしますが、例外があります。INI ファイルで指定された Projects (-prj) および System (-s) パラメータの値は、コマンドライン上に指定された同じパラメータを上書きしません。それぞれの値が両方とも使われます。</p>
AllVersions (-av)	Yes No (下記の注意を参照)	いいえ	資産のすべてのバージョンをエクスポートします。デフォルト値は、「No」です。
ReportResult (-rr)	ユーザー定義	いいえ	作成される要約レポート ファイルの名前と場所が含まれています。デフォルトの場所は現在のディレクトリです。
SilentMode (-sm)	Yes No (下記の注意を参照)	いいえ	コンソール出力の表示を抑止します。
Projects (-prj)	資産のノード	いいえ	<p>エクスポートする単一のプロジェクト資産またはプロジェクト資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべての資産がエクスポートされます。ノードのグループは縦線文字 () で区切られます。形式は、{ProjectName}. {AssetType}. {AssetName}. {Ver#} です。例: -prj ProjectA.Visualtest.SampleTest.1。縦線文字 () で区切る文字列は、二重引用符で囲む必要があります。例: -prj "Common ProjectB.Result"。</p> <p> 注: 資産の種類を指定する際は、単数形を使用してください。たとえば、「results」ではなく、</p>

構文	値	必須かどうか	説明
			「result」を使用します。資産名が 2 語で構成される場合 (visual test) は、スペースを削除してください (visualtest)。ただし、テストスクリプトだけは例外です。テストスクリプトを指定する際は、「script」という単語のみを使用してください。
System (-s)	資産のノード	いいえ	エクスポートする単一のプロジェクト資産またはプロジェクト資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべてのシステム資産がエクスポートされます。ノードのグループは縦線文字 () で区切られます。形式は、{SystemName}、{SystemType}、{SystemAssetName} です。例：-s Options.Record.RecA。縦線文字 () で区切る文字列は、二重引用符で囲む必要があります。
ExportAssociated (-ea)	Yes No (下記の注意を参照)	いいえ	データベースのすべてのプロジェクトから、参照されるすべての資産がエクスポートされるようにします。デフォルト値は、「No」です。
OverwriteFile (-of)	Yes No (下記の注意を参照)	いいえ	エクスポート時にファイル名が競合した場合、既存のファイル (読み取り専用ファイルを含む) を上書きします。デフォルト値は、「Yes」です。

 **注:** コマンドラインパラメータのデフォルト値を指定するには、エクスポートコマンドでパラメータの指定を省略します。デフォルト以外の値を指定するには、値なしでパラメータだけを含めます。たとえば、資産のすべてのバージョンをエクスポートするには、エクスポートコマンドで AllVersions (-av) パラメータを値「Yes」なしで指定します。資産の最新バージョンだけをエクスポートするには、単にこのパラメータを省略するとデフォルト値の「No」が適用されます。

エクスポート構成ファイルでパラメータ値を指定するとき、パラメータまたはパラメータ値を省略すると、デフォルト値が使用されます。デフォルト以外の値の場合は、値を明示的に指定する必要があります。たとえば、資産のすべてのバージョンをエクスポートするには、AllVersions (-av) パラメータを値「Yes」付きで指定します。

例

```
C:/>STWExport.exe -u Admin -p admin -d MyDBName -mn "c:¥MySTAssetFiles" -mt ds -rr "c:¥STExport.txt" -prj ProjectA -ex
```

上記の例では、ProjectA の資産ファイルの最新バージョンが C:¥MySTAssetFiles にエクスポートされ、結果の要約レポートを含むファイルが作成されます。

構成ファイルの例

```
C:¥>STWExport.exe -u Admin -p admin -d YourDBName -f "C:¥STExport.ini"
```

コマンドラインエクスポート構成ファイルの作成

エクスポート設定を永続的に保存するには、コマンドラインエクスポート構成ファイルを作成します。エクスポートパラメータの大部分は、このファイルに保存できます。Username、Password、DSN、Filename などのいくつかのパラメータは、コマンドライン上で指定する必要があります。

1. オプションのエクスポートパラメータをすべて含めるための .INI ファイル (たとえば STExport.ini) を作成します。

2. ファイルを必要な場所に保存します。
3. 以下のエクスポートパラメータのリストをコピーして、構成ファイルに貼り付けます。

```
[SilkTestExport]
Execute=
ExportFile=
AllVersions=
ReportResult=
SilentMode=
Projects=
System=
ExportAssociated=
OverwriteFile=
```

4. 各エクスポートパラメータの値を設定します。値は引用符で囲む必要はありません。値が設定されていない場合は、デフォルトが使用されます。パラメータ値では大文字と小文字は区別されません。
5. ファイルを保存します。
6. コマンドラインからファイルにアクセスするには、Filename (-f) パラメータとファイルの場所を指定します。例：

```
C:¥>STWExport.exe -u Admin -p admin -d YourDBName -f "C:¥STExport.ini"
```



注： 構成ファイルの見出し [SilkTestExport] は、ファイルで最初に指定されている必要がありますが、パラメータの順番は変更できます。パラメータをコメントアウトする必要がある場合、パラメータ名の前に円記号 (¥) を入力します。構成ファイルに指定されたパラメータは、コマンドライン上に指定されている同じパラメータを上書きしますが、例外が 1 つあります。構成ファイル内とコマンドライン上の両方に Projects または System パラメータが指定されている場合、それぞれの値が使用されます。

ログ

コマンドラインからの実行中に、Silk Test Workbench は、%LOCALAPPDATA%¥silk¥silktest¥logs ディレクトリにある .log ファイルにイベントを記録します。デフォルトでは、ログレベルは情報モードに設定されており、コマンドラインからの実行ごとに、PID と日時を使った名前の新しいログファイルが作成されます。同じプログラムを複数回実行すると、古いファイルは削除され、新しいファイルが作成されません。

次の表には、コマンドライン実行可能ファイルに対応するファイル名のサンプルを示します。

プログラム	サンプル ファイル名
SilkTest.exe	SilkTest(Pid=25184)_20171030_1635.log
STW.exe	STW(Pid=13204)_20171031_0839.log
STWDBConfig.exe	STWDBConfig(Pid=32924)_20171030_1635.log
STWExport.exe	STWExport(Pid=32560)_20171030_1635.log
STWImport.exe	STWImport(Pid=26640)_20171030_1635.log
STWMaint.exe	STWMaint(Pid=13488)_20171030_1618.log
STWPurge.exe	STWPurge(Pid=30096)_20171030_1635.log

ログの設定は、Silk Test Workbench インストールの gui フォルダにある workbench.properties ファイルを変更して行います。デフォルトでは、このフォルダは C:¥Program Files (x86)¥Silk¥SilkTest ¥ng¥gui になります。このファイルでは、ログレベル、ログ形式、ログディレクトリなどを指定できます。

たとえば、1 つのプログラムに対して複数のログファイルを作成する必要がある場合は、nativeLog.clearPerProcessLogs を 0 に設定すると、ファイルが削除されなくなります。

また、`nativeLog.LogPerProcess` を 0 に設定すると、ファイルの作成方法を変更できます。この場合、すべての実行可能ファイルに対して、`silktest.log` という名前のファイルにログが記録されるようになります。



注: `nativeLog.OverWriteExisting` を 0 に設定しない限り、実行するたびにファイルが作成しなおされます。

Silk Test Workbench のオプション

テスト環境とテスト方法を反映するには、デフォルト オプションを変更します。オプションを変更するには、**ツール > オプション** をクリックします。

グローバル オプション


すべてのプロジェクトのプロパティを設定するには、**グローバル** オプションを使用します。

グローバル オプションは、以下のとおりです。

資産管理	デフォルトの保存動作	ツールバーのボタン、資産のダイアログ ボックス、またはショートカット キーの組み合わせ (Ctrl+S) で資産を保存するときの、デフォルトの保存動作を設定します。動作は、資産の新しいバージョンを作成する、または資産の現在のバージョンを置き換える、のいずれかから選択します。
	資産のバージョンの最大数	保持する資産のバージョンの最大数。0 を指定すると、すべてのバージョンが保持されます。バージョンの最大数に到達すると、資産の新しいバージョンが保存され、最も古いバージョンが削除されます。資産をインポートする場合、バージョンの数が最大数を超えていたとしても、すべてのバージョンがインポートされます。このようにしてインポートされた資産に対して新しいバージョンが保存されるとき、指定した資産の最大数と一致するように、最も古いバージョンの資産から削除されます。
	ページの実行に必要なアクセス権	プロジェクトの資産をパーズするために最低限必要なユーザー アクセス権を設定します。デフォルト値は、 スクリプト作成者 です。 フル アクセス も選択できます。このオプションは、管理者権を持つ Silk Test Workbench ユーザーだけが変更できます。現在のユーザーがパーズを実行できるアクセス権を持つプロジェクトがない場合、Silk Test Workbench メニューの ファイル > 資産のバージョンをパーズ... 項目は無効になります。それ以外の場合、 資産のバージョンをパーズ ダイアログの プロジェクト ツリーに、ユーザーが資産をパーズできるプロジェクトが表示されます。
	[資産のバージョンをパーズ]のプロジェクトの選択状態を保持	はい に設定すると、 資産のバージョンをパーズ ダイアログ ボックスで、前回実行時のプロジェクトの選択状態が保持されます。 いいえ に設定すると、ダイアログ ボックスを開いたときに、プロジェクトは選択されません。

eCATT

ツール > オプション > グローバル > 統合

 **注:** これらのオプションは、Silk Test Workbench が eCATT と統合されている場合にのみ利用できません。

RFC のトレース Silk Test Workbench の RFC 呼び出しの診断トレースを有効にするには **オン** に設定します。診断トレースは、トレース ファイルに保存されます。



注: RFC は、librfc32.dll という SAP .dll に実装されている SAP API です。トレース ファイルはこの .dll 内のコードによって作成され、トレース ファイルの保存場所もこのコードによって設定されます。デフォルトの場所はシステム フォルダです。

RFC トレース ファイルをダンプするパスの Windows 環境変数を設定します。変数は `RFC_TRACE_DIR` で、その値はパスにする必要があります。これについては、SAP が提供している RFCSDK のヘルプで説明されています。

全般オプション

ツール > オプション > 全般

全般オプションは、以下のとおりです。

デフォルトの結果表示	Silk Test Workbench によって 結果 ウィンドウに表示されるタブを選択します。
デフォルトの待機タイムアウト (ミリ秒)	Silk Test Workbench がオブジェクトを待機する期間を指定します。デフォルトは 5000 です。
記録時にフラッシュする	テストの記録中に、Silk Test Workbench がタスクバーの Silk Test Workbench ウィンドウをフラッシュするかどうかを指定します。
検索時に最小化	検索 ボタンをクリックしたときに、Silk Test Workbench が Silk Test Workbench ウィンドウを最小化するかどうかを指定します。
再生時に最小化	テストの再生中に、Silk Test Workbench が Silk Test Workbench ウィンドウを最小化するかどうかを指定します。
記録時に最小化	テストの記録中に、Silk Test Workbench が Silk Test Workbench ウィンドウを最小化するかどうかを指定します。
結果を自動的に開く	テストの再生後に Silk Test Workbench が結果を自動的に表示するかどうかを指定します。
再生時にすべて保存	テストの再生中に、Silk Test Workbench がすべてのファイルを保存するかどうかを指定します。
ブラウザー構成ダイアログの表示	資産にブラウザー構成がある場合にブラウザーを選択できるように、Silk Test Workbench が ブラウザー構成 ダイアログを表示するかどうかを指定します。この値を いいえ に設定すると、Silk Test Workbench はアプリケーション構成で指定したブラウザーの種類を使用します。
再生中にローカル変数ウィンドウを表示する	再生中に Silk Test Workbench が ローカル変数 ウィンドウを表示し、再生終了時に非表示にするかどうかを指定します。
[テスト ロジック デザイナー] の [ロジック ステップ] ページを表示する	Silk Test Workbench が テスト ロジック デザイナー ウィザードに ロジック ステップ ペインを表示するかどうかを指定します。ウィザードの ロジックのステップ ペインのタイトルの横にある矢印ボタンをクリックして、このペインの表示/非表示を切り替えることもできます。
モバイル構成ダイアログの表示	モバイル デバイス アプリケーション構成を使用した記録、再生、または、ビジュアル テストまたは .NET スクリプトの識別中に モバイル デバイス構成 ダイアログを表示するかどうかを指定します。
[再生完了] ダイアログの表示	テスト再生が終了したときに、Silk Test Workbench が 再生完了 ダイアログ ボックスを表示するかどうかを指定します。
[再生] ダイアログの表示	テストを開始する前に、Silk Test Workbench が 再生 ダイアログ ボックスを表示するかどうかを指定します。

[記録完了] ダイアログの表示	Silk Test Workbench が 記録完了 ダイアログ ボックスを表示するかどうかを指定します。
[状態の記録] ダイアログの表示	記録を開始するときに、Silk Test Workbench が 状態の記録 ダイアログ ボックスを表示するかどうかを指定します。
[テスト ロジック デザイナ] の [要約] ページを表示する	Silk Test Workbench が テスト ロジック デザイナ ウィザードに 要約 ペインを表示するかどうかを指定します。この設定は、 テスト ロジック デザイナ ウィザードの 要約 ペインで変更することもできます。
ウィザードの [ようこそ] 画面の表示	Silk Test Workbench のウィザードごとに ようこそ ページを表示するかどうかを指定します。この設定は、各ウィザードの ようこそ ページで変更することもできます。

全般オプションの変更

全般 オプションは、起動やビジュアル テストまたはスクリプトの記録および再生を行う際の全般的な動作を制御します。


1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーで **全般** をクリックします。全般オプションが右側のパネルに表示されます。
3. 最初の列でオプションの名前を選択し、2 番目の列で新しい値を選択または入力して、オプションを変更します。
オプションを選択すると、簡単な説明が下のパネルに表示されます。
4. Common プロジェクトに追加するすべての資産に接頭辞 *Common* を追加するには、**Common プロジェクト資産に接頭辞を追加する** リスト ボックスから **資産名と 'Common' を使用する** を選択します。
5. **デフォルトの結果表示** リスト ボックスから、**結果** ウィンドウに表示するタブを選択します。
6. Silk Test Workbench がオブジェクトを待機する時間を指定するには、使用する時間を **デフォルトの待機タイムアウト (ミリ秒)** テキスト ボックスに入力します。
7. テストの記録中にタスクバーの Silk Test Workbench ウィンドウをフラッシュするには、**記録時にフラッシュする** リスト ボックスから **はい** を選択します。
8. **検索** ボタンをクリックしたときに Silk Test Workbench ウィンドウを最小化するには、**検索時に最小化** リスト ボックスから **はい** を選択します。
9. テストの再生時に Silk Test Workbench ウィンドウを最小化するには、**再生時に最小化** リスト ボックスから **はい** を選択します。
10. テストの記録時に Silk Test Workbench ウィンドウを最小化するには、**記録時に最小化** リスト ボックスから **はい** を選択します。
11. テストの再生後に結果を自動的に表示するには、**結果を自動的に開く** リスト ボックスから **はい** を選択します。
12. テストの再生時にすべてのファイルを保存するには、**再生時にすべて保存** リスト ボックスから **はい** を選択します。
13. 資産にブラウザ構成があるときに、ブラウザを選択するために、**ブラウザ構成** ダイアログを開くには、**ブラウザ構成ダイアログの表示** リスト ボックスから **はい** を選択します。
14. 再生中に **ローカル変数** ウィンドウを表示して、再生が終わったら非表示にするには、**ローカル変数ウィンドウの表示** リスト ボックスから **はい** を選択します。
15. **テスト ロジック デザイナ** ウィザードで **ロジック ステップ** ペインを表示するには、**[テスト ロジック デザイナ] の [ロジック ステップ] ページを表示する** リスト ボックスから **はい** を選択します。
ウィザードの **ロジックのステップ** ペインのタイトルの横にある矢印ボタンをクリックして、このペインの表示/非表示を切り替えることもできます。
16. テスト再生の終了時に **再生完了** ダイアログを表示するには、**[再生完了] ダイアログの表示** リスト ボックスから **はい** を選択します。

- 17テストの開始前に **再生** ダイアログを表示するには、**[再生] ダイアログの表示** リスト ボックスから **はい** を選択します。
再生 ダイアログ ボックスでは、結果名、説明、およびオプション ファイルの詳細を割り当てることができます。
- 18テストの開始前に **記録完了** ダイアログを表示するには、**[記録完了] ダイアログの表示** リスト ボックスから **はい** を選択します。
記録完了 ダイアログ ボックスでは、テストの再生、ビジュアル テストへの移動、またはテストの保存を行うことができます。
- 19記録の開始時に **状態の記録** ダイアログ ボックスを表示するには、**[状態の記録] ダイアログの表示** リスト ボックスから **はい** を選択します。
- 20テスト **ロジック デザイナ** ウィザードで **要約** ペインを表示するには、**[テスト ロジック デザイナ] の [要約] ページを表示する** リスト ボックスから **はい** を選択します。
この設定は、**テスト ロジック デザイナ** ウィザードの **要約** ページで変更することもできます。
- 21各ウィザードで [ようこそ] ページを表示するには、**ウィザードの [ようこそ] 画面の表示** リスト ボックスから **はい** を選択します。
この設定は、各ウィザードの **ようこそ** 画面で変更することもできます。
- 22OK をクリックします。

開始画面オプション


ツール > オプション をクリックしてから **開始画面** をクリックします。

このオプションを使用し、**開始画面** の設定を制御します。

 **注:** いつでも Ctrl+Alt+S を押すか、または **表示 > 開始画面** をクリックして、開始画面を表示できます。

Silk Test Workbench の開始時に開始画面を表示する	Silk Test Workbench の開始時に 開始画面 を表示するには、 はい に設定します。
[最近使った資産] に表示する数	最近使った資産 リストに表示する項目の数を選択します。

フラグ **割り当て先** 選択されたユーザーに割り当てられているフラグを表示するには、このオプションを使用します。現在のユーザーに割り当てられているフラグを表示するには、**現在のユーザー** に設定します。プロジェクト管理者に割り当てられているフラグを表示するには、**Admin** に設定します。このオプションを **Admin** に設定するには、管理者権限が必要です。

 **注:** ユーザーがシステムから削除されると、そのユーザーの名前は **割り当て先** リストから削除され、そのユーザーに割り当てられていたフラグはプロジェクト管理者に再割り当てされます。

最終更新者 すべてのユーザーによって変更されたフラグを表示するか、または特定のユーザーによって変更されたフラグのみを表示するには、このオプションを設定します。現在のデータベースでビジュアル テストに対してフラグを最後に入力または変更したユーザーがリストに表示されます。

日付順 更新日時が最も古いフラグをリストの最上部に表示するには、**昇順** に設定します。更新日時が最も新しいフラグをリストの最上部に表示するには、**降順** に設定します。

プロジェクト 選択されたプロジェクトで作成されたフラグのみを表示するには、このオプションを使用します。デフォルト設定では、すべてのプロジェクトのフラグが表示されるようになっています。Common プロジェクトまたは別のプロジェクトをリストから選択します。

作成したフラグの表示 ビジュアルテストで作成されたフラグ、結果で作成されたフラグ、またはビジュアルテストと結果の両方で作成されたフラグを表示するよう、リストからオプションを選択します。

開始画面オプションの変更

開始画面 オプションを使用して、Silk Test Workbench が起動したときに **開始画面** を非表示にします。また、**開始画面** の **フラグ** ペインで、フラグの表示方法を指定することもできます。また、いつでも Ctrl+Alt+S を押すか、または **表示** > **開始画面** を選択して、開始画面を表示できます。

1. ツール > **オプション** をクリックします。
2. **オプション** メニュー ツリーの **開始画面** の隣にあるプラス記号 (+) をクリックします。**開始画面** オプションが右側のパネルに表示され、**フラグ** サブカテゴリが **開始画面** の下のアウトライン ビューに表示されます。**Silk Test Workbench** の開始時に**開始画面を表示する** を **はい** に設定して、Silk Test Workbench の開始時に **開始画面** を表示できます。**開始画面** では、ユーザーは、ビジュアルテストを記録し、開いてクイックテストを実行する、管理タスク、テストの記録タスク、編集タスクを実行する、フラグを使用して、プロジェクト協業を実施する、アプリケーション テストを迅速に作成するためのアシスタンスを利用するといった作業が可能です。
3. アウトライン ビューで、**フラグ** をクリックしてオプションを設定します。
4. 環境に適したオプションを設定します。

割り当て先 選択されたユーザーに割り当てられているフラグを表示するには、このオプションを使用します。現在のユーザーに割り当てられているフラグを表示するには、**現在のユーザー** に設定します。プロジェクト管理者に割り当てられているフラグを表示するには、**Admin** に設定します。このオプションを **Admin** に設定するには、管理者権限が必要です。



注: ユーザーがシステムから削除されると、そのユーザーの名前は **割り当て先** リストから削除され、そのユーザーに割り当てられていたフラグはプロジェクト管理者に再割り当てされます。

最終更新者 すべてのユーザーによって変更されたフラグを表示するか、または特定のユーザーによって変更されたフラグのみを表示するには、このオプションを設定します。現在のデータベースでビジュアルテストに対してフラグを最後に入力または変更したユーザーがリストに表示されます。

日付順 更新日時が最も古いフラグをリストの最上部に表示するには、**昇順** に設定します。更新日時が最も新しいフラグをリストの最上部に表示するには、**降順** に設定します。

プロジェクト 選択されたプロジェクトで作成されたフラグのみを表示するには、このオプションを使用します。デフォルト設定では、すべてのプロジェクトのフラグが表示されるようになっています。Common プロジェクトまたは別のプロジェクトをリストから選択します。

作成したフラグの表示 ビジュアルテストで作成されたフラグ、結果で作成されたフラグ、またはビジュアルテストと結果の両方で作成されたフラグを表示するよう、リストからオプションを選択します。

5. **OK** をクリックします。

記録オプション

記録オプションは、以下のカテゴリに分けられます。

全般	資産とマウスの移動の設定、オブジェクト マップの場所、無視するクラス、および、記録時にテスト対象アプリケーション (AUT) のサイズを変更して、AUT の隣に Silk Recorder を表示するかを設定します。
ロケーター	記録を一時停止するためのホット キーの組み合わせ、およびロケーターの記録に使用するモードを指定します。
出力	記録中にコントロールをキャプチャするかどうか、プロパティを記録するかどうか、画面のキャプチャ方法、個別の画面をキャプチャするかどうかを指定します。
ホット キー	どのキーの組み合わせが記録中に検証を挿入するか、または記録を開始および停止するかを指定します。
xBrowser	カスタム属性、記録中に無視するブラウザ属性、DOM 関数の代わりに、ユーザーの入力そのものを記録するかどうかを指定します。
WPF	カスタム属性、および記録や再生の対象にする WPF クラスの名前を指定します。
Rumba	画面検証を作成するときに無視する Rumba 画面上のオブジェクトを指定します。... をクリックして、 除外オブジェクトの編集 ダイアログ ボックスを開きます。最初の行にある + を押すと、新しい除外オブジェクトが追加されます。各オブジェクトの行 ("オブジェクト 1"、"オブジェクト 2" など) にある x ボタンを使用すると、個々の除外オブジェクトを削除できます。この設定では、画面検証が記録されるたびに、設定内の除外オブジェクトがその画面検証に設定されます。個々の画面検証内の除外オブジェクトは手動で編集できますが、これによって、(必要に応じて、個々の検証に合わせて除外オブジェクトが調整されるようにして) 設定が更新されることはありません。
JavaSWT	ロケーター生成中に使用するカスタム属性を指定します。
SAP	ロケーター生成中に使用するカスタム属性を指定します。
Swing	ロケーター生成中に使用するカスタム属性を指定します。
Win32	ロケーター生成中に使用するカスタム属性を指定します。
WinForms	ロケーター生成中に使用するカスタム属性を指定します。
Open Agent の設定	エージェントに直接送信されるカスタム設定を行います。サポートに勧められた場合など、非常に特殊なテスト シナリオでのみ設定されるべきです。

イメージ資産記録設定の設定

イメージのクリックまたは画面検証を記録するかどうか、およびイメージ資産の精度レベルにデフォルト値を設定するかどうかを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般** をクリックします。
4. **イメージ資産のデフォルト精度レベル** リスト ボックスから、1 (低精度) から 10 (高精度) までの値を選択し、新しいイメージ資産のデフォルト精度レベルを設定します。
5. オブジェクト クリックを記録するには、**'ImageClick' を記録する** リスト ボックスから **はい** を選択します。
6. 画面検証を記録するには、**画面検証を記録する** リスト ボックスから **はい** を選択します。
7. **OK** をクリックします。

オブジェクト マップの詳細設定の設定

オブジェクト マップを記録するかどうか、オブジェクト マップを記録するプロジェクト、ロックされたオブジェクト マップの処理方法を指定します。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。 記録 オプション が右側のパネルに表示されます。
3. ロケーター をクリックします。
4. オブジェクト マップ エントリを記録するか、XPath ロケーターを記録するかを定義するには、 **オブジェクト マップ エントリを記録する** リストから適切な記録モードを選択します。
 - **オブジェクト マップ エントリ (新しいオブジェクトと既存のオブジェクト)**。これはデフォルトのモードです。
 - **XPath ロケーター (新しいオブジェクトと既存のオブジェクト)**。
 - **XPath ロケーター (新しいオブジェクトのみ)**。オブジェクト マップに既に存在するオブジェクトに対しては、オブジェクト マップ エントリが再利用されます。この設定を選択すると、AUT のメインコントロールに対するオブジェクト マップを作成し、AUT に対して追加のテストを作成する間、これらのオブジェクト マップを保持することができます。
5. オプション メニュー ツリーで **全般** をクリックします。
6. ロックされたオブジェクト マップのロケーターを記録するには、 **ロック済みのオブジェクト マップに対してロケーターを記録する** リスト ボックスから **はい** を選択します。

はい を選択した場合、オブジェクト マップがロックされていると、以前に識別されたオブジェクト マップ項目が適用可能な場合に使用され、新しいオブジェクトを記録する場合にはロケーターが使用されます。

いいえ を選択した場合、記録する前に、オブジェクト マップがロック解除されるのを待機する必要があります。
7. Silk Test Workbench がオブジェクト マップ項目を検索し、一致した項目が見つからない場合に項目を作成するプロジェクトを指定するには、 **オブジェクト マップの場所** リスト ボックスからオプションを選択します。

デフォルトでは、Silk Test Workbench は現在のプロジェクトを検索して、記録時に使用する一致したオブジェクト マップ項目が決まります。一致する項目が見つからない場合、Silk Test Workbench はすべての参照プロジェクトを検索します。一致する項目が参照プロジェクトで見つかり、その項目が使用されます。一致する項目が参照プロジェクトで見つからない場合、Silk Test Workbench によって、現在のプロジェクトに新しい項目が作成されます。

現在のプロジェクト を選択した場合、オブジェクト マップ項目は、ビジュアル テストまたはスクリプトが格納されているプロジェクトを使用します。
8. **OK** をクリックします。

マウス移動の詳細設定

マウス移動イベントを使用する Web アプリケーション、Win32 アプリケーション、および Windows Forms アプリケーションでマウス移動操作を記録するかどうかを指定します。たとえば、Apache Flex や Swing など、xBrowser テクノロジ ドメインの子ドメインのマウス移動イベントを記録することはできません。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。 記録 オプション が右側のパネルに表示されます。
3. **全般** をクリックします。
4. マウス移動操作を記録するには、 **マウスの移動操作を記録する** リスト ボックスから **はい** を選択します。

マウスの移動イベントを使用する Web ページをテストする場合は、 **はい** を選択します。

Silk Test Workbench では、スクリプトを短くするために、マウスが置かれた要素またはその親が変化するマウスの移動イベントのみが記録されます。
5. マウスの移動操作を記録する場合、MoveMouse 操作が記録される前に、どのくらいの間マウスが不動状態になければならないかを、 **マウスの移動記録遅延** テキスト ボックスにミリ秒単位で指定します。デフォルト値は、200 に設定されています。

マウスの移動操作は、この時間、マウスが静止している場合にのみ記録されます。遅延を短くすると、予期しないマウスの移動操作が増加します。遅延を長くすると、操作を記録するためにマウスを静止しておく必要があります。

6. **OK** をクリックします。

無視するクラスの設定

オブジェクト階層を単純化し、テストスクリプトや関数のコードの行の長さを短くするために、次のテクノロジーの確実に不要なクラスに対するコントロールを抑制できます。

- Win32
- Java AWT/Swing
- Java SWT/Eclipse
- Windows Presentation Foundation (WPF)

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般** をクリックします。
4. **無視するクラス** グリッドで、記録や再生中に無視するクラスの名前を入力します。
複数のクラス名を指定する場合にはカンマで区切ります。
5. **OK** をクリックします。

ロケータの詳細設定の設定

記録を一時停止するためのホット キーの組み合わせ、およびロケータの記録に使用するモードを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **ロケータ** をクリックします。
4. 記録を一時停止するために使用するホット キーの組み合わせを設定するには、**ロケータ識別ホットキー** リスト ボックスからキーの組み合わせを選択します。
デフォルトのホット キーの組み合わせは、Ctrl+Shift です。ただし、Silk Test Classic のユーザーは Ctrl+Alt を使用することを好む場合があります。
5. **ロケータ識別モード** リスト ボックスから、ロケータの記録に使用するモードを選択します。
 - **クリック** : 記録するオブジェクトをクリックで識別するには、このモードを使用します。デフォルトでは、最適なパフォーマンスを提供するこのモードが、Silk Test Workbench がロケータをキャプチャする際に使用されます。ただし、ロケータをキャプチャするときにアプリケーションは停止しません。この結果、このモードはテスト対象アプリケーションの画面の変化が急激でない場合に最適に機能します。たとえば、複数のオブジェクトが同時に表示される Web サイトをテストしている場合には、ホットキー モードを使用してテストを記録します。
 - **ホット キー** : 記録するオブジェクトをホット キーの組み合わせを押して識別するには、このモードを使用します。このモードでは、ロケータをキャプチャするときにアプリケーションは一時停止します。たとえば、複数のウィンドウを開いたり閉じたりする Web サイトをテストしている場合には、正しいウィンドウをキャプチャすることを確実にするためにアプリケーションを一時停止するこのモードを使用すると良いでしょう。デフォルトのホット キーの組み合わせは、Ctrl+Shift です。
6. オブジェクト マップ エントリを記録するか、XPath ロケータを記録するかを定義するには、**オブジェクト マップ エントリを記録する** リストから適切な記録モードを選択します。
 - **オブジェクト マップ エントリ (新しいオブジェクトと既存のオブジェクト)**。これはデフォルトのモードです。

- **XPath ロケーター (新しいオブジェクトと既存のオブジェクト)**。オブジェクト マップ エントリは記録されません。
- **XPath ロケーター (新しいオブジェクトのみ)**。オブジェクト マップに既に存在するオブジェクトに対しては、オブジェクト マップ エントリが再利用されます。この設定を選択すると、AUT のメインコントロールに対するオブジェクト マップを作成し、AUT に対して追加のテストを作成する間、これらのオブジェクト マップを保持することができます。

7. **OK** をクリックします。

UI オートメーション オプションの設定

Microsoft UI オートメーション (MSUIA) サポートの有効/無効の切り替え、ロケーター生成用のカスタム属性の指定、ロケーターから除外する属性の指定を行います。

UI オートメーション サポートを有効にして、ロケーターから除外する属性と値を指定することで、UI オートメーション プロバイダー インターフェイスを実装するアプリケーション (JavaFX や QT ベースのアプリケーションなど) のオブジェクトを、より正確に Silk Test Workbench が識別できるようにすることができます。

Silk Test Workbench には、ロケーターが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケーター生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。

また、UI オートメーション プロバイダー インターフェイスを実装するアプリケーションの任意のプロパティをカスタム属性として設定して使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。
3. **UI オートメーション** をクリックします。
4. 通常の Win32 コントロールの認識の代わりに Microsoft UI オートメーション サポートも有効化する場合は、**Microsoft UI オートメーション サポートの有効化** を **はい** に設定します。



注: Java FX アプリケーションをテストする場合は、UI オートメーション サポートをオンにする必要はありません。Silk Test Workbench では、Java FX アプリケーションに対して、デフォルトでオンになります。



注: UI オートメーション サポートは、標準のテクノロジー ドメイン固有のサポートをオーバーライドします。UI オートメーション サポートが必要なコントロールに対する操作を完了したら、UI オートメーション サポートをオフにして、標準コントロールに対する操作に戻すこともできます。

5. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
6. **UI オートメーション** をクリックします。
7. UI オートメーション プロバイダー インターフェイスを実装するアプリケーションのカスタム属性を追加する場合は、**カスタム属性** テキスト ボックスに使用する属性を入力します。
カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のオブジェクトが追加されると変更される可能性があるためです。
8. **ロケーター属性名除外リスト** グリッドで、記録中に無視する属性名を入力します。
たとえば、automationid という名前の属性を持つコントロールが AUT に複数存在するため無視したい場合は、automationid 属性名をリストに追加します。
複数の属性名を指定する場合にはカンマで区切ります。
9. **ロケーター属性値除外リスト** グリッドで、記録中に無視する属性値を入力します。
たとえば、すべての JavaFX コントロールに JavaFX<番号> という属性値があるため、デフォルトで値 JavaFX* がリストに追加されています。

複数の属性名を指定する場合にはカンマで区切ります。

100K をクリックします。

ブラウザの記録オプションの設定

カスタム属性、記録中に無視するブラウザ属性、DOM 関数の代わりに、ユーザーの入力そのものを記録するかどうかを指定します。

Silk Test Workbench には、ロケーターが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケーター生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。

xBrowser アプリケーションでは、任意のプロパティを取得し、カスタム属性として使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。


1. ツール > オプション をクリックします。

2. オプション メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。記録 オプション が右側のパネルに表示されます。


3. xBrowser をクリックします。

4. Web アプリケーションのカスタム属性を追加するには、**カスタム属性** テキスト ボックスに、使用する属性を入力します。

カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のオブジェクトが追加されると変更される可能性があるためです。

 **注:** Web アプリケーションにカスタム属性を含めるためには、HTML タグとして追加します。たとえば、MyAutomationID という属性を追加するには、`<input type='button' MyAutomationID='abc' value='click me' />` と入力します。

複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として loginName を 2 つの異なるテキスト フィールドに割り当てた場合は、loginName 属性を呼び出したときに、両方のフィールドが返されます。

 **注:** 属性名の長さは、62 文字までという制限があります。

5. **ロケーター属性名除外リスト** テキスト ボックスで、記録中に無視する属性名を入力します。

このリストを使用して、サイズ、幅、高さ、スタイルなどの頻繁に変更される属性を指定します。ワイルドカード '*' および '?' を **ロケーター属性名除外リスト** で使用できます。

たとえば、height という名前の属性を記録しない場合には、height 属性名をリストに追加します。

複数の属性名を指定する場合にはカンマで区切ります。

6. **ロケーター属性値除外リスト** テキスト ボックスで、記録中に無視する属性値を入力します。

たとえば、x-auto という値を持つ属性を記録しない場合には、x-auto 属性値をリストに追加します。

一部の AJAX フレームワークでは、ページが再読み込みされるたびに変わる属性値が生成されます。このリストを使用して、そのような値を無視します。このリストでワイルドカードを使用することもできます。

複数の属性名を指定する場合にはカンマで区切ります。

7. DOM 関数の代わりにユーザーの入力そのものを記録するには、**ネイティブなユーザー入力を記録する** リスト ボックスから、**はい** を選択します。

たとえば、DomClick の代わりに Click を記録し、SetText の代わりに TypeKeys を記録するには、**はい** を選択します。

アプリケーションでプラグインまたは AJAX を使用している場合は、**はい** を指定して、ユーザーの入力そのものを使用します。アプリケーションでプラグインまたは AJAX を使用していない場合は、再生中にブラウザにフォーカスを設定したりブラウザをアクティブにしたりする必要がない高レベル DOM 関

数を使用することをお勧めします。テストで DOM 関数を使用すると、より高速になり、信頼性も高まります。

8. ブラウザの iframe および frame サポートを有効化するには、**iframe サポートの有効化** リストで **はい** を選択します。

Web アプリケーションの iframe の内容を無視して良い場合、iframe のサポートを無効にすると再生速度が速くなる可能性があります。たとえば、多くの広告が表示される Web ページやモバイル ブラウザでテストする際に iframe サポートを無効にすると、再生パフォーマンスが大幅に向上する可能性があります。このオプションは、Internet Explorer では無視されます。このオプションは、デフォルトで有効になっています。

9. **OK** をクリックします。

カスタム属性の設定

Silk Test Workbench には、ロケーターが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケーター生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。


頻繁には変更されない属性を利用して、適切に定義されたロケーターでは、メンテナンス作業が少なく抑えられます。カスタム属性を使用すると、caption や index などの他の属性を使用するよりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は他のオブジェクトが追加されると変更される可能性があるためです。

カスタム属性 タブのリスト ボックスに一覧表示されているテクノロジー ドメインの場合、任意のプロパティ (*myCustomProperty* を定義する WPFButton など) を取得し、それらのプロパティをカスタム属性として使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。Web アプリケーションでは、操作する要素に `<div myAutomationId= "my unique element name" />` などの属性を追加できます。また、Java SWT では、GUI を実装する開発者が属性 (testAutomationId など) をウィジェットに対して定義することによって、アプリケーション内でそのウィジェットを一意に識別できます。テスト担当者は、その属性をカスタム属性 (この場合は testAutomationId) のリストに追加し、その一意の ID によってコントロールを識別できます。この手法によって、ロケーターの変更に伴うメンテナンス作業を回避することができます。

caption のように、複数のオブジェクトで同じ属性値が共有されている場合、Silk Test Workbench は、複数の利用可能な属性を "and" 操作で結合してロケーターを一意にするよう試み、一致したオブジェクトのリストを単一のオブジェクトになるまで絞り込んでいきます。それができなくなった場合には、索引を付加します。つまり、ロケーターは caption が xyz である *n* 番目のオブジェクトを探すことを意味します。


複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として loginName を 2 つの異なるテキスト フィールドに割り当てた場合は、loginName 属性を呼び出したときに、両方のフィールドが返されます。


1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. テストするアプリケーションのテクノロジー ドメインをクリックします。
たとえば、SAP アプリケーションをテストする場合は、メニュー ツリーで **SAP** をクリックします。


 **注:** Flex または Windows API ベースのクライアント/サーバー (Win32) アプリケーションには、カスタム属性を設定できません。

4. **カスタム属性** テキスト ボックスに、使用する属性を入力します。
カスタム属性が利用可能な場合は、ロケーター生成プログラムは、他の属性の前にそれらの属性を使用します。リストの順番は、ロケーター生成プログラムが使用する属性の優先順位を表しています。指定した属性が選択したオブジェクトに対して利用可能ではなかった場合には、Silk Test Workbench はテストしているアプリケーションのデフォルトの属性を使用します。

複数の属性名を指定する場合にはカンマで区切ります。

 **注:** Web アプリケーションにカスタム属性を含めるためには、HTML タグとして追加します。たとえば、bcauid という属性を追加するには、`<input type='button' bcauid='abc' value='click me' />` と入力します。

 **注:** Java SWT コントロールにカスタム属性を含めるには、`org.swt.widgets.Widget.setData(String key, Object value)` メソッドを使用します。

 **注:** Swing コントロールにカスタム属性を含めるには、`putClientProperty("propertyName", "propertyValue")` メソッドを使用します。

5. **OK** をクリックします。

記録/再生の対象とする WPF クラスの設定

Silk Test Workbench は、一般的に機能テストに無関係なコントロールは除外します。たとえば、レイアウトを目的として使用されるコントロールは含まれません。しかし、カスタム コントロールが除外されたクラスから派生している場合、除外されたコントロールを記録/再生の対象とするためには、関連する WPF クラスの名前を指定します。

記録や再生の対象にしたい WPF クラスの名前を指定します。たとえば、*MyGrid* というカスタム クラスが WPF Grid クラスから継承された場合、*MyGrid* カスタム クラスのオブジェクトは記録や再生に使用できません。Grid クラスはレイアウト目的のためにのみ存在し、機能テストとは無関係であるため、Grid オブジェクトは記録や再生に使用できません。この結果、Grid オブジェクトはデフォルトでは公開されません。機能テストに無関係なクラスに基づいたカスタム クラスを使用するには、カスタム クラス (この場合は *MyGrid*) を **OPT_WPF_CUSTOM_CLASSES** オプションに追加します。これによって、記録、再生、検索、プロパティの検証など、すべてのサポートされる操作を指定したクラスに対して実行できるようになります。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。
3. **WPF** をクリックします。
4. **カスタム WPF クラス名** グリッドで、記録や再生中に公開するクラスの名前を入力します。
複数のクラス名を指定する場合にはカンマで区切ります。
5. **OK** をクリックします。

記録の出力オプションの設定

記録の出力 オプションを変更して、記録中にコントロールをキャプチャするかどうか、プロパティを記録するかどうか、画面のキャプチャ方法、個別の画面をキャプチャするかどうかを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **出力** の隣にあるプラス記号 (+) をクリックします。 **ビジュアル テスト** サブカテゴリが表示されます。
4. **ビジュアル テスト** をクリックします。
5. オプション パネルの最初の列でオプションの名前を選択し、2 番目の列で新しい値を選択または入力して、これらのオプションを変更します。

オプションを選択すると、その簡単な説明が下のパネルに表示されます。

以下の出力オプションを設定できます。

画面キャプチャ Silk Test Workbench で記録中にテスト アプリケーションの画面をキャプチャする方法を指定します。

テストアプリケーションおよびテストアプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。

テストアプリケーションのウィンドウのうち、記録時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。

すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、記録時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

記録時に画面をキャプチャしない場合は、**なし** に設定します。

画面キャプチャ (テストステップ)

画面プレビュー に表示される個別の画面をキャプチャする場合は、**はい** に設定します。これによって、**テスト ステップ** ペインの個々のテストステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較できます (再生時にもテスト ステップの画面がキャプチャされる場合)。

記録中に個別のテスト ステップをキャプチャしない場合は、**いいえ** に設定します。**画面キャプチャ** 設定が **なし** 以外に設定された場合、ダイアログ ボックスまたは Web ページで発生する操作のグループのメイン コンテキストに対して画面がキャプチャされます。この結果、各テストステップは **画面プレビュー** でグループ画面に関連付けられます。


コントロールのキャプチャ

記録中にコントロールとそのプロパティをキャプチャするには、**はい** に設定します。コントロールは、テスト アプリケーションにアクセスしなくても **画面プレビュー** を使用して選択できます。再生のパフォーマンスを向上させるには、このオプションを **いいえ** に設定します。

Web アプリケーションの記録中は、キャプチャされたコントロールとともに以下の属性が格納されます。


- id
- class
- name
- value
- href
- innerText

属性を追加するには、**ツール > オプション > 記録 > xBrowser** の **カスタム属性** フィールドを使用します。

 **注:** 属性のテキスト値は、**ツール > オプション > 記録 > xBrowser** の **属性値の最大の長さ** フィールドで指定した長さに切り詰められます。

ブラウザコントロールのキャプチャ

Web ブラウザの記録中にコントロールとそのプロパティをキャプチャするかどうかを定義します。デフォルト値は、**いいえ** です。**はい** に設定すると、Web ブラウザにアクセスすることなく、**プレビュー画面** でコントロールとそのプロパティを扱うことができます。**コントロールのキャプチャ** オプションが **いいえ** に設定されていると、Silk Test Workbench はこのオプションを無視します。

 **警告:** このオプションを **はい** に設定すると、テストの再生に時間がかかることがあります。再生のパフォーマンスを向上させるには、このオプションを **いいえ** のままにしてください。

 **ヒント:** これらのオプションを **再生結果: ビジュアル テスト** オプションと組み合わせて使用して、再生結果を、ビジュアル テストを最初に記録したときにキャプチャした画面と比較できます。

6. 省略可能 : **名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。

7. **OK** をクリックします。

記録の出力：ビジュアルテストオプション

記録の出力：ビジュアルテスト オプションを使用して、ビジュアルテストでコントロールおよび画面を記録する場合の動作を制御します。これらのオプションの使用の詳細については、「データベースの膨張の抑制」を参照してください。

以下の出力オプションを設定できます。

画面キャプチャ

Silk Test Workbench で記録中にテスト アプリケーションの画面をキャプチャする方法を指定します。

テスト アプリケーションおよびテスト アプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。

テスト アプリケーションのウィンドウのうち、記録時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。

すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、記録時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

記録時に画面をキャプチャしない場合は、**なし** に設定します。

画面キャプチャ (テストステップ)

画面プレビュー に表示される個別の画面をキャプチャする場合は、**はい** に設定します。これによって、**テスト ステップ** ペインの個々のテストステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアルテストを最初に記録したときにキャプチャされた画面と比較できます (再生時にもテスト ステップの画面がキャプチャされる場合)。

記録中に個別のテスト ステップをキャプチャしない場合は、**いいえ** に設定します。**画面キャプチャ** 設定が **なし** 以外に設定された場合、ダイアログ ボックスまたは Web ページで発生する操作のグループのメイン コンテキストに対して画面がキャプチャされます。この結果、各テスト ステップは **画面プレビュー** でグループ画面に関連付けられます。

コントロールのキャプチャ

記録中にコントロールとそのプロパティをキャプチャするには、**はい** に設定します。コントロールは、テスト アプリケーションにアクセスしなくても **画面プレビュー** を使用して選択できます。再生のパフォーマンスを向上させるには、このオプションを **いいえ** に設定します。

Web アプリケーションの記録中は、キャプチャされたコントロールとともに以下の属性が格納されます。

- id
- class
- name
- value
- href
- innerText

属性を追加するには、**ツール > オプション > 記録 > xBrowser** の **カスタム属性** フィールドを使用します。



注: 属性のテキスト値は、**ツール > オプション > 記録 > xBrowser** の **属性値の最大の長さ** フィールドで指定した長さに切り詰められます。

ブラウザコントロールのキャプチャ

Web ブラウザの記録中にコントロールとそのプロパティをキャプチャするかどうかを定義します。デフォルト値は、**いいえ** です。**はい** に設定すると、Web ブラウザにアクセスすることなく、**プレビュー画面** でコントロールとそのプロパティを扱うことができます。**コントロールのキャプチャ** オプションが **いいえ** に設定されていると、Silk Test Workbench はこのオプションを無視します。



警告: このオプションを **はい** に設定すると、テストの再生に時間がかかることがあります。再生のパフォーマンスを向上させるには、このオプションを **いいえ** のままにしてください。

記録のホットキー オプションの設定

ホットキー オプションを変更して、記録の開始および停止に使用するキーの組み合わせ、または記録中に検証を挿入するキーの組み合わせを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **ホットキー** をクリックします。

次のようなホットキーの組み合わせを設定できます。

検証の挿入	記録を一時的に停止するキーの組み合わせを指定し、 テスト ロジック デザイナ ウィザードを表示します。ウィザードが開いたあと、検証するオブジェクトを選択します。
現在のオブジェクトを使用して検証を挿入	記録を一時的に停止するキーの組み合わせを指定し、現在ハイライトされているオブジェクトを検証用を選択します。このキーの組み合わせでは、 テスト ロジック デザイナ ウィザードが開き、選択されたオブジェクト セットが検証対象のコントロールとなります。
記録の開始/停止	記録のオン/オフを切り替えるために使用するキーの組み合わせを指定します。

4. ホットキーの組み合わせを変更するには、以下のステップを実行します。
 - a) 右側のオプション パネルで、変更するホットキー オプションを選択します。
 - b) 選択した操作の横にある省略記号ボタンをクリックします。 **ホットキーの定義** ダイアログ ボックスが開きます。
 - c) 修飾キーを使用するには、該当するチェック ボックスをオンにします。
 - d) **キー** リストから、ホットキーとして使用するキーを選択します。
キーの値は、A から Z、0 から 9、F1 から F12 です。
Silk Test Workbench により、このキー シーケンスがテキスト ボックスに入力されます。
 - e) **OK** をクリックします。
5. **OK** をクリックします。

記録オプションプロファイルの作成

記録環境をすばやく構成できるように、記録オプションのデフォルト値を変更したあと、各オプションへの変更を記録オプションプロファイルに保存できます。プロファイルを作成すると、テストスクリプトまたはビジュアルテストを実行する前に、保存した記録オプションを適用できます。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般**、**ロケータ**、**xBrowser**、**SWT**、**出力**、または **ホットキー** をクリックし、**記録** オプションを変更して、Silk Test Workbench が記録中にアプリケーションの動作を識別してキャプチャする方法を変更します。
4. **名前を付けて保存** をクリックします。 **記録設定** ダイアログ ボックスが開きます。
5. **名前** テキスト ボックスで、記録オプション設定の現在のセットの名前を入力するか、リストから既存のグループを選択します。

既存のグループを選択すると、この名前ですべて以前に保存した設定が現在の設定によって置換されます。

6. **所有者** テキスト ボックスで、これらの設定のユーザーを選択します。


7. **OK** をクリックします。

保存済み記録オプションの適用

記録オプション プロファイルを適用するには、記録オプション プロファイルを事前に作成します。

プロファイルに記録オプションのセットを保存して、それらをあとでまとめて適用します。この機能を使用すると、特定の環境に対してそれぞれ異なる記録オプション セットを維持できます。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般、ロケーター、xBrowser、SWT、出力**、または **ホット キー** をクリックします。
4. **記録設定** リストから、適用する保存済みオプションの名前を選択します。元の設定を復元する場合は、**システム デフォルト** を選択します。

 **注:** 既存のオプション プロファイルを変更する場合は、オプション プロファイルを正しく更新してください。既存のオプション プロファイルを変更するには、使用する値にオプションを設定して **名前を付けて保存** をクリックし、適切なオプション プロファイル名を指定する必要があります。このようにしない場合、選択したオプションはダイアログ ボックスに保存されますが、オプション プロファイルには適用されません。

Silk Test Workbench では、保存済みプロファイルの設定によって、現在の **記録** オプションが更新されます。

5. **OK** をクリックします。

再生オプション

再生オプションは、以下のとおりです。

全般	オブジェクトが呼び出されるタイミング、再生中にオブジェクトをハイライトするかどうか、応答のないアプリケーションを閉じるかどうか、再生モード、 出力 ウィンドウにスクリプトのオートメーション呼び出しを表示するかどうか、アプリケーションがアクティブになるまで待機する時間を指定します。
タイミング	アプリケーションが準備完了状態になるまで待機する時間 (ミリ秒)、ウィンドウのスクリーンショットを取得する前に待機する時間 (ミリ秒)、2 つの列挙をキャプチャする間に経過可能な時間、再生中のキー入力間の遅延、スクリプトの各マウス イベントの前の遅延、再生中にオブジェクトが解決されるまで待機する時間、および応答のない再生操作をキャンセルするまでの時間を指定します。
結果	デフォルトの結果名、結果の番号付け方式、合格条件のパーセンテージ、すべての情報を保存するかどうか、再生中にビジュアル テストのコントロールおよび画面をキャプチャするかどうかを設定します。
ホット キー	ツール バーのボタン、資産のダイアログ ボックス、またはショートカット キーの組み合わせ (Ctrl+S) で資産を保存するときの、デフォルトの保存動作を設定します。動作は、資産の新しいバージョンを作成する、または資産の現在のバージョンを置き換える、のいずれかから選択します。
xBrowser	同期から除外する URL、同期モード、および Web アプリケーションのタイムアウト値を指定します。
閉じる	ウィンドウやダイアログ ボックスを閉じるボタン、メニュー項目、確認ダイアログ ボックス、キー入力のリストを指定します。

再生ステータス ダイアログ	テストの再生時に、 再生ステータス ダイアログ ボックスを表示するかどうか、および、ダイアログ ボックスにスクリーンショットを表示するかどうかを指定します。
Open Agent の設定	エージェントに直接送信されるカスタム設定を行います。サポートに勧められた場合など、非常に特殊なテスト シナリオでのみ設定されるべきです。

全般再生オプションの設定

オブジェクトが呼び出されるタイミング、再生中にオブジェクトをハイライトするかどうか、応答のないアプリケーションを閉じるかどうか、再生モード、**出力** ウィンドウにスクリプトのオートメーション呼び出しを表示するかどうか、アプリケーションがアクティブになるまで待機する時間を指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
3. **全般** をクリックします。

例外がスローされたときにキャッチする

この設定の目的は、発生した例外にステップインできるようにすることです。スクリプトの作成とテストを行っている場合は、これを **はい** に設定して問題の場所を確認できるようにします。自動スクリプトを実行している場合は、独自のエラー処理を備える必要があるため、この設定を **いいえ** にします。エラー処理をスクリプトでオンに切り替えた場合は、**続行** をクリックしてそれにアクセスします。この設定を **いいえ** にして独自のエラー処理を備えていない場合は、スクリプトを **終了** することしかできません。

自動 に設定したとき、Silk Test から普通に実行している場合は、この設定は **はい** に設定したかのように動作します。Silk Central Test Manager によって実行されたスクリプトを Silk Test が実行している場合は、**いいえ** に設定したかのように動作します。よって、スクリプトが独自のエラー処理を持ち、オートメーションから実行されている場合は、スクリプトは動作し続けます。

無応答のアプリケーションを閉じる

Silk Test エージェントとアプリケーションとの間の通信がエラーになったかタイムアウトになった場合、アプリケーションをシャットダウンします。

インスタンスを複数実行することができないアプリケーションをテストする場合は、**はい** を指定します。

オブジェクトがアクティブであることを保証する

呼び出しの実行前にオブジェクトがアクティブであることを保証します。

画面検証を実行する

Rumba 用の画面検証が記録/再生されるかどうかを制御します。

再生中にオブジェクトをハイライトする

再生選択中に現在のオブジェクトをハイライトします。

再生モード

以下のオプションのいずれかを選択します。

デフォルト このモードを使用すると、最も信頼できる結果が得られます。デフォルトでは、各コントロールそれぞれが、マウスやキーボード (低レベル)、あるいは API (高レベル) モードのどちらかを使用します。デフォルトモードを使用すると、各コントロールがコントロールの種類に応じて適切なモードが使用されます。

高レベル このモードを使用すると、API を使用して各コントロールが再生されます。

低レベル このモードを使用すると、マウスとキーボードを使用して各コントロールが再生されます。

出力ウィンドウにオートメーション操作を表示する スクリプトのオートメーション呼び出しを **出力** ウィンドウに表示します。
出力 ウィンドウを開いて結果を確認する必要があります。**表示 > 出力** をクリックします。

ローカル変数ウィンドウにプロパティを表示する この設定では、通常メンバの変数とパラメータとともに、クラスのプロパティを **ローカル変数** ウィンドウに表示するかどうかを制御します。クラスのプロパティには実装コード (計算) が含まれることがあるので、これにエラーが含まれて、その値が問題を示すことがあります。また、プロパティの計算にはメンバの変数の取得よりも長い時間がかかることがあります。デフォルト値は **はい** です。

4. **OK** をクリックします。

xBrowser 同期オプションの設定

Web アプリケーションの同期およびタイムアウトの値を指定します。同期は、すべてのメソッド呼び出しの前後に自動的に実行されます。メソッド呼び出しは、同期条件が満たされるまで開始せず、終了もしません。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
3. **xBrowser** をクリックします。
4. **同期除外リスト** テキスト ボックスに、除外するサービスまたは Web ページの URL 全体あるいは URL の一部を入力します。

AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を **同期除外リスト** 設定で指定します。

たとえば、クライアントからデータをポーリングすることによってサーバー時間を表示するウィジェットを Web アプリケーションで使用する場合は、このウィジェットのトラフィックが永続的にサーバーに送信されます。このサービスを同期から除外するには、サービス URL を判別し、除外リストに入力します。

たとえば、以下のように入力します。

- http://example.com/syncsample/timeService
- timeService
- UICallbackServiceHandler

複数のエントリをカンマで区切って指定します。



注: アプリケーションで 1 つのサービスのみが使用されている場合、そのサービスでテストを無効にするには、サービス URL を除外リストに追加するのではなく、HTML 同期モードを使用する必要があります。



ヒント: 除外リストには、URL 全体ではなく URL の部分文字列を追加することを、Micro Focus ではお勧めしています。たとえば、http://example.com/syncsample/timeService ではなく、/syncsample を除外リストに追加します。ブラウザは相対 URL のみを Silk Test Workbench に返す場合があるため、URL 全体を指定すると機能しない場合があります。たとえば、ブラウザは /syncsample/timeService だけを返すのに対して、除外リストに http://example.com/syncsample/timeService を追加していると、Silk Test Workbench は返された URL を除外しません。

5. **同期モード** リスト ボックスから、Web アプリケーションを準備可能状態にするための同期アルゴリズムを選択します。
同期アルゴリズムは、呼び出しが可能になる状態までの待機時間を設定します。
HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれる、より複雑なシナリオが使用される場合は、同期関数を手動で記述することが必要になる可能性があります。
AJAX モードを使用すると、同期関数を手動で記述する必要がなくなるため、スクリプト（オブジェクトの表示または非表示を待機したり、特定のプロパティ値を待機するなど）の作成処理が大幅に簡略化されます。また、この自動同期は、スクリプトを手動で適用しないで記録と再生を正常に行うための基礎となります。
6. **同期タイムアウト** テキスト ボックスで、オブジェクトが準備可能状態になるまで待機する最大時間をミリ秒で入力します。デフォルト値は、300000 に設定されています。
7. **iframe サポートのホワイトリスト** に、テストの対象となる iframe と frame の属性リストを指定します。
リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象から除外されます。ワイルドカードを使用できます。たとえば、"name:*form" という項目を定義すると、<IFRAME name="user-form" src=...> がテスト対象に含まれます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。
8. **iframe サポートのブラックリスト** に、テストから除外する iframe と frame の属性リストを指定します。
リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象に含まれます。ワイルドカードを使用できます。たとえば、"src:*advertising*" という項目を定義すると、<IFRAME src=http://my.domain/advertising-banner.html> がテスト対象から除外されます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。
9. **OK** をクリックします。

再生のタイミング オプションの設定

再生 オプションを使用して、再生中のタイミングを制御できます。

1. **ツール > オプション** をクリックします。 **オプション** ダイアログ ボックスが開きます。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** オプションが右側のパネルに表示され、**タイミング** サブカテゴリが **再生** の下のアウトライン ビューに表示されます。
3. **タイミング** をクリックします。
4. **アプリケーション準備完了タイムアウト** テキスト ボックスで、アプリケーションが準備完了状態になるまで待機する時間をミリ秒で指定します。
指定したタイムアウト時間内にアプリケーションの準備が整わない場合は、Silk Test Workbench によって例外がスローされます。
5. **キーボード イベント遅延** テキスト ボックスで、再生中のキー入力間の遅延をミリ秒で指定します。テストするアプリケーションに応じて、選択する最適な値は異なります。たとえば、Web アプリケーションをテストする場合、1 ミリ秒に設定すると、ブラウザが極端に遅くなります。ただし、この値を 0 (ゼロ) に設定すると、基本的なアプリケーション テストに失敗する可能性があります。
6. **マウス イベント遅延** テキスト ボックスで、スクリプトの各マウス イベントの前に使用される遅延をミリ秒で指定します。
この遅延は、マウスを移動したり、ボタンを押したり離したりする操作に影響します。
7. **オブジェクト有効化タイムアウト** テキスト ボックスで、再生中にオブジェクトが有効になるまで待機する時間をミリ秒で指定します。
8. **オブジェクト解決タイムアウト** テキスト ボックスで、再生中にオブジェクトが解決されるまで待機する時間をミリ秒で指定します。

9. スクリーンショット遅延 の値を指定します。

新しいウィンドウにフォーカスを設定してからウィンドウのスクリーンショットを取得するまで待機する時間をミリ秒単位で入力します。

デフォルト値は、300 ミリ秒に設定されています。

10無応答のアプリケーションのタイムアウト テキスト ボックスで、応答のない再生操作をキャンセルするまでの時間をミリ秒で入力します。

11省略可能：名前を付けて保存 をクリックして、再生 オプションの設定をまとめて保存します。

12OK をクリックします。

UI オートメーション オプションの設定

Microsoft UI オートメーション (MSUIA) サポートの有効/無効の切り替え、ロケータ生成用のカスタム属性の指定、ロケータから除外する属性の指定を行います。

UI オートメーション サポートを有効にして、ロケータから除外する属性と値を指定することで、UI オートメーション プロバイダー インターフェイスを実装するアプリケーション (JavaFX や QT ベースのアプリケーションなど) のオブジェクトを、より正確に Silk Test Workbench が識別できるようにすることができます。

Silk Test Workbench には、ロケータが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケータ生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。

また、UI オートメーション プロバイダー インターフェイスを実装するアプリケーションの任意のプロパティをカスタム属性として設定して使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。

1. ツール > オプション をクリックします。

2. オプション メニュー ツリーの 再生 の隣にあるプラス記号 (+) をクリックします。再生 のサブカテゴリが表示されます。

3. UI オートメーション をクリックします。

4. 通常の Win32 コントロールの認識の代わりに Microsoft UI オートメーション サポートも有効化する場合は、Microsoft UI オートメーション サポートの有効化 を はい に設定します。



注: Java FX アプリケーションをテストする場合は、UI オートメーション サポートをオンにする必要はありません。Silk Test Workbench では、Java FX アプリケーションに対して、デフォルトでオンになります。



注: UI オートメーション サポートは、標準のテクノロジー ドメイン固有のサポートをオーバーライドします。UI オートメーション サポートが必要なコントロールに対する操作を完了したら、UI オートメーション サポートをオフにして、標準コントロールに対する操作に戻すこともできます。

5. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。記録 オプションが右側のパネルに表示されます。

6. UI オートメーション をクリックします。

7. UI オートメーション プロバイダー インターフェイスを実装するアプリケーションのカスタム属性を追加する場合は、カスタム属性 テキスト ボックスに使用する属性を入力します。

カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることが出来ます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のオブジェクトが追加されると変更される可能性があるためです。

8. ロケータ属性名除外リスト グリッドで、記録中に無視する属性名を入力します。

たとえば、automationid という名前の属性を持つコントロールが AUT に複数存在するため無視したい場合は、automationid 属性名をリストに追加します。

複数の属性名を指定する場合にはカンマで区切ります。

9. **ロケータ属性値除外リスト** グリッドで、記録中に無視する属性値を入力します。

たとえば、すべての JavaFX コントロールに JavaFX<番号> という属性値があるため、デフォルトで値 JavaFX* がリストに追加されています。

複数の属性名を指定する場合にはカンマで区切ります。

100K をクリックします。

閉じるオプションの設定

ウィンドウやダイアログ ボックスを閉じるためのボタン、メニュー項目、キー入力のリストを指定します。確認ダイアログ ボックスを閉じるためのボタンのリストも指定します。また、ウィンドウのクローズ方法を試行するまでの間隔も指定します。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。

3. **閉じる** をクリックします。

4. **ウィンドウを閉じるためのボタン** テキスト ボックスで、Close、CloseWindows、および Exit で閉じられるウィンドウを閉じるために使用するボタンのリストを指定します。

5. **確認ダイアログに対するボタン** テキスト ボックスで、確認ダイアログ ボックス (Close、CloseWindows、および Exit でウィンドウを閉じるときに表示されるダイアログ ボックス) を閉じるために使用するボタンのリストを指定します。

6. **ダイアログを閉じるためのキー** テキスト ボックスで、ダイアログ ボックス (Close、CloseWindows、および Exit で使用される) を閉じるために使用するキー シーケンスを指定します。

7. **ウィンドウを閉じるためのメニュー項目** テキスト ボックスで、Close、CloseWindows、および Exit でウィンドウを閉じるために使用するメニュー項目のリストを指定します。

8. **ウィンドウを閉じるためのタイムアウト** の値を指定します。

ウィンドウ閉じるための次の方式を試行する前に待機する時間をミリ秒で入力します。最終的に失敗と判断する前に、4 種類の方式が実行されます。つまり、閉じるのに失敗するまでにかかる合計時間は、指定した値の 4 倍の時間になります。

9. **OK** をクリックします。

再生結果オプションの設定

再生結果 オプションを変更して、デフォルトの結果名、結果の番号付け方式、合格条件のパーセンテージ、すべての情報を保存するかどうかを設定します。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。

3. **結果** をクリックします。**結果** オプションが右側のペインに表示されます。

4. 環境に適したオプションを設定します。

デフォルトの結果名 各結果の結果名として使用する名前を入力します。ビジュアル テストまたはテスト スクリプトの名前を結果名として使用する場合は、このフィールドを空のままにします。

結果オプション 以下のリストから選択します。

- **新しい実行番号で結果を保存** : ビジュアル テストまたはテスト スクリプトの再生の結果は、新しいインクリメント実行番号で保存されます。たとえば、新規の結果は実行番号 1 で保存されます。同じビジュアル テストまたはテスト スクリプトの後続の結果は、実行番号 2、3、4 など保存されます。これはデフォルトの選択です。

- **現在の実行番号に結果を追加**：現在の再生の結果は、最新の既存の実行番号の結果に追加されます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。
- **既存の実行番号を新しい結果で上書き**：現在の再生の結果で、最新の既存の実行番号の結果が上書きされます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。
- **この結果を保存しない**：ビジュアルテストまたはテストスクリプトを再生したときに結果は作成されません。

結果合格条件 (パーセンテージ)

今後すべての実行の合格基準として、検証の合格の割合を設定します。たとえば、結果の合格条件を 90% に設定した場合は、ビジュアルテストにおける 10 件の検証のうち 9 件以上に合格しないと、再生の結果が合格になりません。

すべての情報を保存

結果に、ビジュアルテストの再生に関するすべての情報を保存するか、要約情報とエラー情報のみを保存するかを決定します。

5. 省略可能：**名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。
6. **OK** をクリックします。

ビジュアルテストの再生結果オプションの設定

再生結果：ビジュアルテスト オプションを使用して、再生中にビジュアルテストでキャプチャするコントロールおよび画面を設定します。これらのオプションの使用の詳細については、「データベースの膨張の抑制」を参照してください。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
3. **結果** の隣にあるプラス記号 (+) をクリックし、**ビジュアルテスト** をクリックします。**ビジュアルテスト** オプションが右側のペインに表示されます。
4. 環境に適したオプションを設定します。

コントロールのキャプチャ

再生中にコントロールとそのプロパティをキャプチャします。

ビジュアルテストを記録したときに **記録の出力** オプションで設定されていた **コントロールのキャプチャ** 設定と同じ設定を使用するには、**記録と同じ** に設定します。

再生中にコントロールとそのプロパティをキャプチャするには、**はい** に設定します。コントロールは **画面プレビュー** を使用して選択できます。



注：記録の **画面キャプチャ** 設定を **なし** に設定した場合は、**コントロールのキャプチャ** を **はい**、再生の **画面キャプチャ** 設定を **なし** 以外の値に設定して、再生時にコントロールをキャプチャする必要があります。

再生のパフォーマンスを向上させるには、**いいえ** に設定します。

画面キャプチャ

Silk Test Workbench で再生中にテストアプリケーションの画面をキャプチャする方法を指定します。


ビジュアルテストを記録したときに各画面に設定されていた **画面キャプチャ** 設定と同じ設定を使用するには、**記録と同じ** に設定します。たとえば、**アプリケーション** と **アクティブウィンドウ** の両方が設定された状態で記録された画面が含まれるビジュアルテストは、ビジュアルテストで各画面に適用されたように、これらの設定を使用して再生されます。

テストアプリケーションおよびテストアプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。

テスト アプリケーションのウィンドウのうち、再生時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。


すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、再生時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

再生時に画面のキャプチャを無効にするには、**なし** に設定します。

 **注:** この設定を **なし** に設定すると、**コントロールのキャプチャ** および **画面キャプチャ (テスト ステップ)** 設定に指定した値に関係なく、コントロールまたは画面は **結果** ウィンドウにキャプチャされません。

画面キャプチャ (テスト ステップ) **画面プレビュー** に表示される個別の画面をキャプチャする場合は、**はい** に設定します。これによって、**テスト ステップ** ペインの個々のテスト ステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較できます (記録時にもテスト ステップの画面がキャプチャされた場合)。

再生中に個別のテスト ステップをキャプチャしない場合は、**いいえ** に設定します。**画面キャプチャ** 設定が **なし** 以外に設定された場合、個別のテスト ステップに関連付けられたグループ画面が **画面プレビュー** に表示されます。

 **ヒント:** これらのオプションを **記録の出力** オプションと組み合わせて使用して、再生結果を、ビジュアル テストを最初に記録したときにキャプチャした画面と比較できます。

5. 省略可能 : **名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。

6. **OK** をクリックします。

.NET スクリプトの再生結果オプションの設定

再生結果 : .NET スクリプト オプションを使用して、.NET スクリプトの再生時の画面キャプチャを設定できます。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。

3. **結果** の隣にあるプラス記号 (+) をクリックし、**.NET スクリプト** をクリックします。**.NET スクリプト** オプションが右側のペインに表示されます。

4. **画面キャプチャ** リストから、再生中にテスト アプリケーションの画面を Silk Test Workbench がキャプチャする方法を選択します。

- 再生時に画面のキャプチャを無効にするには、**なし** に設定します。
- テスト アプリケーションおよびテスト アプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。
- テスト アプリケーションのウィンドウのうち、再生時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。
- すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、再生時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

5. 省略可能 : **名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。

6. **OK** をクリックします。

キーワード駆動テストの再生結果オプションの設定

再生結果 : キーワード駆動テスト オプションを使用して、キーワード駆動テストの再生中に画面をキャプチャを設定できます。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。
3. **結果** の隣にあるプラス記号 (+) をクリックし、 **キーワード駆動テスト** をクリックします。 **キーワード駆動テスト** オプションが右側のペインに表示されます。
4. **画面キャプチャ** リストから、再生中にテスト アプリケーションの画面を Silk Test Workbench がキャプチャする方法を選択します。
 - 再生時に画面のキャプチャを無効にするには、 **なし** に設定します。
 - テスト アプリケーションおよびテスト アプリケーション内のすべてのウィンドウをキャプチャする場合は、 **アプリケーション** に設定します。
 - テスト アプリケーションのウィンドウのうち、再生時にアクティブなウィンドウのみをキャプチャするには、 **アクティブ ウィンドウ** に設定します。
 - すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、再生時にデスクトップに表示される画面をキャプチャするには、 **デスクトップ** に設定します。
5. 省略可能 : **名前を付けて保存** をクリックして、 **再生** オプションの設定をまとめて保存します。
6. **OK** をクリックします。

再生ホットキーの変更

Silk Test Workbench が最小化されているときに再生を停止するための再生ホット キーを変更します。

[オプション] メニュー ツリーで、 **再生** の隣にあるプラス記号 (+) をクリックし、 **ホット キー** をクリックします。 **再生の停止** を選択して、キーの指定を編集します。 デフォルトでは、再生を停止するホット キーは Alt+F12 です。

再生オプションプロファイルの作成

プロファイルに再生オプションのセットを保存して、それらをあとでまとめて適用できます。 この機能を使用すると、特定の環境に対してそれぞれ異なる再生オプション セットを維持できます。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。
3. **タイミング**、 **結果**、 **ビジュアルテスト**、または **ホット キー** をクリックし、 **再生** オプションを変更して、Silk Test Workbench が再生中にアプリケーションの動作を識別してキャプチャする方法を変更します。
4. **名前を付けて保存** をクリックします。 **再生設定** ダイアログ ボックスが開きます。
5. **名前** テキスト ボックスで、再生オプション設定の現在のセットの名前を入力するか、ドロップダウン リストから既存のグループを選択します。 既存のグループを選択すると、この名前ですべて以前に保存した設定が現在の設定によって置換されます。
6. **所有者** テキスト ボックスで、これらの設定のユーザーを選択します。
7. **OK** をクリックします。

保存済み再生オプションの適用

再生記録オプション プロファイルを適用するには、再生オプション プロファイルを事前に作成する必要があります。

テスト基準に適合するように、デフォルトの再生オプションを復元するか、またはテスト再生用に作成した新規プロファイルを適用します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。

3. **タイミング**、**結果**、**ビジュアルテスト**、または **ホットキー** をクリックします。
4. パネル最下部の **再生設定** リストから、適用する保存済みオプションの名前を選択します。Silk Test Workbench の元の設定を復元する場合は、**システムデフォルト** を選択します。



注: 既存のオプションプロファイルを変更する場合は、オプションプロファイルを正しく更新してください。既存のオプションプロファイルを変更するには、使用する値にオプションを設定して **名前を付けて保存** をクリックし、適切なオプションプロファイル名を指定する必要があります。このようにしない場合、選択したオプションはダイアログボックスに保存されますが、オプションプロファイルには適用されません。

Silk Test Workbench では、保存済みプロファイルの設定によって、現在の **再生** オプションが更新されます。

5. **OK** をクリックして、新しいオプション設定をアクティブにします。

スクリプト作成オプション

スクリプト作成 オプションによって、フォントの種類やサイズなどのスクリプトのレイアウトを制御します。

1. **ツール > オプション** をクリックします。
2. **オプション** ツリーで **スクリプト作成** をクリックします。スクリプト作成オプションが右側のパネルに表示されます。
3. **フォント** テキストボックスで、省略記号ボタンをクリックします。**フォント** ダイアログボックスが開きます。
4. スクリプトで使用するフォントの種類、スタイル、サイズを選択し、**OK** をクリックします。
5. **OK** をクリックします。

詳細オプションの設定

詳細オプションでは、フォールバックサポートを有効化したり、ローケター属性名の大文字小文字を区別するかどうか、などを設定できます。**詳細設定 > アクティブデータ** オプションを変更することにより、**アクティブデータ資産の設定** ウィンドウを使用してテキストベース (.csv または .txt) のアクティブデータファイルを編集する際に、そのファイルに Silk Test Workbench からデータを書き込む方法を制御することができます。

1. **ツール > オプション** をクリックします。**オプション** ダイアログボックスが表示されます。
2. **詳細設定** タブをクリックします。**詳細オプション** ページが表示されます。
3. 埋め込み Chrome アプリケーションをテストするには、**埋め込み Chrome サポートの有効化** フィールドに実行可能ファイルとポートを値ペア形式で指定します。
たとえば、myApp.exe=9222 のように指定します。
複数の埋め込み Chrome アプリケーションを指定する場合は、複数の値ペアをカンマで区切って指定します。
4. デフォルトのブラウザーサポートではテストできないハイブリッドモバイルアプリケーションに対するフォールバックサポートを使用するには、**Android と iOS 上で Web ビューのフォールバックサポート** を有効化します。
5. 標準の Win32 コントロールの解決に加えて、Microsoft ユーザー補助を有効にするには、**Microsoft ユーザー補助** を有効化にします。
6. テキストをキャプチャする前にウィンドウからフォーカスを外すには、**テキストのキャプチャ時にフォーカスを外す** を有効化します。
テキストのキャプチャは、次のメソッドによる記録および再生中に実行されます。

- TextClick
- TextCapture
- TextExists
- TextRect

7. ロケータ属性名の大文字と小文字が区別されるように設定するには、**ロケータ属性の大文字小文字を区別する** を有効化します。モバイル Web アプリケーションのロケータ属性の名前は、常に大文字と小文字の区別はされません。つまり、モバイル Web アプリケーションの記録や再生時に、このオプションは無視されます。

8. Mozilla Firefox で外部リンクを新しいウィンドウではなく新しいタブで開く場合は、**Firefox のシングルウィンドウモードを使用する** をオンにします。



注: このオプションは、Mozilla Firefox 52 以降を使用している場合にのみ動作します。

9. オプション メニュー ツリーの **詳細設定** の隣にあるプラス記号 (+) をクリックし、**アクティブ データ** をクリックします。**アクティブ データ** のオプションが右側のパネルに表示されます。

10 環境に適したオプションを設定します。

フィールドを常に引用符で囲む

新しいアクティブ データ ファイルの各セル内のデータを引用符で囲むには、**はい** を選択します。このオプションは新しいアクティブ データ ファイルにのみ関連します。このオプションを **はい** に設定しても、既存のファイルには影響しません。

アクティブ データ ファイルから使用する予定のデータに引用符が必要な場合に、このオプションを **はい** に設定します。



注: **はい** に設定しても、編集中は、新しいアクティブ データ ファイルのデータに引用符は表示されません。ファイルの元のアプリケーションを使ってアクティブ データ ファイルを編集する場合に、引用符が表示されます。

空の後続フィールドを追加する

各データ項目に対して文字の後ろに後続スペースを入力する場合は、**はい** に設定します。たとえば、フィールドに必要なデータが「data」の場合にこのオプションを **はい** に設定すると、後続のスペースはデータの一部として含まれるため、アクティブ データ ファイルのフィールドには「data」と書き込まれます。**いいえ** を選択すると、後続スペースを除いた文字のみが含まれます。たとえば、**いいえ** に設定した場合に「data」がフィールドに書き込まれると、アクティブ データ ファイルには「data」として保存されます。

アクティブ データ テストで使用されるデータで、アプリケーションを正しくテストするために後続スペースが必要な場合、このオプションを **はい** に設定してください。

11 OK をクリックします。

アクティブ データ詳細設定オプション


詳細設定 **アクティブ データ** オプションを変更することにより、**アクティブ データ資産の設定** ウィンドウを使用してテキストベース (.csv または .txt) のアクティブ データ ファイルを編集する際に、そのファイルに Silk Test Workbench からデータを書き込む方法を制御することができます。

アクティブ データに対して以下のオプションを設定できます。

フィールドを常に引用符で囲む

新しいアクティブ データ ファイルの各セル内のデータを引用符で囲むには、**はい** を選択します。このオプションは新しいアクティブ データ ファイルにのみ関連します。このオプションを **はい** に設定しても、既存のファイルには影響しません。

アクティブ データ ファイルから使用する予定のデータに引用符が必要な場合に、このオプションを **はい** に設定します。

 **注:** **はい** に設定しても、編集中は、新しいアクティブ データ ファイルのデータに引用符は表示されません。ファイルの元のアプリケーションを使ってアクティブ データ ファイルを編集する場合に、引用符が表示されます。

空の後続フィールドを追加する

各データ項目に対して文字の後ろに後続スペースを入力する場合は、**はい** に設定します。たとえば、フィールドに必要なデータが「data」の場合にこのオプションを **はい** に設定すると、後続のスペースはデータの一部として含まれるため、アクティブ データ ファイルのフィールドには「data」と書き込まれます。**いいえ** を選択すると、後続スペースを除いた文字のみが含まれます。たとえば、**いいえ** に設定した場合に「data」がフィールドに書き込まれると、アクティブ データ ファイルには「data」として保存されます。

アクティブ データ テストで使用されるデータで、アプリケーションを正しくテストするために後続スペースが必要な場合、このオプションを **はい** に設定してください。

Silk Test Workbench の管理

Silk Test Workbench データベース、プロジェクト、およびユーザーの管理手順について説明します。

Silk Test Workbench の実行に必要な管理者権限


Silk Test Workbench のインストールまたは実行には、次の権限が必要です。


- Silk Test Workbench のインストールには、ローカル管理者権限が必要です。
- Windows サーバー上への Silk Test Workbench のインストールには、ドメイン レベルの管理者権限が必要です。
- Silk Test Workbench を実行するには、次のフォルダとそのすべてのサブフォルダに対するフルアクセス権が必要です。
 - C:\ProgramData\Silk\SilkTest
 - %APPDATA%\Roaming\Silk\SilkTest
 - %APPDATA%\Local\Silk\SilkTest
 - %TEMP%

Silk Test Workbench データベースの構成


Silk Test Workbench は、資産や設定を格納するためにデータベースを必要とします。デフォルトの Silk Test Workbench インストールでは、単一ユーザーによる運用用に Microsoft Access データベースが含まれています。本格的なエンタープライズでの運用には、Silk Test Workbench は Microsoft SQL Server および Oracle データベースをサポートします。Microsoft SQL Server または Oracle データベースを Silk Test Workbench で使用するには、中央サーバー マシン上にデータベースを作成して構成し、各クライアント マシン上で中央データベースを対象としたデータ ソース名 (DSN) を作成します。同じプロジェクトに複数のユーザーが関与する長期間にわたる自動化プロジェクトには、Microsoft SQL Server を使用することを Micro Focus は推奨しています。

サポートするデータベースについての情報は、『[リリース ノート](#)』を参照してください。

 **注:** Silk Test Workbench の新しいバージョンをインストールしたときにデータベースが更新された場合、このデータベースを使用するすべての Silk Test Workbench ユーザーが、Silk Test Workbench の新しいバージョンにアップグレードする必要があります。

 **注:** Silk Test では、64 ビットの DSN は使用できません。64 ビット マシンのデフォルトの DSN と ODBC ドライバは、C:\Windows\SysWow64 にある WOW64 ツールを使用して確認できます。

SQL Server データベースの構成

 **重要:** データベースの設定と構成は、データベース管理者またはデータベースの管理に関する一般的な知識を持つ方が行うことをお勧めします。


Silk Test Workbench は、Silk Test Workbench で使用するために構成されたデータベースにテスト資産を格納してアクセスします。このセクションでは、以下の SQL Server データベースを Silk Test Workbench データベースとして使用する場合の構成方法について説明します。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリース ノート](#)』を参照してください。

SQL Server の要件

Silk Test Workbench で使用するよう SQL Server データベースを構成する場合は、以下の要件があります。

- Silk Test Workbench を実行するすべてのコンピュータは、SQL Server データベースが常駐するコンピュータにアクセスでき、新しいデータ ソースが構成されている必要があります。
- Silk Test Workbench は、データベースにアクセスするすべてのコンピュータにインストールされている必要があります。
- Silk Test Workbench データベース接続ファイルが、SQL Server データベースを指すように構成されている必要があります。
- SQL Server データベースとの接続に使用するユーザーは、データベースの VIEW_SERVER_STATE 権限を持つ必要があります。

 **注:** Silk Test では、64 ビットの DSN は使用できません。64 ビット マシンのデフォルトの DSN と ODBC ドライバは、C:\Windows\SysWow64 にある WOW64 ツールを使用して確認できます。

Silk Test Workbench データベースの保守の詳細については、*Silk Test Workbench* のヘルプを参照してください。

SQL Server データベースの新規作成

このセクションでは、SQL Server Management Studio を使用して SQL Server データベースを作成する方法について説明します。以下の手順以外に、Silk Test Workbench データベース メンテナンス ユーティリティを使用して、データベースに Silk Test Workbench テーブルを入力する必要もあります。

1. SQL Server Management Studio のオブジェクト エクスプローラで、**データベース** フォルダを右クリックして **新しいデータベース** を選択します。 **新しいデータベース** ダイアログ ボックスが開きます。
2. **データベース名** テキスト ボックスにデータベースの名前を入力します。
3. **OK** をクリックします。

SQL Server 管理ユーザーの新規作成

これ以降の設定手順を実行するには、システム管理者権限を持つ SQL Server ユーザーが必要です。

このユーザーには次のアクセス許可が必要です。

- sysadmin または、db_ddladmin と db_owner の両方
- db_datareader
- db_datawriter

これらのアクセス許可は、**CREATE INDEX (Transact-SQL)** 文を使用してインデックスを作成するために必要になります。Silk Test Workbench データベースが作成された後は、データベースに接続するすべてのユーザーに対して、db_datareader と db_datawriter アクセス許可だけが必要になります。


1. システム管理者権限を持つユーザーを新規作成するには、以下のステップを実行します。
 - a) SQL Server Management Studio のオブジェクト エクスプローラで、**セキュリティ** フォルダに移動して、展開します。
 - b) **ログイン** フォルダを右クリックして、**新しいログイン** をクリックします。 **ログイン - 新規作成** ダイアログ ボックスが開きます。
 - c) **全般** ページを選択し、**ログイン名** テキスト ボックスにユーザー名を入力します。
 - d) [SQL Server 認証] を選択し、パスワードを入力します。
 - e) **既定のデータベース** リストから、デフォルト データベースを選択します。
 - f) **サーバー ロール** ページを選択し、**サーバー ロール** リストで **sysadmin** のチェック ボックスをオンにします。
 - g) **ユーザー マッピング** ページを選択します。
 - h) **マップ** 列で、新しいログインでアクセスできるデータベースのチェック ボックスをオンにします。デフォルトで、**ユーザー** 列にログイン名が表示されています。この値はそのままにします。

- i) データベース ロールのメンバシップ リストで、**db_owner** のチェック ボックスをオンにします。
 - j) **OK** をクリックします。
2. 新しいユーザーのスキーマを新規作成するには、以下のステップを実行します。
- a) **オブジェクト エクスプローラ** で、スキーマを作成するデータベースに移動します。
 - b) データベース オブジェクト ツリーを展開して、**スキーマ** フォルダを表示します。
スキーマ フォルダは、データベースの **セキュリティ** フォルダの子です。
たとえば、スキーマをマスタ データベースに追加する場合は、**データベース > システム データベース > マスタ > セキュリティ** フォルダを展開して **スキーマ** フォルダを表示します。
 - c) **スキーマ** フォルダを右クリックして、**新しいスキーマ** を選択します。
 - d) **スキーマ名** テキスト ボックスに、新しいスキーマの名前を入力します。
新しいスキーマ名は、以前にシステム管理者権限を使用して作成したユーザー名と一致する必要があります。
 - e) スキーマの所有者として新しいユーザーにシステム管理者権限を割り当てます。
 - f) **OK** をクリックします。
3. データベースにスキーマを追加するには、次の操作を実行します。
- a) データベース オブジェクト ツリーを展開して、**ユーザー** フォルダを表示します。
ユーザー フォルダは、データベースの **セキュリティ** フォルダの子です。
たとえば、*master* データベースにスキーマを追加する場合は、**データベース > システム データベース > master > セキュリティ > ユーザー** をクリックします。
 - b) 作成した管理ユーザーをダブルクリックします。**データベース ユーザー** ダイアログ ボックスが表示されます。
 - c) **所有されているスキーマ** ページを選択し、作成したスキーマが新しいデータベースに割り当てられるようにチェックします。


SQL Server におけるユーザーの設定


以下の手順は、SQL Server データベースに接続する必要がある各ユーザーに対し、データベース サーバー コンピュータで実行してください。

1. SQL Server Management Studio のオブジェクト エクスプローラで、**セキュリティ** フォルダに移動して、展開します。**ログイン** フォルダを右クリックして、**新しいログイン** をクリックします。**ログイン-新規作成** ダイアログ ボックスが開きます。
2. **全般** ページを選択し、**ログイン名** テキスト ボックスにデータベースの名前を入力します。
3. **Windows 認証** または **SQL Server 認証** を選択します。
4. **既定のデータベース** リストから、デフォルト データベースを選択します。
5. **ユーザー マッピング** ページを選択します。
6. **マップ** 列で、ユーザーのログインでアクセスできるデータベースのチェック ボックスをオンにします。
デフォルトで、**ユーザー** 列にログイン名が表示されています。この値はそのままにします。
7. **既定のスキーマ** 列に、既定のスキーマを入力します。
Silk Test Workbench で既定のスキーマを使用するには、既定のスキーマが、以前にシステム管理者権限を使用して作成したユーザーのスキーマと一致する必要があります。
8. **データベース ロールのメンバシップ** リストで、デフォルト オプション **public** を選択したままにしておきます。
9. **db_datareader** および **db_datawriter** チェック ボックスをオンにします。
10. **OK** をクリックします。

 **重要:** 複数のユーザーが SQL Server Express を使用するには、各 SQL Server Express インストールでリモート接続が有効になっている必要があります。SQL Server Express でリモート接続を有効にする方法については、Microsoft のサポート技術情報『*How to enable remote connections on SQL Server*』を参照してください。


SQL Server Silk Test Workbench データベースの準備

 **注:** SQL Server 認証モードを SQL Server と Windows (混合モード) に設定して、データベース メンテナンス ユーティリティを使用して SQL Server データベースに接続できるようにする必要があります。この設定は、データベース メンテナンス タスクを実行したあと変更できます。

 **注:** SQL Server データベースが空の場合は、このトピックで説明しているタスクのみ実行できます。

Silk Test Workbench で使用する新しい SQL Server データベースを準備する必要があります。このセクションでは、データベース メンテナンス ユーティリティを使用してデータベースに Silk Test Workbench テーブルを入力する方法について説明します。


1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > 新規作成 > SQL Server** をクリックします。 **SQL Server データ ソースの接続** ダイアログ ボックスが開きます。
3. 新しい SQL Server データ ソースの名前を入力するか、**参照** をクリックして **ODBC データ ソースの選択** ダイアログ ボックスからデータ ソース名を選択します。
4. 該当するテキスト ボックスにデータベースの所有者の名前と、SQL Server のユーザー ID およびパスワードを入力し、**作成** をクリックします。データベースの所有者は、db_owner ロールを持つユーザーです。

 **注:** 各 DSN 名に関連付けることができるスキーマは 1 つだけです (ODBC 接続)。選択した DSN 名がすでにスキーマに関連付けられている場合は、**置換** をクリックして DSN 名に関連付けを現在のスキーマから新しいスキーマに変更するか (ユーザーは、古いスキーマの Silk Test Workbench テーブルにはアクセスできなくなります)、**エイリアス** をクリックして DSN 名のエイリアスを作成します。

これで、Silk Test Workbench を起動してログオンする準備が整いました。

Oracle データベースの構成

このセクションでは、Silk Test Workbench で使用する Oracle データベースを設定する方法について説明します。

 **注:** 以下の手順は、その実行者が Oracle データベース管理システムに精通していることを前提としています。データベースの設定と構成は、データベース管理者またはデータベースの管理に関する一般的な知識を持つ方が行うことをお勧めします。

各クライアント コンピュータでは、データベースに接続するために ODBC データ ソースが必要となります。この結果、Silk Test Workbench 用の Oracle データベースに接続するには、以下の条件を満たす必要があります。

- Silk Test Workbench を実行するすべてのコンピュータは、データベースが常駐するコンピュータにアクセスでき、新しいデータ ソースが構成されている必要があります。
- Silk Test Workbench は、データベースにアクセスするすべてのコンピュータにインストールされている必要があります。
- Silk Test Workbench を実行するすべてのコンピュータに、適切なデータベース クライアント接続ソフトウェアがインストールされている必要があります。
- データベース接続は、Silk Test Workbench で使用するために構成する必要があります。

認証方式の選択

Silk Test Workbench では、Oracle OS 認証 (Windows NT) と Oracle データベース認証のいずれかを使用することができます。Oracle OS 認証を使用する場合、追加のインストール手順が必要になります。

Oracle OS 認証

このトピックでは、Oracle OS 認証を使用する場合に、ユーザーを作成して接続権限を付与する方法について説明します。これらのステップを実行する前に、以下の点を考慮してください。


- ORA_DBA グループのメンバであり、かつ、SYSDBA 権限がある必要があります。
- ユーザーを作成する際は、ユーザーが DOMAIN コントローラによって認証されていることを Oracle が認識できるようにする必要があります。
- ユーザーの作成には SQL*Plus または SQL*Plus Worksheet を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。
- ユーザーは Windows NT ドメイン内に作成し、ORA_DBA グループに追加する必要があります。Windows ドメインの使用の詳細については、Microsoft Windows のマニュアルを参照してください。
- Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマを作成します。このスキーマはドメインで認証されず、パスワードで認証されます。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを "DOMAIN¥USERNAME" IDENTIFIED EXTERNALLY として作成し、接続権限およびシステム権限を付与するコマンドを、以下の例に示すように入力します。

各ユーザーは "DOMAIN¥USERNAME" のように入力します。DOMAIN¥USERNAME は大文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのドメインとユーザー名です。

例：

```
CREATE USER "NT-DOMAIN¥JEFF" IDENTIFIED EXTERNALLY;  
GRANT SELECT ANY SEQUENCE TO "NT-DOMAIN¥JEFF";  
GRANT UNLIMITED TABLESPACE TO "NT-DOMAIN¥JEFF";  
GRANT "CONNECT" TO "NT-DOMAIN¥JEFF";  
GRANT "RESOURCE" TO "NT-DOMAIN¥JEFF";  
GRANT "SELECT_CATALOG_ROLE" TO "NT-DOMAIN¥JEFF";
```

 **重要:** Microsoft のオペレーティング システムでは、ユーザーを照会すると DOMAIN ¥USERNAME が返されます。この値がデータベースの定義と一致しない場合は、Oracle OS 認証で ORA-1017 のエラーが発生します。

4. SQL*Plus を使用してデータベースに接続することにより、Oracle OS 認証が正しく設定されているかどうかをテストします。
 - a) ユーザー名には「/」と入力します。
 - b) [パスワード] テキスト ボックスは空のままにし、ホスト文字列を正しく入力します。
 - c) **OK** をクリックします。ログオンできた場合は、Oracle OS 認証が正しく機能していることを意味します。

Oracle データベース認証

このトピックでは、Oracle データベース認証を使用する場合に、ユーザーを作成して接続権限を付与する方法について説明します。これらのステップを実行する前に、以下の点を考慮してください。

- ORA_DBA グループのメンバであり、かつ、SYSDBA 権限がある必要があります。
- ユーザーの作成には SQL*Plus または SQL*Plus Worksheet を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。
- Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマを作成します。このスキーマはドメインで認証されず、パスワードで認証されます。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを "USERNAME" IDENTIFIED BY "ORACLEPASSWORD" として作成し、接続権限およびシステム権限を付与するコマンドを、以下の例に示すように入力します。

各ユーザーは "USERNAME" のように入力します。USERNAME は大文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのユーザー名です。

例：

```
CREATE USER "JEFF" IDENTIFIED BY "ORACLEPASSWORD";
GRANT SELECT ANY SEQUENCE TO "JEFF";
GRANT UNLIMITED TABLESPACE TO "JEFF";
GRANT "CONNECT" TO "JEFF";
GRANT "RESOURCE" TO "JEFF";
GRANT "SELECT_CATALOG_ROLE" TO "JEFF";
```


Oracle データベースの新規作成

次の文字セットを使用する新しい Oracle データベースを作成します。

- UTF8：Oracle 12 より前のバージョンの Oracle
- AL32UTF8：Oracle 12 以降


詳細については、Oracle の製品マニュアルを参照してください。

Oracle OS 認証の設定

 **注：** この手順は、Oracle OS 認証を使用する場合にのみ必要です。Oracle データベース認証を使用する場合は、Oracle クライアントをセットアップする必要があります。


1. データベースがあるサーバーで、Oracle をインストールしたディレクトリにある Oracle 初期化ファイルを探します。
2. Oracle 初期化ファイルで、以下のパラメータの値を設定します。

- remote_login_passwordfile = none
- remote_os_authent = true
- os_authent_prefix = ""

 **注：** パラメータと値のうち存在しないものがある場合は、追加する必要があります。Oracle 11g では、remote_os_authent および os_authent_prefix パラメータはデフォルトで設定されています。

3. Oracle をインストールしたディレクトリで、SQLNET.ORA ファイルを探します。
このファイルは、Oracle のメイン インストール ディレクトリの下に ORACLE_HOME¥Network ¥Admin ディレクトリにあります。ORACLE_HOME は、インストール時に Oracle ホームに割り当てた名前です。
4. SQLNET.ORA を開いて、以下のようにパラメータの値を設定します。

```
sqlnet.authentication_services = (NTS)
```

 **注：** このパラメータの行頭が # でコメントアウトされていないことを確認します。このパラメータは、追加する必要がある場合があります。

5. **スタート** メニューから、**ファイル名を指定して実行** ダイアログ ボックスにアクセスします。
6. **名前** テキスト ボックスに「regedit」と入力して **OK** をクリックします。**レジストリ エディタ** ダイアログ ボックスが開きます。
7. HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥HOMEn (n は Oracle インストールに関連付けられた番号) パスで、HOMEn を右クリックして、**新規 > 文字列値** を選択します。レジストリ エディタの右側のウィンドウに、新しい文字列の値が表示されます。
8. デフォルト名の新しい値 #1 を OSAUTH_PREFIX_DOMAIN に置き換えます。
9. 作成した文字列の値をダブルクリックします。**文字列の編集** ダイアログ ボックスが開きます。
- 10 **値のデータ** テキスト ボックスに「TRUE」と入力して **OK** をクリックします。

Oracle クライアントの設定

接続元として使用するクライアント コンピュータで、以下を実行します。

1. Oracle をインストールしたディレクトリで、SQLNET.ORA ファイルを探します。

このファイルは、Oracle のメイン インストール ディレクトリの下に ORACLE_HOME¥Network ¥Admin ディレクトリにあります。ORACLE_HOME は、インストール時に Oracle ホームに割り当てた名前です。

2. SQLNET.ORA を開いて、以下のようにパラメータの値を設定します。

```
sqlnet.authentication_services = (NTS)
```



注: このパラメータの行頭が # でコメントアウトされていないことを確認します。このパラメータは、追加する必要がある場合があります。

3. Oracle をインストールしたディレクトリで、TNSNAMES.ORA ファイルを開きます。

このファイルは ORACLE_HOME¥Network¥Admin ディレクトリにあります。

4. サーバーのデータベース エントリをクライアント コンピュータ上の TNSNAMES.ORA ファイルにコピーします。

例:

```
ORACLE_SID.DOMAIN.COM=  
(DESCRIPTION=  
(ADDRESS_LIST=  
(ADDRESS=(PROTOCOL=TCP)(HOST=SERVERNAME)(PORT=1521))  
)  
(CONNECT_DATA=  
(SERVICE_NAME=ORACLE_SID)  
)  
)
```

クライアント接続用の Oracle データベースの準備



注: この手順は、Oracle OS 認証および Oracle クライアントを使用して Silk Test Workbench データベースを設定する場合にのみ必要です。Oracle データベース認証を使用する場合は、ユーザーの設定を開始することができます。

1. ドメイン ユーザー マネージャがインストールされているコンピュータで、Windows の **コントロールパネル** にアクセスし、**管理ツール > コンピュータの管理** を選択します。

コンピュータの管理 ユーティリティが開きます。

2. ツリー ビューで、**ローカル ユーザーとグループ** をクリックします。

3. **グループ** をクリックします。

Oracle がインストールされている場合、ORA_DBA という名前のグループが表示されます。表示されない場合は、追加する必要があります。

次に、ユーザーに対してサーバー アクセスを作成します。

ORA_DBA グループの作成

1. **コンピュータの管理** ユーティリティのツリー ビューで、**グループ** を右クリックして **新しいグループ** を選択します。 **新しいグループ** ダイアログ ボックスが開きます。
2. **グループ名** テキスト ボックスに「ORA_DBA」と入力します。
3. **作成** をクリックします。
4. **閉じる** をクリックします。

ユーザーに対するサーバー アクセスの作成

ユーザー アカウントを個別に作成するには、以下の手順に従います。

1. **コンピュータの管理** ユーティリティのツリー ビューで、**ローカル ユーザーとグループ** をクリックします。
2. **グループ** をクリックします。
3. 右側のペインで、**ORA_DBA** をダブルクリックします。 **ORA_DBA のプロパティ** ダイアログ ボックスが開きます。
4. **追加** をクリックします。 **ユーザー、コンピュータ、またはグループの選択** ダイアログ ボックスが開きます。
5. **詳細設定** をクリックし、**共通クエリ** タブを使用してユーザーを検索し、**今すぐ検索** をクリックして、リストからユーザーを選択します。

または、**選択するオブジェクト名を入力してください** テキスト ボックスをクリックし、domain name ¥user ID を入力します。

domain name はユーザーが所属するドメインの名前であり、user ID はアクセス権を付与する対象ユーザーのネットワーク ID です。

6. **OK** をクリックしてユーザー情報を保存し、**ORA_DBA のプロパティ** ダイアログ ボックスに戻ります。ステップ 4 から 6 を繰り返して、その他のユーザー アカウントを作成します。
7. **Users のプロパティ** ダイアログ ボックスで **OK** をクリックし、**コンピュータの管理** ユーティリティに戻ります。

Oracle における Silk Test Workbench のユーザーの設定

このセクションでは、ユーザーを作成してユーザーの接続権限を付与する方法について説明します。この手順は Oracle OS 認証または Oracle データベース認証のどちらを使用する場合にも必要です。手順は認証タイプによって異なります。

Oracle OS 認証

ユーザーを作成する際は、ユーザーが DOMAIN コントローラによって認証されていることを Oracle が認識できるようにする必要があります。



重要:

Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマも作成する必要があります。このスキーマはドメインで認証されず、パスワードで認証されます。詳細については、「[パスワード認証スキーマの作成](#)」を参照してください。

ユーザーは Windows NT ドメイン内に作成する必要があります。Windows ドメインの使用の詳細については、Microsoft Windows のマニュアルを参照してください。

ユーザーは ORA_DBA グループに追加する必要があります。ユーザーの作成には SQL*Plus または SQL*Plus Worksheet を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。

以下の手順を実行するユーザーは、ORA_DBA グループのメンバーであり、SYSDBA 権限を持っている必要があります。Oracle データベースにアクセスできるユーザーを指定するには、以下を実行します。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを作成するコマンドを "DOMAIN¥USERNAME" IDENTIFIED EXTERNALLY として入力し、各ユーザーの接続権限を "DOMAIN¥USERNAME" として付与します。DOMAIN¥USERNAME は大

文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのドメインとユーザー名です。例：

```
SQL> create user "NT-DOMAIN¥JEFF" IDENTIFIED EXTERNALLY;
SQL> grant connect, resource to "NT-DOMAIN¥JEFF";
```


 **注:** SELECT_ANY_SEQUENCE と SELECT_CATALOG_ROLE もユーザーに付与する必要があります。例：

```
SQL> grant select any sequence to "JEFF";
SQL> grant select_catalog_role to "JEFF";
```

Microsoft のオペレーティング システムでは、ユーザーを照会すると DOMAIN¥USERNAME が返されます。この値がデータベースの定義と一致しない場合は、NT 認証で ORA-1017 のエラーが発生します。

4. SQL*Plus を使用してデータベースに接続することにより、Windows NT 認証が正しく設定されているかどうかをテストします。ユーザー名には「/」と入力します。パスワード フィールドは空のままにし、ホスト文字列を正しく入力します。**OK** をクリックします。ログオンできた場合は、Windows NT 認証が正しく機能していることを意味します。

Oracle データベース認証

 **重要:** Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマも作成する必要があります。スキーマはパスワード認証されます。詳細については、「[パスワード認証スキーマの作成](#)」を参照してください。

ユーザーの作成には SQL*Plus を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。以下の手順を実行するユーザーは、ORA_DBA グループのメンバーであり、SYSDBA 権限を持っている必要があります。

Oracle データベースにアクセスできるユーザーを指定するには、以下を実行します。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを作成するコマンドを "USERNAME" IDENTIFIED BY "ORACLEPASSWORD" として入力し、各ユーザーの接続権限を "USERNAME" として付与します。USERNAME は大文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのユーザー名です。例：


```
SQL> create user "JEFF" IDENTIFIED BY "ORACLEPASSWORD";
SQL> grant connect, resource to "JEFF";
```

 **注:** SELECT_ANY_SEQUENCE、SELECT_CATALOG_ROLE、SELECT_ANY_DICTIONARY もユーザーに付与する必要があります。例：

```
SQL> grant select any sequence to "JEFF";
SQL> grant select_catalog_role to "JEFF";
SQL> grant select any dictionary to "JEFF";
```

パスワード認証スキーマの作成

Silk Test Workbench テーブルが格納されるパスワード認証スキーマを設定するには、以下のステップを実行します。

 **注:** SQL シリーズにおける Oracle の文字数制限のため、パスワード認証スキーマは 5 文字以下で指定する必要があります。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを作成するコマンドを "USERNAME" IDENTIFIED BY "ORACLEPASSWORD" として入力し、各ユーザーの接続権限を "USERNAME" として付与します。

USERNAME は大文字で入力し、二重引用符で囲みます。USERNAME は各ユーザーのユーザー名です。例：

```
SQL> create user "TOM" IDENTIFIED BY "ORACLEPASSWORD";  
SQL> grant connect, resource to "TOM";
```

Oracle データベースの準備

Silk Test Workbench で使用する新しい Oracle データベースを準備する必要があります。このセクションでは、**データベース メンテナンス** ユーティリティを使用してデータベースに Silk Test Workbench テーブルを入力する方法について説明します。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。**データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > 新規作成 > Oracle** をクリックします。**Oracle データ ソースの接続** ダイアログ ボックスが開きます。
3. 新しい Oracle データ ソースの名前を入力するか、**参照** をクリックして **ODBC データ ソースの選択** ダイアログ ボックスからデータ ソース名を選択します。
4. 該当するボックスに、Oracle ユーザー ID の名前、パスワードを入力し、**作成** をクリックします。**スキーマ** ボックスには、ユーザー ID に入力した値が自動的に入力されます。

これで、Silk Test Workbench を起動してログオンする準備が整いました。


ドメインなしでの SQL Server または Oracle データベースの設定

Silk Test Workbench で使用する Oracle または SQL Server のデータベースを設定する際は、ドメイン内の認証を使用することを強くお勧めします。ただし、ドメインに所属しない場合でも、これらのデータベースを設定することは可能です。

このセクションでは、ドメインを使用して SQL Server または Oracle データベースを設定する具体的な方法について説明します。

ドメインなしでのユーザーの作成

ドメインがない場合に SQL Server または Oracle データベースを設定するには、最初にクライアント コンピュータとサーバー コンピュータの両方でユーザー アカウントを作成する必要があります。

 **注:** この手順は Windows NT 認証を使用する場合にのみ適用されます。

1. Windows の **コントロール パネル** にアクセスし、**管理ツール > コンピュータの管理** を選択します。**コンピュータの管理** ユーティリティが開きます。
2. 左側のペインで、**ローカル ユーザーとグループ** をクリックします。
3. **ローカル ユーザーとグループ** で、**ユーザー** フォルダを右クリックし、**新しいユーザー** を選択します。**新しいユーザー** ダイアログ ボックスが開きます。
4. **ユーザー名** テキスト ボックスに有効なユーザー名を入力し、**フル ネーム** テキスト ボックスにそのユーザーのフル ネームを入力します。
5. **パスワード** と **パスワードの確認入力** の両方のテキスト ボックスに有効なパスワードを入力し、**作成** をクリックします。
6. 各クライアント コンピュータでそのユーザー名とパスワードを使用して、上記の手順を繰り返します。

ドメインなしでの SQL Server の設定

ドメインがない場合に SQL Server を設定するには：

- Windows の **コントロール パネル** にアクセスし、**管理ツール > コンピュータの管理** を選択します。
コンピュータの管理 ユーティリティが開きます。
- 左側のペインで、**ローカル ユーザーとグループ** を展開し、**グループ** をクリックします。
- 右側のペインで、**Users** をダブルクリックします。
Users のプロパティ ダイアログ ボックスが開きます。
- 追加** をクリックします。
ユーザー、コンピュータ、またはグループの選択 ダイアログ ボックスが開きます。
- 詳細設定** をクリックし、**共通クエリ** タブを使用してユーザーを検索します。
- 今すぐ検索** をクリックして、リストからユーザーを選択します。
または、**選択するオブジェクト名を入力してください** テキスト ボックスをクリックし、アクセス権を付与する対象ユーザーの [user ID] を入力します。[user ID] :
- OK** をクリックしてユーザー情報を保存し、**Users のプロパティ** ダイアログ ボックスに戻ります。
- 上記のステップを繰り返して、その他のユーザー アカウントを作成します。
- OK** をクリックします。

ドメインなしでの Oracle データベースの設定

ドメインがない場合に Oracle を設定するには、個別のユーザー アカウントに対してサーバー アクセスを作成する手順に従います。ただし、ドメインなしでユーザーを作成する手順で作成したユーザー名を置き換えます。

- 選択するオブジェクト名を入力してください** テキスト ボックスに、[domain name]¥[user ID] や [domain name]¥[group name] ではなくユーザー名を入力します。

コンピュータの管理 ユーティリティが表示されます。

- ユーザーを設定します。

SQL*Plus を使用してユーザーを作成する場合は、以下の例に示すように [domain name¥user name] を [computer name¥user name] に置き換えます。

```
SQL> create user "FH0002RED¥USER1" IDENTIFIED EXTERNALLY;
SQL> grant connect, resource to "FH0002RED¥USER1";
```



注:

[computer name¥user name] の場合、大文字を使用して引用符で囲みます。

データ ソース名の作成

Silk Test Workbench 用にセットアップしたデータベースにアクセスするために、データベースに接続する各マシン上で、ODBC ユーティリティを使用して Silk Test Workbench データベースに対するデータ ソース名 (DSN) を作成します。



注: このステップは、Silk Test Workbench インストーラによってインストールされたデフォルトの Access データベースには適用されません。これらのデータベースでは、データ ソース名が自動的に作成されます。


Access データベース用のデータ ソース名の作成

Access データベースに接続する各クライアント コンピュータで、以下の手順に従って DSN を設定します。

- Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データ ソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データ ソース (32 ビット)** をクリックします。**ODBC データ ソース アドミニストレータ** ダイアログ ボックスが開きます。

2. **システム DSN** タブをクリックし、**追加** をクリックします。 **データソースの新規作成** ダイアログボックスが開きます。このダイアログボックスには、使用可能な ODBC ドライバがリストされます。
3. ドライバのリストから **Microsoft Access Driver** を選択し、**完了** をクリックします。 **ODBC Microsoft Access セットアップ** ダイアログボックスが開きます。
4. **データソース名** テキストボックスに、データソース名を入力します。
この名前はログオンダイアログボックスでデータベース名として表示されるため、論理名を入力します。
5. **説明** テキストボックスに、データソースの説明を入力します。
たとえば、「Silk Test Workbench データベースへの接続」です。
6. **選択** をクリックして、Access データベースを参照します。
7. **ODBC Microsoft Access セットアップ** ダイアログボックスで、**OK** をクリックします。
8. **ODBC アドミニストレータ** ダイアログボックスで、**OK** をクリックしてプロセスを完了します。

SQL Server データベース用のデータソース名の作成

 **注:** Silk Test Workbench では、64 ビットの DSN は使用できません。64 ビットマシンで DSN を作成するには、Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックして、32 ビット DSN を作成します。C:\%WINDOWS%\SysWOW64\odbcad32.exe にある WOW64 ツールを使用することもできます。

SQL Server データベースに接続する各クライアントコンピュータで、以下の手順に従って DSN を設定します。

1. Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックします。 **ODBC データソース アドミニストレータ** ダイアログボックスが開きます。
2. **システム DSN** タブをクリックし、**追加** をクリックします。 **データソースの新規作成** ダイアログボックスが開きます。このダイアログボックスには、使用可能な ODBC ドライバがリストされます。
3. ドライバのリストから **DBC Driver 11 for SQL Server** を選択し、**完了** をクリックします。 **SQL Server に接続するための新規データソースを作成する** ダイアログボックスが開きます。
4. **名前** ボックスに、データソース名を入力します。
この名前はログオン画面でデータベース名として表示されるため、論理名を入力します。
5. **説明** テキストボックスに、データソースの説明を入力します。
たとえば、「Silk Test Workbench データベースへの接続」です。
6. サーバー名を **サーバー** テキストボックスに入力するか、ドロップダウンリストから選択します。
7. **次へ** をクリックします。
8. 以下のいずれかを実行します。
 - DSN を使用して SQL Server データベースにアクセスするユーザーが非ネイティブのアカウント情報によって接続している場合、**ネットワークへのログイン ID で、Windows の認証メカニズムを使う** オプションを選択します。
 - DSN を使用して SQL Server データベースにアクセスするユーザーがネイティブ認証によって接続している場合、**ユーザーが入力する SQL Server 用のログイン ID とパスワードを使う** オプションを選択します。 **SQL Server に接続して追加の構成オプションの既定設定を取得する** チェックボックスがオンになっていることを確認してください。ユーザーのネイティブ SQL Server ログオン情報を、**ログイン ID とパスワード** の各テキストボックスに入力します。
9. **次へ** をクリックします。
- 10 **既定のデータベースを以下のものに変更する** チェックボックスをオンにして、リストからデータベースの名前を選択します。

11. その他のエントリはすべてそのままにし、ウィザードの最後のパネルが表示されるまで **次へ** をクリックします。

12. デフォルトをそのまま使用し、**完了** をクリックします。


13. 接続をテストするには、**データソースのテスト** をクリックします。**SQL Server ODBC データソーステスト** ダイアログ ボックスが開きます。

14. **OK** をクリックします。

ODBC データソースアドミニストレータ ダイアログ ボックスが再表示されます。新しく作成したデータソースが **システムデータソース** リストに表示されます。

15. **OK** をクリックします。

Oracle データベース用のデータソース名の作成

 **注:** Silk Test Workbench では、64 ビットの DSN は使用できません。64 ビットマシンで DSN を作成するには、Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックして、32 ビット DSN を作成します。C:\%WINDIR%\SysWOW64\odbcad32.exe にある WOW64 ツールを使用することもできます。

Oracle データベースに接続する各クライアント コンピュータで、以下の手順に従って DSN を設定します。

1. Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックします。**ODBC データソースアドミニストレータ** ダイアログ ボックスが開きます。

2. **システム DSN** タブをクリックし、**追加** をクリックします。**データソースの新規作成** ダイアログ ボックスが開きます。このダイアログ ボックスには、使用可能な ODBC ドライバがリストされます。


3. ドライバのリストから、使用する Oracle データベースのバージョンに適した Oracle ODBC ドライバ (Microsoft ODBC for Oracle ドライバではない) を選択し、**完了** をクリックします。**Oracle ODBC ドライバ構成** ダイアログ ボックスが開きます。

4. **Oracle データソース名** テキスト ボックスに、データソース名を入力します。

この名前はログオン ダイアログ ボックスでデータベース名として表示されるため、論理名を入力します。


5. **説明** テキスト ボックスに、データソースの説明を入力します。
たとえば、「Silk Test Workbench データベースへの接続」です。

6. **TNS サービス名** テキスト ボックスで、接続するデータベースのサービス名を ORACLE_TNS.DOMAIN の形式で選択してください。大抵の場合、ORACLE_TNS は、作成時にデータベースに割り当てられた Oracle SID であり、DOMAIN は、接続するドメインになります。

 **注:** ODBC を使用して Silk Test Workbench に接続するには、TNS サービス名は必須です。

7. Oracle 認証を使用する場合は、接続先のデータベースの有効なユーザー ID を **ユーザー ID** テキスト ボックスに入力します。それ以外の場合は、**ユーザー ID** テキスト ボックスは空のままにします。

8. 接続をテストするには、**接続テスト** をクリックします。

 **注:** Oracle データベース認証を使用する場合、ユーザー名を入力してから、パスワードを入力します。Oracle OS 認証では、ユーザー名を指定しないでください。

正常に接続されたことを示すメッセージが表示されます。

9. **OK** をクリックします。


データベースへ接続する

Silk Test Workbench をインストールしてデータベースを構成したあと、データベース接続を作成する必要があります。この接続によって、Silk Test Workbench データベースと Silk Test Workbench インストー


ルとの関係を定義します。このセクションでは、ログオンしてデータベース接続を作成する方法について説明するトピックを示します。

ログオン

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > クライアント > Silk Test Workbench**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Workbench** をクリックします。Silk Test Workbench が起動され、**SilkTest Workbench にログイン** ダイアログ ボックスが表示されます。
2. **認証** リストボックスから、**Silk Test Workbench 認証**、または **Windows 認証**のいずれかを使用する方を選択します。

 **注:** **Windows** 認証を使用できるユーザーは、**ツール > 管理** を使用して Silk Test Workbench ユーザーに追加されたユーザーだけです。

3. **Silk Test Workbench 認証** を使用している場合、**ユーザー名** フィールドにユーザー名を入力し、**パスワード** フィールドにパスワードを入力します。**Windows 認証** を使用している場合は、ユーザー名とパスワードを入力する必要はありません。

 **注:** Silk Test Workbench をはじめて使用する場合は、管理者権限を持つユーザー名とパスワードが必要になる場合があります。デフォルトのユーザー名は Admin で、デフォルトのパスワードは admin です。不正アクセスを防ぐため、このパスワードはログオンのあとで変更してください。ログオンパスワードはいつでも変更できます。

4. **データベース** リストから、使用するデータベースを選択します。
データベースが **データベース** リストに表示されるようにするには、まずデータベース接続を使用できるように構成する必要があります。データベース接続を構成するには、**ログオン** ダイアログ ボックスで **構成** ボタンをクリックします。
5. **OK** をクリックします。

Silk Test Workbench データベースに初めてログインすると、*Project1* という名前のデフォルトプロジェクトがデータベースに作成されます。デフォルトでは、作成する資産は、このプロジェクトに保存されます。プロジェクト名は、**プロジェクトの編集** をクリックすると変更できます。

開始画面 が開きます。Silk Test Workbench に始めてログインすると、Silk Test Workbench は *Project1* または最初に利用可能な非グローバル オブジェクトをアクティブ オブジェクトとして選択します。

ログオンパスワードの変更

ログオンパスワードを変更するには、以下の手順に従います。

1. **ツール > パスワードの変更** をクリックします。**パスワードの変更** ダイアログ ボックスが開きます。現在のユーザーのログオン名が、ダイアログ ボックスのタイトルバーに表示されます。
2. パスワードを変更し、**OK** をクリックします。このユーザー ID を使って次回ログオンしたとき、変更が有効になります。


データベース接続の構成


Silk Test Workbench でデータベースを使用する前に、データベース接続を構成する必要があります。構成処理には、**データベース接続の構成** ダイアログ ボックスを使用した以下のタスクの実行が含まれています。

- データベース接続を開始するために必要なデータベース接続データを指定します。
- データベースへの接続を検証します。
- データベース接続のデータを保存します。

既存のすべての構成済みデータベース接続は、Silk Test Workbench ダイアログ ボックスにある **データベース** リストに表示されます。

データベース接続の構成 ダイアログ ボックスは、既存の構成済みデータベース接続の表示、編集、および削除に使用することもできます。

 **注:** Silk Test Workbench で使用するデータベース接続を構成する前に、データベース インスタンスと ODBC データ ソース名 (DSN) を作成および構成しておく必要があります。

 **注:** Silk Test Workbench では、64 ビットの DSN は使用できません。64 ビット マシンで DSN を作成するには、Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データ ソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データ ソース (32 ビット)** をクリックして、32 ビット DSN を作成します。C:\%WINDOWS%\SysWOW64\odbcad32.exe にある WOW64 ツールを使用することもできます。

1. Silk Test Workbench に**ログイン** ダイアログ ボックスで、**構成** をクリックします。 **データベース接続の構成** ダイアログ ボックスが開きます。
2. **データ ソースの種類** セクションで、データ ソースの種類を選択します。
3. **データ ソースの選択** リストで、使用可能な DSN のリストから選択します。
4. 選択した DSN を Silk Test Workbench に**ログイン** ダイアログ ボックスの **データベース** リストに表示される構成済みデータベース接続のリストに追加するには、**Silk Test Workbench データベースとして使用** チェック ボックスをオンにします。
データベース接続をはじめて構成するときは、必ずこのチェック ボックスをオンにしてください。既存の構成済みデータベース接続に対して **Silk Test Workbench データベースとして使用** チェック ボックスをオフにして **適用** をクリックすると、選択した DSN がダイアログ ボックスの **データベース** リストから削除されます。Silk Test Workbench は関連するデータベース接続のデータを保持しません。
5. **データベースの設定** セクションで、選択した DSN に対して適切な情報を指定します。Access では、**データベース** ボックスは読み取り専用で、データベース ファイルの場所を示します。この値は ODBC DSN から直接読み取られます。他の **データベースの設定** テキスト ボックスは、Access には適用されず、無効になっています。Oracle では、**サーバー** テキスト ボックスは読み取り専用で、データベース名のみが表示されます。この値は ODBC DSN から直接読み取られます。
6. **データベース** ボックスに、適切な値を入力します。SQL Server/MSDE では、**データベース** テキスト ボックスは編集可能で、適切なデータベース名を入力することができます。
7. **所有者** ボックスに、適切な値を入力します。
8. 認証の種類を選択します。
認証 オプションは、Oracle データベースまたは SQL Server/MSDE データベースにのみ適用されます。Windows NT 認証を使用するか、データベースのネイティブ認証機能を使って認証するかを選択できます。データベース認証を選択する場合、適切なテキスト ボックスにユーザー名とパスワードを入力する必要があります。
9. **検証** をクリックして、指定したデータベース接続データを使用してデータベースに接続できることを確認します。
- 10 **適用** をクリックして、データベース接続データを保存します。
- 11 **閉じる** をクリックして、ダイアログ ボックスを閉じます。

データベース メンテナンス

データベース メンテナンス ユーティリティは、データベース スキーマの更新、データベースのレコードのロック解除、データベースのコンパクト化など、いくつかのメンテナンス タスクを実行します。

メンテナンス タスクを実行する前に、Silk Test Workbench データベース接続を構成する必要があります。また、Silk Test インストーラを使用せずにインストールされたデータベースでは、**データベース メンテナンス** ユーティリティを使用して新しい Silk Test Workbench ODBC データベースを作成する前に、システム管理者が新しい ODBC データ ソースを作成する必要があります。詳細については、『Silk Test インストール ガイド』を参照してください。

Silk Test Workbench データベースを開く

すでに作成されている Access、SQL Server、または Oracle Silk Test Workbench データベースを開いて、データベーススキーマの更新、データベースのレコードのロック解除、データベースのコンパクト化などのメンテナンス タスクを実行できます。

Microsoft SQL Server データベースを開く

データベース メンテナンス ユーティリティを起動して Microsoft SQL Server データベースを開くには、以下の手順に従います。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > データベースを開く > SQL Server** をクリックします。 **SQL Server データソースの接続** ダイアログ ボックスが開きます。
3. **参照** をクリックします。 **ODBC データソースの選択** ダイアログ ボックスが開きます。
4. 目的の Microsoft SQL Server DSN を選択して、**選択** をクリックします。 **SQL Server データソースの接続** ダイアログ ボックスが開き、**SQL Server データソース名** テキスト ボックスに選択した DSN が表示されます。



注: デフォルトの Silk Test Microsoft SQL Server 2008 Express データベースの DSN は Silk Test_SQL です。

5. **所有者**、**ユーザー ID**、および **パスワード** テキスト ボックスに、Silk Test Microsoft SQL Server データベース インスタンスに関する適切な情報を入力します。



注: デフォルトの Silk Test Microsoft SQL Server 2008 Express データベースの所有者は Silk Test DB であり、管理者権限を持つユーザーのユーザー名は SilkTestAdmin、パスワードは SilkTestAdmin です。

6. **開く** をクリックします。データベースが開き、**データベース メンテナンス ユーティリティの ツール** メニュー オプションが使用可能になります。選択したデータベースに現在接続しているユーザーがいる場合は、メッセージが表示されます。詳細については、「接続ユーザーがいるデータベースのメンテナンスの実行」を参照してください。

Oracle データベースを開く

データベース メンテナンス ユーティリティを起動して Oracle データベースを開くには、以下の手順に従います。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > データベースを開く > Oracle** をクリックします。
Oracle データソースの接続 ダイアログ ボックスが開きます。
3. **参照** をクリックします。 **ODBC データソースの選択** ダイアログ ボックスが開きます。
4. 目的の Oracle DSN を選択して、**選択** をクリックします。 **Oracle データソースの接続** ダイアログ ボックスが開き、**Oracle データソース名** テキスト ボックスに選択した DSN が表示されます。
5. **スキーマ**、**ユーザー ID**、および **パスワード** テキスト ボックスに、Silk Test Oracle データベース インスタンスに関する適切な情報を入力します。
6. **開く** をクリックします。データベースが開き、**データベース メンテナンス ユーティリティの ツール** メニュー オプションが使用可能になります。選択したデータベースに現在接続しているユーザーがいる場合は、Silk Test Workbench によってメッセージが表示されます。

詳細については、「接続ユーザーがいるデータベースのメンテナンスの実行」を参照してください。

Access データベースを開く

データベース メンテナンス ユーティリティを起動して Microsoft Access データベースを開くには、以下の手順に従います。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > データベースを開く > Access** をクリックします。 **Access データベースの接続** ダイアログ ボックスが開きます。
3. **参照** をクリックします。 **ODBC データ ソースの選択** ダイアログ ボックスが開きます。
4. 目的の Access DSN を選択して、**選択** をクリックします。
5. **開く** をクリックします。

詳細については、「接続ユーザーがいるデータベースのメンテナンスの実行」を参照してください。

データベースが開き、**データベース メンテナンス** ユーティリティの **ツール** メニュー オプションが使用可能になります。 選択したデータベースに現在接続しているユーザーがいる場合は、Silk Test Workbench によってメッセージが表示されます。

データベース バージョンの更新

Silk Test Workbench を最新のバージョンにアップグレードする場合、既存のデータベースを変換して新しいソフトウェアと互換性を持たせる必要があります。Silk Test Workbench では、データベースを読み込んだときに自動的にデータベース スキーマのバージョンがチェックされ、データベースのアップグレードが必要な場合は警告が表示されます。この警告は、データベース スキーマが現在の Silk Test Workbench スキーマと一致しない場合には必ず表示されます。データベースは、**データベース メンテナンス** ユーティリティを使用して変換できます。

変換速度は、変換するデータ量に関係しています。大量のデータを含むデータベースは変換に時間がかかる場合があります。変換後、現在のデータベース バージョン番号が表示されます。



注: データベースを新しいスキーマに変換する場合、Silk Test Workbench の古いバージョンでデータベースにアクセスできなくなります。つまり、すべての Silk Test Workbench ユーザーが新しいバージョンの Silk Test Workbench にアップグレードする必要があります。

1. 必要に応じて、Silk Test Workbench のログイン要求を取り消し、ユーザーがデータベースにログオンしないようにします。
2. **データベース メンテナンス** ユーティリティを起動して、データベースを開きます。 **データベース メンテナンス** ユーティリティのメイン ウィンドウが表示されます。
3. **ファイル > データベースを開く** をクリックし、データベースの種類を選択します。 **データベース接続** ダイアログ ボックスが表示されます。
4. データベース名とパスワードを入力するか、**参照** をクリックします。
5. **開く** をクリックします。
データベースの更新が必要な場合は、**ツール > 変換** をクリックしてデータベースを更新するよう求めるメッセージ ボックスが表示されます。
6. **OK** をクリックします。データベースが開き、**データベース メンテナンス** ユーティリティの **ツール** メニュー オプションが有効になります。
7. **変換** をクリックします。 **データベースの変換** ダイアログ ボックスが開き、変換前に現在のデータベースをバックアップするかどうかを尋ねられます。データベースのバックアップを選択します。
8. **今すぐ変換** をクリックして、データベースを現在のスキーマ バージョンに更新します。
9. 変換が完了したら、**閉じる** をクリックします。

データベースレコードのコピー

データベース メンテナンス ユーティリティでは、1 つのデータベースから別のデータベースにレコードをコピーすることによって、データベースの複製コピーを作成できます。たとえば、あるサイトで Silk Test Workbench 資産リポジトリとして Microsoft Access を使用しており、Silk Test Workbench のデータを Oracle データベースに格納する場合は、**データベース メンテナンス** ユーティリティを使用して、Access データベースに格納されているデータを Oracle データベースにコピーします。コピー ユーティリティを使用すると、Silk Test Workbench のすべてのデータがターゲット データベースで読み取り可能な形式に自動的にコピーされます。

ネットワーク データベースではコピー操作を実行しないでください。ネットワーク データベースをローカルのワークステーションにコピーし、コピー機能を実行してからネットワークの元の場所にデータベースを戻してください。これによって、ネットワークの障害や中断によるコピー作業への影響を防ぐことができます。

1. ユーザーがデータベースに接続していないことを確認してください。
2. **データベース メンテナンス** ユーティリティを起動して、コピー元データベースを開きます。
3. **コピー** ツールバー ボタンをクリックします。**コピー先** ダイアログ ボックスが開きます。
4. コピー先データベースの種類を選択して、**OK** をクリックします。
 - Access データベースへのコピー
 1. **DSN** テキスト ボックスで、**参照** をクリックして、コピー先データベースの DSN を選択します。**Access** テキスト ボックスに、データベースのパスと名前が表示されます。
 2. **コピー** をクリックします。**テーブルのコピー中** ダイアログ ボックスが開き、データベース コピー処理の進行状況が表示されます。
 - Microsoft SQL Server データベースのコピー
 1. Microsoft SQL Server データ ソースの名前を入力するか、または **参照** をクリックしてデータ ソース名を選択します。所有者名はデフォルトで dbo になります。
 2. ユーザー ID を入力します。ユーザーは、dbo 権限を持っている必要があります。
 3. データベース サーバー上の Silk Test Workbench データベース インスタンスのパスワードを入力します。このユーザー ID とパスワードは、データベース サーバー上の Silk Test Workbench データベース インスタンスへの接続に使用されます。
 4. **コピー** をクリックします。所有者名、ユーザー ID、パスワードがデータベース サーバーで承認された場合は、**データソースの作成** ダイアログ ボックスが開き、データベース コピー処理の進行状況が表示されます。
 - Oracle データベースへのコピー
 1. 新しい Oracle データ ソースの名前を入力するか、または **参照** をクリックして既存のデータ ソース名を選択します。**スキーマ** および **ユーザー ID** テキスト ボックスのデフォルトは **システム** であるため、デフォルト設定を使用する場合は、システム管理者または該当するパスワードを把握しているユーザーのみがメンテナンスを実行できます。
 2. **コピー** をクリックします。スキーマ名、ユーザー ID、パスワードがデータベース サーバーで承認された場合は、**データソースの作成** ダイアログ ボックスが開き、コピー処理の進行状況が表示されます。**データベース メンテナンス** ユーティリティによって、Silk Test Workbench データベース インスタンスへの必要な Silk Test Workbench テーブルのコピーが開始されます。

データベースのレコードのロック解除

資産が開いているとき、他のユーザーが同じ資産を同時に編集しないように、そのレコードはロックされます。リソースを保存するかシステムを終了すると、ロックが解除されます。システムがクラッシュした場合は、使用していたレコードが起動時にロックされたままになっている可能性があります。システムのクラッシュ後に、誤ってロックされたままになっているレコードがあると、Silk Test Workbench はロック解除を試みます。

プログラムを使用する前に、すべてのユーザーがシステムからログオフしていることを確認してください。接続しているユーザーがいる場合は、メッセージが表示されます。

1. **データベース メンテナンス** ユーティリティを起動して、データベースを開きます。
2. **レコードのロック解除** をクリックします。データベースに接続しているユーザーがいる場合は、データベース メンテナンス ユーティリティでデータベースのロック ダウンの際に接続ユーザーが発見されました。というメッセージが表示されます。
3. **OK** をクリックします。 **接続ユーザー** ダイアログ ボックスが開きます。 **ユーザー** リストに、現在接続しているユーザーのマシン名が表示されます。
4. **送信** をクリックして、これらのユーザーにログオフするよう勧告する通知を送ります。これらのユーザーがログオフしたら、ステップ 1 と 2 を繰り返し、データベースレコードのロックを解除します。



注: 接続しているユーザーにメッセージを送信するには、オペレーティング システムの Messenger サービスを有効にする必要があります。Messenger サービスを有効にするには、**コントロール パネル > 管理ツール > サービス** をクリックし、**Messenger** を選択して、サービスを開始します。

Access データベースのコンパクト化

データベースが日常的に使用されている場合は、最適なパフォーマンスを得るために定期的にデータベースのコンパクト化を行います。リソースの更新や削除を繰り返すうちに、データベース ファイルの断片化が進みます。断片化はディスクスペースを浪費し、パフォーマンスを低下させることがあります。Access データベースの場合、この手順を実行するまでは、データベースから資産が削除されてもファイルサイズの減少は登録されません。**データベース メンテナンス** ユーティリティを使用すると、不要なレコードを削除してデータをディスクの連続的な領域に書き込み直すことで、Microsoft Access データベースをコンパクト化できます。

ネットワーク データベースにはデータベースのコンパクト化処理を実行しないでください。ネットワーク データベースをローカルのワークステーションにコピーしてからデータベースのコンパクト化処理を行い、作業が終わったらネットワークの元の場所に戻してください。これによって、ネットワークの障害や中断によるデータベース メンテナンス作業への影響を防ぐことができます。

1. すべてのユーザーがシステムからログオフしていることを確認し、Silk Test Workbench を終了します。
2. **データベース メンテナンス** ユーティリティを起動して、データベースを開きます。
3. **コンパクト化** をクリックします。確認のためのダイアログ ボックスが開きます。
4. **はい** をクリックして続行します。処理が完了すると、データベースが正しくコンパクト化されたことを知らせるダイアログ ボックスが表示されます。
5. **OK** をクリックします。これで、データベースを使用できます。

データベース メンテナンス設定の構成

データベース メンテナンス ユーティリティでは、コンパクト化、変換、およびコピーの各処理中に使用される設定を構成できます。

メンテナンス システムの操作の設定 ダイアログ ボックスのバックアップ設定は、Microsoft Access データベースにのみ適用されます。Silk Test Workbench データベースとして使用する他の種類のデータベースに対してバックアップ設定を構成するには、各データベースの製品ドキュメントを参照してください。

データベースのコンパクト化や変換を行うときに、現在のバージョンをバックアップできます。**メンテナンス システムの操作の設定** ダイアログ ボックスを使用して、保持する最大バックアップ コピー数を指定します。コンパクト化または変換のバックアップ数を 3 つと指定した場合、データベースを変換またはコンパクト化するたびに、新しいバックアップが 1 つ作成されます。バックアップが 3 つ作成されたあとにもう一度変換またはコンパクト化を実行すると、一番古いバックアップが削除されて新しいバックアップが追加されます。

トランザクション中にバッファに入れるレコードの数を選択します。トランザクションとは、パフォーマンスを向上させるために大量のデータをデータベースにサイクルで書き込むプロセスです。トランザクシ


ョンがデータベースにコミットされるまで、各データベース処理がバッファに入れられます。ただし、1 つでも処理が失敗すると、そのトランザクションのすべての処理が失敗します。データをデータベースに保存する前にバッファに入れる処理の数を指定できます。

1. すべてのユーザーがシステムからログオフしていることを確認し、Silk Test Workbench を終了します。
2. **データベース メンテナンス ユーティリティ** を起動して、データベースを開きます。
3. **オプション > 構成** を選択します。**メンテナンス システムの操作の設定** ダイアログ ボックスが開きます。
4. コンパクト化および変換時に維持するバックアップ数の指定、コミットするレコード数の設定、および何秒経過するとクエリを失敗とみなすかを示す秒数の指定を行うには、**設定** タブでオプションをダブルクリックします。

接続ユーザーがいるデータベースのメンテナンスの実行

選択したデータベースに接続しているユーザーがいる場合、**データベース メンテナンス ユーティリティ** にメッセージが表示されます。また、**接続ユーザー** ダイアログ ボックスに接続ユーザーのリストが表示され、ここからこれらのユーザーに通知を送ることができます。デフォルトのメッセージを使用するか、または固有のメッセージを作成し、**送信** をクリックして、データベースからログオフするように接続ユーザーに通知します。

接続ユーザーのいるデータベースでメンテナンスを実行すると、ローカル マシンのシステムが不安定になるので、これは行わないでください。接続ユーザーがいなくなるまで待機する場合は、**一覧情報を更新** ボタンを使用して、接続ユーザーの最新のビューを表示できます。

 **注:** ユーザーがメッセージを受信するには、すべてのユーザーが Windows のメッセージ機能を実効にしている必要があります。それには、ユーザーがクライアント マシンの[サービス]にアクセスし、[Messenger]サービスをオンにします。メッセージの送信に使用される Windows のメカニズムによって複数のメッセージが送られる場合があります。

データベース整合性チェックの実行

データベース整合性チェックを正しく実行するには、Silk Test インストール ディレクトリの書き込み権が必要です。データベース整合性チェックに、Silk Test Workbench はこのディレクトリにデータベース整合性チェックの結果を書き込みます。書き込み件がない場合は、データベースがメンテナンスのためにロックされてしまう可能性があります。


データベース メンテナンス ユーティリティ を使用すると、データベースの整合性を検証して、検出されたエラーを修正できます。

1. データベースを開き、**ツール > 整合性チェックの実行** を選択します。**資産の検証中** メッセージ ボックスが表示され、整合性チェックの進行状況が示されます。エラーがない場合は、メッセージ ボックスが閉じます。エラーが見つかったら、メッセージ ボックスが表示され、結果を確認するように求められます。
2. **はい** をクリックして、整合性チェックの結果を確認します。結果はデフォルトのテキスト エディタで表示されます。整合性チェックの結果が記録されたファイルは、Silk Test の logs ディレクトリ (通常は %LOCALAPPDATA%\Silk Test\logs) に IntegrityCheck.log という名前で保存されます。

データベースの膨張の抑制

ビジュアル テストの記録と再生中、Silk Test Workbench は記録と再生の各段階で、テスト アプリケーションの画面とコントロールをキャプチャし、格納します。多数のイメージやコントロールがキャプチャされる可能性があるため、データベースのサイズが急速に増えることがあります。

データベース サイズの増加を抑えるには、記録と再生の設定で **画面キャプチャ** および **コントロールのキャプチャ** を無効にします。詳細については、「記録出力オプションの設定」および「再生結果のビジュアル テスト オプションの設定」を参照してください。

 **注:** コントロールをキャプチャすると、**画面プレビュー** からコントロールを更新したり、記録したビジュアルテストにテストアプリケーションのコントロールを挿入したりできます。また、この機能を無効にすると、選択したステップと関連するコントロールを示す強調ボックスが**画面プレビュー**で表示されなくなります。

たとえば、XML形式にエクスポートしたとき、一般的なテストアプリケーションの再生結果のサイズは、キャプチャ設定によって以下のようになります。


- 画面とコントロールの両方をキャプチャした場合：7.8MB
- 画面だけをキャプチャした場合：5.9MB
- 画面とコントロールのキャプチャを無効にした場合：382KB

プロジェクトの管理

プロジェクトは資産のコレクションであり、各プロジェクトではアクセス権限を各ユーザーに割り当てます。Silk Test Workbench 環境の管理者が行うタスクの1つに、プロジェクトリストの管理があります。資産を論理的に関連付けたプロジェクトとしてまとめて保管します。たとえば、特定のアプリケーションまたはビルドに関連付けられた資産をすべて一緒に、1つのプロジェクトに保管できます。

管理者は、プロジェクトを作成して、Silk Test Workbench データベースに追加できます。作成した新しいプロジェクトのアクセス権は、各ユーザーに設定された [デフォルトの権限] が割り当てられます。資産を別々のプロジェクトに保管すると、セキュリティのレベルを変えることができます。たとえば、プロジェクト1はフルアクセス権、プロジェクト2は読み取り専用、プロジェクト3はアクセス不可、のように設定できます。

現在のユーザーには、アクセス権を割り当てられているプロジェクトの資産だけが表示されます。資産は、**アセットブラウザ** を使用して表示および管理します。


 **注:** Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

Common プロジェクト

デフォルトのプロジェクトは Common プロジェクトです。これは、複数のプロジェクトで使用される資産の中央リポジトリとして機能します。Common プロジェクトの資産は、他のすべてのプロジェクトで使用できます。Common プロジェクトはデフォルトでアクティブプロジェクトになっています。

アクティブプロジェクト

新しく作成される資産は、アクティブプロジェクトとして設定されているプロジェクトに追加されます。アクティブに追加された資産は、そのプロジェクト内と、アクティブプロジェクトを参照しているすべてのプロジェクト内でのみ使用可能になります。

 **注:** プロジェクトの資産は、同じプロジェクトまたは参照プロジェクトの他の資産にのみアクセスできます。

プロジェクトの追加

プロジェクトを Silk Test Workbench データベースに追加するには、管理者としてログインする必要があります。プロジェクトを作成する場合、新しいプロジェクトのアクセス権には、デフォルトのアクセス権が設定されます。

1. **ツール > 管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログボックスが開きます。
2. **プロジェクト** タブをクリックします。

3. **追加** をクリックします。 **プロジェクトの追加** ダイアログ ボックスが開きます。
4. **名前** テキスト ボックスに、新しいプロジェクトの名前を入力します。
5. **OK** をクリックします。

プロジェクトの参照

プロジェクトの参照を利用すると、参照プロジェクトのコードや資産を再利用して、資産を体系化できます。さらに、大規模なチームでは、様々なチームがアプリケーションの異なる領域をテストしている場合などに、プロジェクトを参照することで、複数の場所で資産を共有することができます。

他のプロジェクトからプロジェクトを参照するには：

1. Silk Test Workbench メニューで、**ツール** > **管理** を選択します。 **管理** ダイアログ ボックスが開きます。
2. **プロジェクト** タブを選択します。
3. 他のプロジェクトを参照するプロジェクトを選択します。
現在のユーザーがアクセス権を持たないプロジェクトには、取り消し線が引かれます。
4. **変更** をクリックします。
5. **利用可能** リストから参照するプロジェクトを選択します。
6. 矢印を使って、プロジェクトを **参照中** リストに移動します。

参照プロジェクトのすべての資産が、現在のプロジェクトから利用できるようになります。

プロジェクト参照の表示

プロジェクト参照を表示して、現在のプロジェクトが参照しているプロジェクトや、現在のプロジェクトを参照しているプロジェクトを確認できます。

1. Silk Test Workbench メニューから、**表示** > **プロジェクト参照** を選択します。
2. プロジェクト参照が開きます。
3. **参照** タブをクリックすると、現在のプロジェクトが参照しているプロジェクトが表示されます。
4. **参照元** タブをクリックすると、現在のプロジェクトを参照しているプロジェクトが表示されます。

グローバルプロジェクトの定義

プロジェクトをデータベース内のすべての他のプロジェクトから参照できるようにするには：

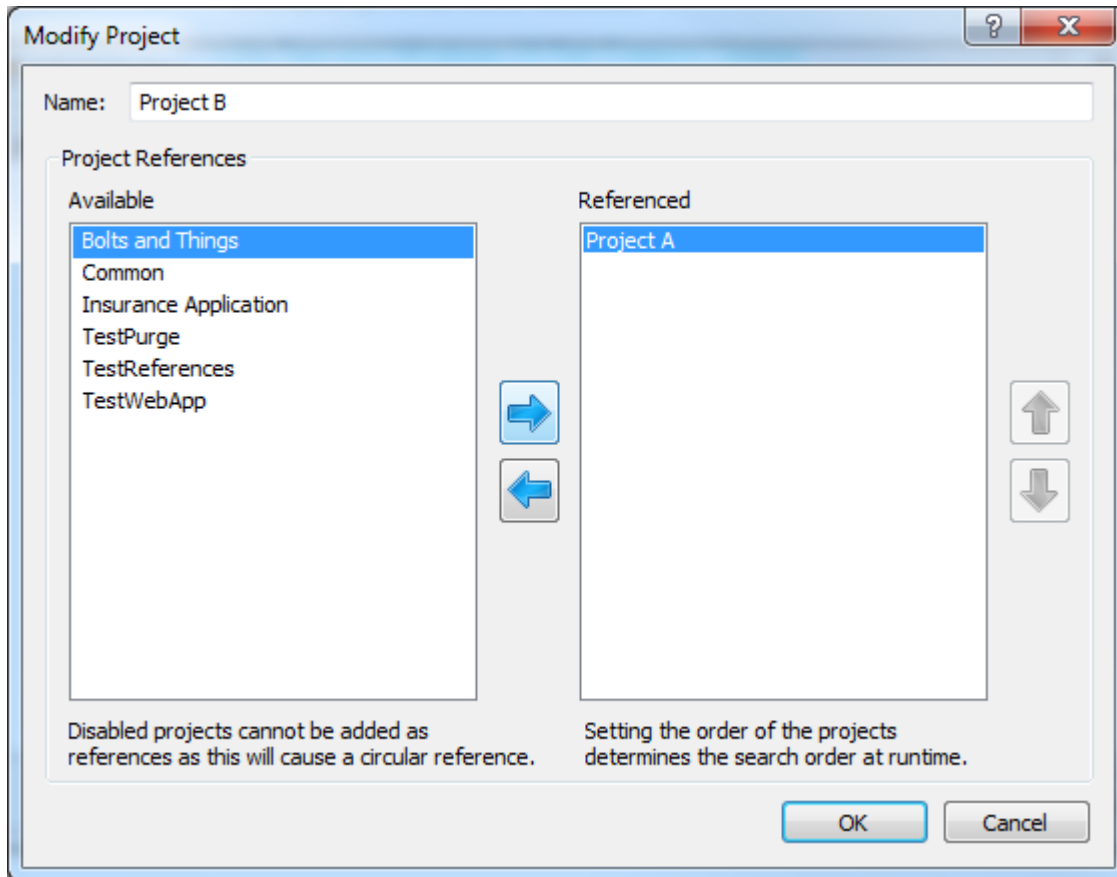
1. Silk Test Workbench メニューで、**ツール** > **管理** を選択します。 **管理** ダイアログ ボックスが開きます。
2. **プロジェクト** タブを選択します。
3. **グローバルプロジェクトの定義** をクリックします。 **グローバルプロジェクトの定義** ダイアログが表示されます。他のプロジェクトを参照していない、かつ他のプロジェクトから参照されていないプロジェクトが **利用可能** リストに表示されます。
4. **利用可能** リストからグローバルに参照するプロジェクトを選択します。
5. 矢印を使って、プロジェクトを **グローバルプロジェクト** リストに移動します。

グローバルプロジェクトのすべての資産が、データベース内のすべての他のプロジェクトから利用できるようになります。

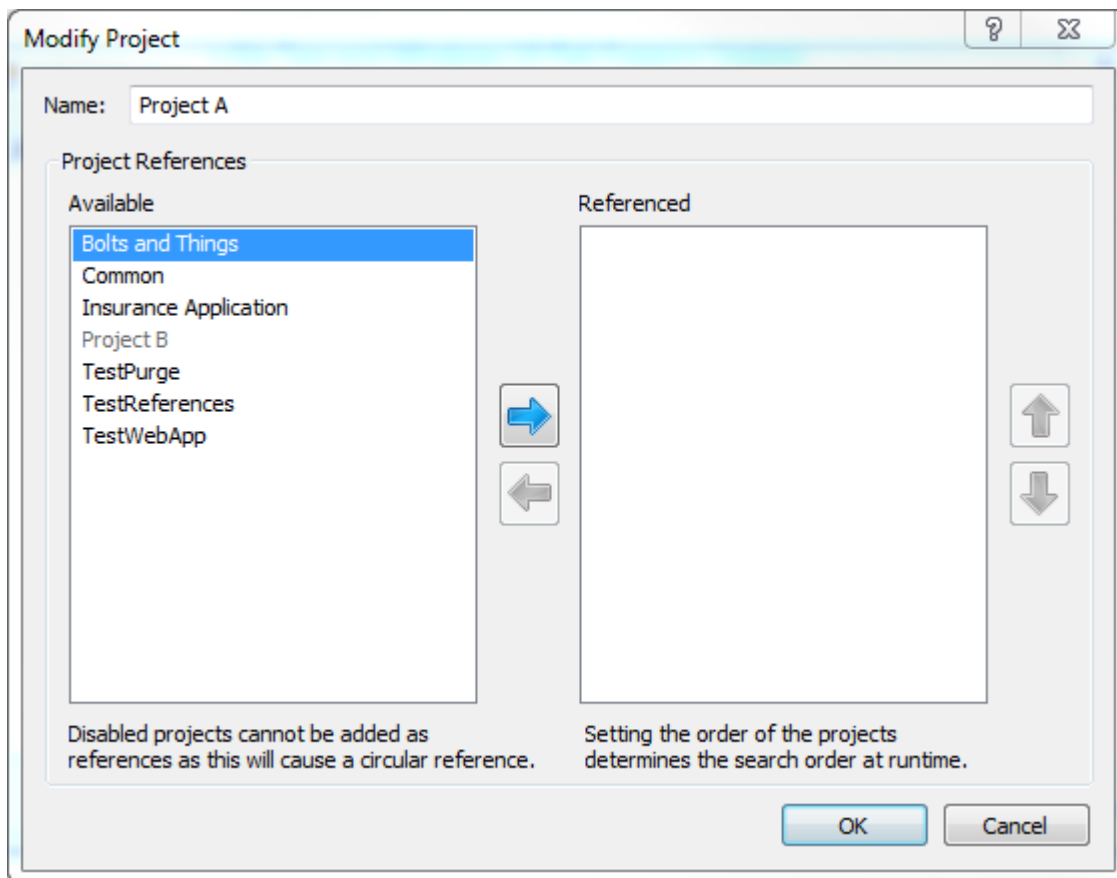
Silk Test Workbench がプロジェクトの循環参照を回避する方法

あるプロジェクトから他のプロジェクトへの新しい参照を追加するとき、Silk Test Workbench はプロジェクト間の参照が確実に循環しないことをチェックします。このトピックでは、Silk Test Workbench での循環参照のチェック方法について説明します。

循環参照の最も簡単な例として、2つのプロジェクト A とプロジェクト B があり、プロジェクト B がプロジェクト A を参照している状態を考えます。



プロジェクト A からプロジェクト B への参照を追加すると、循環参照が形成されます。このため、Silk Test Workbench では、この参照の追加は行えません。プロジェクト A を変更する場合、**プロジェクトの変更** ダイアログ ボックスの **利用可能** リストにプロジェクト B は表示されますが、無効になっています。



Silk Test Workbench では、これをチェーン参照のような複数の参照に対しても適用することで、プロジェクトの参照が循環することを回避します。

プロジェクト名の変更

Silk Test Workbench データベースのプロジェクト名を変更するには、管理者としてログインする必要があります。

1. **ツール > 管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **プロジェクト** タブをクリックします。
3. 名前を変更するプロジェクトを選択し、**名前の変更** をクリックして、**プロジェクト名の変更** ダイアログ ボックスを開きます。
4. **名前** テキスト ボックスに、プロジェクトの新しい名前を入力します。
5. **OK** をクリックします。
6. プロジェクトが他のプロジェクトから参照されている場合は、その処理方法を選択します。
 - プロジェクト名を変更して、参照を更新する場合は、**このプロジェクトの名前を変更して参照を更新する** をクリックします。名前を変更したプロジェクトを参照するプロジェクトへのスクリプト作成者アクセス許可があれば、そのプロジェクト内のすべての資産から、名前を変更したプロジェクトへの参照を探し出して、プロジェクトの新しい名前に参照を更新します。
 - 参照を更新せずにプロジェクト名を変更する場合は、**このプロジェクトの名前のみ変更する** をクリックします。



注: 参照しているプロジェクトの資産は、正しく再生できなくなる可能性があります。

- プロジェクト名の変更をキャンセルする場合は、**キャンセル** をクリックします。
7. **OK** をクリックして **管理** ダイアログを閉じます。すべての参照しているプロジェクトの資産が更新されます。**管理** ダイアログを閉じずに複数のプロジェクトの名前を変更すると、参照しているプロジェクトの資産は、一度だけ更新されることとなります。

プロジェクトの複製

プロジェクトのベースラインを作成する場合など、プロジェクトを複製して、プロジェクトの完全なコピーを作成することができます。Silk Test Workbench データベースのプロジェクトを複製するには、管理者としてログインする必要があります。

1. **ツール** > **管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **プロジェクト** タブをクリックします。
3. 複製するプロジェクトを選択し、**複製** をクリックします。**プロジェクトの複製** ダイアログ ボックスが表示されます。
4. 複製するプロジェクトの名前を **新しい名前** フィールドに指定します。
5. 省略可能：**すべての資産のバージョンをコピーする** チェック ボックスをオンにします。
デフォルトでは、このチェック ボックスはオフになっているため、元のプロジェクトの各資産のアクティブなバージョンだけがコピーされます。
6. プロジェクトが他のプロジェクトを参照している場合、各参照プロジェクトごとに **アクション** 列から適用するアクションを選択します。
 - 複製するプロジェクトから参照プロジェクトへの参照をそのまま保持する場合は、**参照の保持** を選択します。これは、デフォルトの設定です。
 - 参照プロジェクトも複製し、複製するプロジェクトからその複製した参照プロジェクトを参照する場合は、**アクション** 列で **プロジェクトの複製** を選択します。複製したプロジェクトの名前は、**新しい名前** 列で変更できます。
 - 子の参照プロジェクトを複製する場合は、親のプロジェクトも複製する必要があります。たとえば、プロジェクト B を参照するプロジェクト A を複製する際に、プロジェクト B がプロジェクト C を参照している場合は、プロジェクト C を複製するにはプロジェクト B も複製する必要があります。
7. **OK** をクリックします。



注: 参照プロジェクトや複製するプロジェクト名にエラーがある間は、このボタンは有効になりません。



注: 元のプロジェクトにある結果はコピーされません。

プロジェクトの削除

Silk Test Workbench データベースからプロジェクトを削除するには、管理者としてログインする必要があります。

1. **ツール** > **管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **プロジェクト** タブをクリックします。
3. 削除するプロジェクトを選択します。
4. **削除** をクリックします。プロジェクトの削除を確認するメッセージ ボックスが表示されます。
5. **OK** をクリックして、プロジェクトを完全に削除します。


ユーザーの管理

管理者は、Silk Test Workbench のリソースや情報にアクセスすることを許可されたユーザーのリストを管理する必要があります。

データベースに接続中のユーザーの表示

Silk Test Workbench データベースに接続中の接続ユーザーを表示するには：

Silk Test Workbench メニューから、**ツール > 接続ユーザーの表示** を選択します。**接続ユーザー** ウィンドウが開き、すべてのユーザーがリスト表示されます。表示される列は、ユーザー名、ユーザーのフルネーム、データベースに接続したマシン名、データベースにログインした日時になります。このリストは、5 分ごとに更新されます。

 **注:** 同じユーザーが複数のマシンからデータベースにログインした場合や、同じマシン上で複数の Silk Test Workbench を実行している場合など、同じユーザーがリストに複数表示される場合があります。

ユーザーの追加

Silk Test Workbench データベースにユーザーを追加するには、管理者としてログインする必要があります。ユーザー プロファイルを編集すると、プロジェクトに対するユーザーのデフォルト権限をいつでも上書きできます。

1. **ツール > 管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. **追加** をクリックします。**ユーザーの追加** ダイアログ ボックスが開きます。

4. ユーザーの認証情報を Silk Test Workbench が保存できる Windows 認証をこのユーザーに対して使用したい場合は、**ドメイン** フィールドにドメインを入力します。Silk Test Workbench 認証を使用したい場合は、**ドメイン** フィールドを空のままにします。

5. **ユーザー ID** テキスト ボックスに、ユーザーのログオン ID を入力します。

6. **パスワード** テキスト ボックスに、ユーザーのパスワードを入力します。

7. **パスワードの確認入力** テキスト ボックスに、同じパスワードを入力します。

8. **フルネーム** テキスト ボックスに、ユーザーの名前を入力します。

9. **種類** リストを使用して、ユーザーが管理者であるかどうかを指定します。

10 **デフォルトの権限** リストを使用して、ユーザーに対して新しい Silk Test Workbench プロジェクトに対するデフォルトのアクセス レベルを割り当てます。

以下の種類の権限から選択します。

- **アクセス不可**

ユーザーは、プロジェクト情報の表示または変更、あるいは記録および再生の実行ができません。

- **読み取り専用**

ユーザーは、プロジェクト情報を表示できますが、資産を変更したり、記録および再生を実行したりできません。

- **実行者**

ユーザーは、プロジェクト情報を表示して結果を変更できますが、他の資産は変更できません。

- **スクリプト作成者**

ユーザーは、プロジェクト情報の表示と資産の変更を実行できます。

- **フルアクセス**

ユーザーは、すべての資産に対する完全な読み取り/書き込みアクセス権を持ちます。

110K をクリックします。

ユーザーの削除

Silk Test Workbench データベースから他のユーザーを削除するには、管理者としてログインする必要があります。

1. **ツール > 管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. 削除するユーザーを選択します。



注: 管理権を持つ他のユーザーが存在する場合だけ、デフォルトのユーザー *Admin* を削除することができます。

4. **削除** をクリックします。選択したユーザーの削除を確認するメッセージ ボックスが表示されます。

5. **OK** をクリックして、ユーザーを完全に削除します。

ユーザー プロファイルの編集

1. **ツール > 管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. 変更するユーザー プロファイルを選択して、**変更** をクリックします。**ユーザーの変更** ダイアログ ボックスが開きます。

4. 適切なテキスト ボックスに新しい情報を入力します。

5. **OK** をクリックして、変更内容を保存します。

プロジェクトに対するユーザーのアクセス権の変更

1. **ツール > 管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. アクセス権限を変更するユーザーを選択します。

管理 ダイアログ ボックスの右側のパネルには、ユーザーがアクセスできるプロジェクト、および各プロジェクトに対するユーザーのアクセス レベルがリストされます。

4. プロジェクトの **ユーザー アクセス** リストを使用して、ユーザーに割り当てる新しいアクセス権限レベルを選択します。

5. **OK** をクリックします。

データベースからのユーザーのログアウト

管理者として資産のメンテナンス タスクを実行するために、資産メンテナンス モードをオンにすると、管理者以外のユーザーをデータベースからログアウトさせることができます。

1. 資産メンテナンス モードをオンにするには、Silk Test Workbench メニューから **ツール > 資産メンテナ** **ンス モード** を選択します。資産メンテナンス モードをオンにすると、次のようになります。
 - Silk Test Workbench ウィンドウの下部にあるステータス バーに、資産メンテナンス モードがアクティブであることが表示されます。
 - データベース メンテナンス モードを開始すると、管理者以外のすべてのユーザーがログアウトされます。管理者以外のユーザーが Silk Test Workbench ダイアログを開いており、ログアウトできない場合は、Silk Test Workbench にログインしたままになりますが、データベース中の資産を変更できなくなります。
 - すべての開いている資産は保存され、閉じられます。
 - 実行中の記録や再生は、可能な限り速やかに終了されます。
 - 資産メンテナンス モードがアクティブな間は、管理者以外のユーザーは Silk Test Workbench データベースにログインできません。
2. データベースが資産メンテナンス モード中に、管理者が Silk Test Workbench からログアウトしようとする、ログアウトする前に資産メンテナンス モードをオフにするかどうかを確認されます。**はい** をクリックすると、資産メンテナンス モードはオフになります。

グループの管理

Windows ドメイン アカウントのグループを同じ権限でデータベースにログインできるようにします。

グループが共有データベースを持つ場合、単にそのグループを追加して、すべてのグループ メンバが同じ権限でデータベースにログインすることができます。新しいメンバがグループに追加されると、データベースへのアクセス権が自動的に与えられ、メンバがグループから削除されると、データベースへのアクセス権がなくなります。

ローカルマシンのすべてのユーザーがデータベースにアクセスできるようにするには、**BUILTIN¥ALL Users** をグループ リストに追加します。新しい Access データベースには、デフォルトでこのグループがグループ リストが追加され、新しいユーザーが簡単に開始できるようになっています。

グループの追加

同じ資格情報でデータベースにアクセスするために、Silk Test Workbench データベースを共有するグループを有効化するには、グループ プロファイルを追加します。

1. **ツール > 管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **グループ** タブをクリックします。
3. **追加** をクリックします。**グループの追加** ダイアログ ボックスが開きます。
4. グループの認証情報を Silk Test Workbench が保存できる Windows 認証をこのグループに対して使用したい場合は、**ドメイン** フィールドにドメインを入力します。Silk Test Workbench 認証を使用したい場合は、**ドメイン** フィールドを空のままにします。
5. **種類** リストを使用して、グループが管理者グループであるかどうかを指定します。
6. **デフォルトの権限** リストを使用して、グループに対して新しい Silk Test Workbench プロジェクトに対するデフォルトのアクセス レベルを割り当てます。
以下の種類の権限から選択します。
 - **アクセス不可**
ユーザーは、プロジェクト情報の表示または変更、あるいは記録および再生の実行ができません。
 - **読み取り専用**
ユーザーは、プロジェクト情報を表示できますが、資産を変更したり、記録および再生を実行したりできません。

- **実行者**

ユーザーは、プロジェクト情報を表示して結果を変更できますが、他の資産は変更できません。

- **スクリプト作成者**

ユーザーは、プロジェクト情報の表示と資産の変更を実行できます。

- **フルアクセス**

ユーザーは、すべての資産に対する完全な読み取り／書き込みアクセス権を持ちます。

7. **OK** をクリックします。

グループの削除

Silk Test Workbench データベースからグループを削除するには、管理者としてログインする必要があります。

1. **ツール > 管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **グループ** タブをクリックします。

3. 削除するグループを選択します。

4. **削除** をクリックします。選択したグループの削除を確認するメッセージ ボックスが表示されます。

5. **OK** をクリックして、グループを完全に削除します。

グループの変更

1. **ツール > 管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **グループ** タブをクリックします。

3. 変更したいグループを選択し、**変更** をクリックします。**グループの変更** ダイアログ ボックスが開きます。

4. 適切なテキスト ボックスに新しい情報を入力します。

5. ユーザーの種類と権限を適切なリスト ボックスから選択します。

6. **OK** をクリックして、変更内容を保存します。

UI トピック

Silk Test Workbench のデスクトップは、テストにかかわるすべての操作の開始点になります。メニューバーを除き、デスクトップに表示されるウィンドウやペインをはじめとする各種機能の表示と外観を制御することができます。

開始画面

開始画面 は、Silk Test Workbench の機能を使い始める拠点であり、アプリケーションのテスト ソリューションをすばやく作成し始めることができます。

アプリケーションのテストをすばやく、最低限の手順で開始できるようにするため、**開始画面** には次の領域が含まれています。

領域 説明

開始 ビジュアル テスト、.NET スクリプト、キーワード駆動テストをワンクリックで記録したり、開いたりできます。

資産 最近使ったビジュアル テスト、.NET スクリプト、キーワード駆動テストをワンクリックで開くことができます。最近使った資産のリストは、リアルタイムで更新されるので、変更や更新により、現在の資産がリストの先頭に移動します。資産の右側にあるアイコンをクリックすることによって、資産を再生したり、資産の最後の実行結果を表示したりすることができます。



注: **開始画面** 設定 (**ツール > オプション > 開始画面**) では、最近使った資産のリストに表示する項目の数を制御できます。

新機能 Silk Test Workbench の現在のバージョンに含まれている新しい機能や機能強化を一覧します。Silk Test Workbench の更新が利用可能な場合も、ここに表示されます。



注: Silk Test Workbench の起動時に開始画面を表示しないように選択している場合、**ヘルプ > 製品更新の確認** をクリックして利用可能な更新を確認できます。

入門ガイド 生産性向上のため、広く使われるタスクや有用なヘルプ トピックへの直接リンクを提供します。

リソース サポート技術情報や Silk Test コミュニティなどの Silk Test Workbench に関連した情報へのリンクを提供します。

お知らせとコミュニティ Silk Test コミュニティの最新のフォーラム記事とブログ投稿へのリンクを提供します。

フラグ ビジュアル テストのテスト ステップ、およびフラグが付けられてフォローアップ対象として割り当てられた結果が表示されます。**フラグ** ペインを使用すると、テストスイートやテストプロジェクトについての情報を交換でき、データベースのテストプロジェクトのほかのテスト担当者やユーザーと共同作業が行えます。

デフォルトでは、フラグは作成日および表示される資産で分類されます。フラグの情報をすばやく表示するには、マウスのポインタをフラグの上に移動します。フラグの説明、更新データ、プロジェクト、フラグが割り当てられているユーザーの ID を示すツール ヒントが表示されます。**開始画面のフラグ** オプション (**ツール > オプション > 開始画面 > フラグ**) を変更して、フラグの表示方法を変更できます。

領域 説明

フラグに関連付けられているビジュアル テストまたは結果を開くには、**フラグ** ペインでフラグをダブルクリックします。



注: **フラグ** ペインは、少なくとも 1 つのフラグがログインしているユーザーに割り当てられているか、少なくとも 1 つのフラグがフィルタで表示されるように設定されている場合にのみ、**開始画面** に表示されます。

開始画面 の右上隅にあるウィンドウの **閉じる** をクリックすると、**開始画面** を閉じることができます。



ヒント: ツールバーの **開始画面** をクリックすることで、メイン画面から **開始画面** にいつでもアクセスできます。また、いつでも Ctrl+Alt+S を押すか、または **表示** > **開始画面** を選択して、開始画面を表示できます。



注: ユーザー インターフェイスは、Windows 標準の大きいフォントや小さいフォントを使用します。カスタムのフォント サイズを使用すると、ユーザー インターフェイスのテキスト表示が不正確になる可能性があります。

Silk Test Workbench のメニュー

このセクションでは Silk Test Workbench メニューについて説明します。

メニュー コマンドは、現在使用しているウィンドウ、資産、またはテスト モードの種類に応じて、有効または無効にできます。

[ファイル] メニュー

ファイル メニューには以下のコマンドがあります。

コマンド (ショートカット)	説明
新規作成 (Ctrl+N)	新規作成 ウィンドウを表示し、任意の種類の資産を作成することができます。
開く (Ctrl+O)	アセットブラウザ を開き、資産を開くことができます。前回作業した資産の種類が強調表示されます。
閉じる	現在フォーカスが設定されているデスクトップのウィンドウを閉じます。
新しいバージョンとして保存 (Ctrl+S)	<ul style="list-style-type: none">資産の現在の状態を資産の新しいバージョンとして保存し、増分の数値識別子をそのバージョンに割り当てます。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
現在のバージョンとして保存 (Ctrl+S)	資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「 資産保存時のデフォルト動作を設定する 」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
名前を付けて保存	名前を付けて保存 ダイアログ ボックスを開いて、現在の資産を別の名前で保存します。
すべて保存 (Ctrl+Shift+L)	<ul style="list-style-type: none">開いているすべての資産の現在の状態を資産の新しいバージョンとして保存し、各資産のバージョン識別子をインクリメントします。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合

コマンド (ショートカット)	説明
	<p>に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。</p> <ul style="list-style-type: none"> 開いているすべての資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
複製	アセット ブラウザ で資産を選択してからこのメニュー項目を選択し、新しい名前を使って資産を複製します。
名前の変更	アセット ブラウザ で資産を選択してからこのメニュー項目を選択し、選択した資産の名前を変更します。
削除 (Del)	選択した資産を削除します。
すべてのバージョンを表示	アセット ブラウザ で資産を選択してからこのメニュー項目を選択し、資産のすべてのバージョンを表示します。
要約情報 (Alt+Enter)	ビジュアル テストまたはスクリプトから 要約情報 ダイアログ ボックスを表示します。
ページ設定	ブラウザ コントロールの ページ設定 を表示します。このダイアログ ボックスでの設定は、現在選択されている資産に適用され、[印刷プレビュー] ダイアログに表示されません。
印刷プレビュー	印刷プレビュー ダイアログ ボックスを表示して、現在の資産を印刷する前にそのレイアウトを確認できます。
印刷 (Ctrl+P)	[印刷] ダイアログ ボックスを表示して、選択した資産を印刷できます。
資産のエクスポート	資産のデータベース間でのエクスポートは、 資産エクスポート ウィザードを使用して行います。詳細については、「 資産のエクスポート 」を参照してください。
資産のインポート	資産のデータベース間でのインポートは、 資産インポート ウィザードを使用して行います。詳細については、「 資産のインポート 」を参照してください。
最近使った資産	最近アクセスしたテスト資産のサブメニューを表示します。
終了	Silk Test Workbench を終了します。

[編集] メニュー

編集 メニューには以下のコマンドがあります。

コマンド	説明
元に戻す (Ctrl+Z)	テスト スクリプトで直前に加えた編集内容を元に戻します。ビジュアル テストでは利用できません。
やり直し (Ctrl+Y)	テスト スクリプトの直前の 元に戻す 操作を無効にします。ビジュアル テストでは利用できません。
切り取り (Ctrl+X)	選択した項目を切り取って、クリップボードに貼り付けます。
コピー (Ctrl+C)	選択した項目をコピーします。
貼り付け (Ctrl+V)	選択した項目を貼り付けます。
すべて選択 (Ctrl+A)	現在のビジュアル テストのすべてのテスト ステップ、または現在のテスト スクリプトのコード行を選択します。

コマンド	説明
検索と置換	スクリプト内の特定のコードを検索し、必要に応じて置換します。
指定行へ移動 (Ctrl+G)	スクリプト内の特定の行へ移動します。
ブックマーク	ブックマークのオンとオフの切り替え、前のブックマークまたは次のブックマークへの移動、またはスクリプト内のブックマークのクリアを実行します。
行コメント (Ctrl+K)	スクリプトで現在選択されている行 (複数) をコメントアウトして無効化します。コード ブロック全体をコメントアウトするために、このショートカットを使用できます。また、 Ctrl+F11 を使用することもできます。
行コメント解除 (Ctrl+Shift+K)	スクリプトで現在選択されている行 (複数) のコメントを解除して有効化します。
フラグ	現在選択されているフラグを編集または削除します。
画面の更新	選択したテスト ステップについてテスト アプリケーションから画面情報を更新します。テスト アプリケーションの画面が実行されており、更新が必要なステップが選択されている必要があります。
条件	条件 ダイアログ ボックスを表示して、検証全体の合格基準とする、再生での検証の合格の割合を設定できます。
フィルタの管理	結果に表示するデータをフィルタ処理するための フィルタの管理 ダイアログ ボックスを表示します。フィルタは、現在表示されているビューにかかわらず、 要約 ビューと 詳細 ビューの両方に適用されます。

[操作] メニュー

操作 メニューには以下のコマンドがあります。

コマンド	説明
記録	すべてのユーザー操作を、アクティブなビジュアル テストまたはテスト スクリプトの現在の位置に記録します。
一時停止	現在再生中のビジュアル テストまたはテスト スクリプトを一時停止して、実行が中断された地点を強調表示します。
再生 (F5)	アクティブなビジュアル テストまたはスクリプトを再生します。
停止	現在アクティブなビジュアル テストまたはスクリプトの再生を中止します。
基本状態の実行	テスト対象アプリケーションを基本状態に戻します。基本状態とは、テストの実行を始める前にアプリケーションがあるべき既知の安定した状態を言います。詳細については、「 基本状態 」を参照してください。
相違点の表示	選択した 2 つの資産の違いを比較します。
画面の更新	選択したテスト ステップに対して、テスト アプリケーションから画面情報を更新します。テスト アプリケーションの画面が実行されており、更新が必要なステップが選択されている必要があります。

[表示] メニュー

表示 メニューには以下のコマンドがあります。

コマンド	説明
移動	<p>以下に挙げる 1 つまたは複数の項目に移動するためのコマンドを示すサブメニューを表示します。</p> <p>[詳細] ページ : 結果の 合格 タブまたは 失敗 タブで選択したステップから、結果の 詳細 タブの同じステップに移動します。</p> <p>ビジュアルテストのウィンドウ : 結果ウィンドウからビジュアルテストウィンドウに移動します。</p> <p>結果ウィンドウ : ビジュアルテストウィンドウから結果ウィンドウに移動します。</p>
ロジック ツールボックス	<p>テスト ロジック デザイナ へのリンクを表示する [テスト ロジック ツールボックス] を表示します。</p>
新しいウィンドウで開く	<p>選択したイメージを 画面プレビュー に表示するか、または選択した結果を新しいウィンドウに表示します。</p>
テスト ステップ	<p>ビジュアル ナビゲータ の テスト ステップ ウィンドウまたは 結果 ウィンドウで、特定のステップの表示を切り替えます。</p> <p>ステップのみ : テスト アプリケーションに対してテスト操作を実行するステップを表示します。</p> <p>画面のみ : テストに関連付けられた画面を表示するステップを表示します。</p> <p>ステップと画面 : 操作を実行するステップと画面を表示するステップの両方を表示します。</p> <p>ステップの説明 : 表示されるテスト ステップに入力されている説明を表示します。</p>
列	<p>結果 ウィンドウの 詳細 タブに結果情報を表示します。</p> <p>基本表示 : 標準的な テスト ステップ ペインの情報、および 結果 と 結果の詳細 の追加の列を表示します。</p> <p>詳細表示 : 各ステップの詳細情報を表示します。</p>
開始画面 (Ctrl+Alt+S)	<p>開始画面 を表示します。</p>
アセット ブラウザ (Ctrl+Alt+A)	<p>アセット ブラウザ を開きます。</p>
エラー リスト	<p>テスト ステップ ペインで、エラーを赤色で表示します。</p>
出力	<p>出力 ウィンドウを表示します。</p>
インクルード	<p>インクルード ウィンドウを表示します。</p>
キーワード	<p>作業中のプロジェクトのすべてのキーワード、および統合した Silk Central インスタンスで定義されたキーワードを一覧する キーワード エディタ を表示します。</p>
更新	<p>ウィンドウを更新します。</p>
ズーム	<p>画面プレビュー の現在のイメージのサイズを拡大/縮小します。アプリケーション ウィンドウがアクティブな場合に有効になります。</p>
ツール バー	<p>ツールバーの表示/非表示を切り替えます。</p>
ステータス バー	<p>現在のセッション情報を表示するステータスバーを表示します。</p>
レイアウトの復元	<p>ビジュアル ナビゲータ のレイアウトを元の表示設定に戻します。</p>

[挿入] メニュー

挿入 メニューには以下のコマンドがあります。

コマンド	説明
テスト ロジック	現在のビジュアル テストに対してロジックを作成します。詳細については、「 ツールバー 」または「 テスト ロジック 」を参照してください。
コントロール	次のひとつからビジュアル テストにコントロールを挿入します。 <ul style="list-style-type: none">• テスト対象アプリケーション。• 画面プレビュー。• オブジェクトの識別 ダイアログ。
ファイル	外部ファイルをスクリプトに挿入します。スクリプトがアクティブな場合に有効になります。
ビジュアル テスト	ビジュアル テストの参照 ダイアログ ボックスを開いて、ビジュアル テストをアクティブなビジュアル テストに追加します。ビジュアル テストがアクティブな場合に有効になります。
.NET スクリプト	.NET スクリプトの参照 ダイアログ ボックスを開いて、.NET スクリプトをアクティブなビジュアル テストに追加します。ビジュアル テストがアクティブな場合に有効になります。
外部プログラム	アクティブなビジュアル テストの再生中に、選択したポイントでアプリケーションまたはプログラムを開始します。
式	式デザイナー を表示して、アクティブなビジュアル テストに式を挿入できます。
検証資産	ビジュアル テストの現在のステップに既存の検証資産を挿入します。現時点では、イメージ検証のみが検証資産としてサポートされています。詳細については、「 イメージ検証 」を参照してください。
結果コメント	ビジュアル テストの現在のステップに結果のコメントを挿入します。詳細については、「 ビジュアル テストへのコメントの挿入 」を参照してください。
ラベル	空白の編集可能なステップを、プロパティ値を指定せずにビジュアル テストに挿入します。
エラー処理	ビジュアル テストの現在のステップにエラー処理を挿入します。
メッセージ ボックス	メッセージ ボックスのテスト ステップを挿入します。
コメント	ビジュアル テストの現在のステップにコメントを挿入します。
再生設定	再生設定とその値を現在のテストに設定するか、または現在のテストから取得します。
コントロールのプロパティ	空のステップを挿入し、そのステップを使用してテスト アプリケーションのコントロールからプロパティをキャプチャして、変数に設定します。ステップの プロパティ ペインで、 ロケータ フィールドの 識別 ボタンを使用してコントロールをキャプチャし、コントロールの特定のプロパティを選択します。
プログラム フロー	テスト フローを制御するステップを挿入するか、またはテストの信頼性を高めます。
同期とタイミング	オブジェクトを待機するテスト ステップ、指定した時間だけビジュアル テストの再生を遅らせるテスト ステップ、または再生のタイマーを作成するテスト ステップを挿入します。

コマンド	説明
SAP eCATT	<ul style="list-style-type: none"> 引数の取得 引数の設定 <p>ビジュアルテストと SAP eCATT 資産の間で変数を受け渡しできるようにします。SAP が Silk Test Workbench と同じコンピュータにインストールされている場合に有効になります。詳細については、「SAP eCATT の操作」を参照してください。</p>
アクティブ データ	<p>次の操作があります。</p> <p>新規作成 新しいアクティブ データ資産を作成して、ビジュアルテストに関連付けます。詳細については、「アクティブ データ資産のビジュアルテストへの関連付け」を参照してください。</p> <p>既存の関連付け 既存のアクティブ データ資産をビジュアルテストに関連付けます。詳細については、「アクティブ データ資産のビジュアルテストへの関連付け」を参照してください。</p> <p>直ちに保存 アクティブ データ ファイルを直ちに保存するステップをビジュアルテストに挿入します。詳細については、「ビジュアルテスト内で更新されたアクティブ データの即時保存」を参照してください。</p> <p>保存のキャンセル アクティブ データ ファイルを保存する前の要求をキャンセルします。</p>
変数	<p>ビジュアルテストに対して、ローカル変数の追加や、グローバルパラメータの設定または取得を行います。詳細については、「ビジュアルテストにおける変数」を参照してください。</p>
フラグ	<p>選択したテスト ステップにフラグまたはメモを設定します。詳細については、「フラグ」を参照してください。</p>
画面の更新	<p>選択したテスト ステップに対して、テスト アプリケーションから画面情報を更新します。テスト アプリケーションの画面が実行されており、更新が必要なステップが選択されている必要があります。</p>

[デバッグ] メニュー

テストのデバッグの詳細については、「[テストのデバッグ](#)」を参照してください。**デバッグ** メニューには以下のコマンドがあります。

コマンド (ショートカット)	使用可能な資産	説明
コンパイル (F7)	<ul style="list-style-type: none"> スクリプト 	スクリプトの構文がコンパイルのすべての条件を満たしているかをチェックし、コードをコンパイルし、エラーが発生した場合は表示します。
ステップイン (F8)	<ul style="list-style-type: none"> スクリプト ビジュアルテスト 	ビジュアルステップの場合、テスト ステップを一度に 1 つずつ再生します。スクリプトの場合、ステートメントを一度に 1 つずつ再生します。
ステップ オーバー (Shift+F8)	<ul style="list-style-type: none"> スクリプト ビジュアルテスト 	埋め込まれているビジュアルテストまたはスクリプト全体を再生し、元のビジュアルテストまたはスクリプト内の次のステップで停止してデバッグモードになります。VB .NET スクリプト

コマンド (ショートカット)	使用可能な資産	説明
ステップ アウト (Ctrl+Shift+F8)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	<p>プドで押下された場合は、関数呼び出しのステップ オーバーも行います。</p> <p>別のビジュアル テストから再生しているビジュアル テスト内の残りのステップをすべて実行したあと、元のビジュアル テスト内の次のステップで再生が一時停止します。スクリプトの場合は、プロシージャ内の残りすべてのコードを 1 つのステートメントのように実行し、そのプロシージャを最初に呼び出したプロシージャ内の次のステートメントに移ります。</p>
カーソルまで実行 (Ctrl+F8)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	現在選択されているテスト ステップまでビジュアル テストを再生するか、または現在選択されているステートメントまでスクリプトを再生します。
カーソルから実行	<ul style="list-style-type: none"> ビジュアル テスト 	現在選択されているテスト ステップからビジュアル テストを再生します。
ブレイクポイントの設定/クリア (F9)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	選択したステップまたはコード行にブレイクポイントを設定するか、または選択したステップまたはコード行に設定されているブレイクポイントをクリアします。
すべてのブレイクポイントのクリア (Ctrl+Shift+F9)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	すべての開いているビジュアル テストと VB .NET スクリプトのすべてのブレイクポイントをクリアします。
実行ポイント/次のステートメントの設定 (Ctrl+F9)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	次のステートメントまたはポイントまで再生を実行します。デバッグ モードの場合に有効になります。
ローカル変数	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	現在のビジュアル テストまたはスクリプト、および埋め込まれているビジュアル テストまたはスクリプトについて、すべての変数とその値をリストするウィンドウを表示します。

[ツール] メニュー

ツール メニューには以下のコマンドがあります。

コマンド	説明
データベースの変更	ログオン ダイアログ ボックスを表示して、別のデータベースにログオンします。
パスワードの変更	パスワードの変更 ダイアログ ボックスを表示して、現在のユーザーのログオン パスワードを変更します。
オブジェクトの識別 (Ctrl+Shift+I)	選択したオブジェクトのロケータを記録できる オブジェクトの識別 ダイアログ ボックスを開きます。
管理	ユーザーを管理し、プロジェクトにアクセスします。
オプション	構成設定を表示し、それらの設定をカスタマイズできます。
キーワード ライブラリのアップロード	キーワード ライブラリを Silk Central にアップロードします。Silk Test Workbench が Silk Central と統合され

コマンド	説明
カスタム コントロールの管理	<p>ている場合にのみ有効になります。詳細については、「Silk Central へのキーワード ライブラリのアップロード」を参照してください。</p> <p>カスタム コントロールの管理 ダイアログ ボックスを表示します。</p>
アプリケーション構成の編集	<p>アプリケーション構成の編集 ダイアログ ボックスを表示して、特定のプロジェクトでキーワード駆動テストによってテストされるアプリケーションを指定したり、アプリケーションの起動方法を設定することができます。詳細については、「アプリケーション構成の編集」を参照してください。</p>
リモート ロケーションの編集	<p>リモート ロケーション ダイアログ ボックスを表示して、ブラウザやデバイスにアクセスするリモート ロケーションを指定することができます。詳細については、「リモート ロケーションの編集」を参照してください。</p>
ツールのカスタマイズ	<p>ツール メニューにメニュー項目を追加します。</p>
SAP eCATT	<p>SAP eCATT で作業するためのコマンドにアクセスできます。Silk Test Workbench が SAP eCATT と統合されている場合にのみ有効になります。詳細については、「SAP eCATT の操作」を参照してください。</p>

[ツール] メニューのカスタマイズ

Silk Test Workbench では、**ツール** メニューにメニュー項目を追加できます。これらのメニュー項目により、頻繁に使用するプログラムを簡単に開くことができるようになります。メニュー項目の追加は、頻繁に使用するプログラムやドキュメントにすばやくアクセスする方法を提供するという点で、ショートカットの使用に似ています。たとえば、メモ帳用のカスタム メニュー項目を作成すると、Silk Test Workbench での作業を中断せずに、**ツール** メニューからメモ帳を起動できるようになります。

新しいコマンドに関する情報を入力したあとは **追加** をクリックしないでください。クリックすると、別の空白エントリが (**新しいツール**) として追加され、**ツール** メニューの項目として表示されてしまいます。

- ツール** > **ツールのカスタマイズ** をクリックします。 **カスタマイズ** ダイアログ ボックスが開きます。
- 新しい項目を作成するには、**追加** をクリックします。 **メニュー内容** リストに空白エントリが (**新しいツール**) として表示されます。 (**新しいツール**) エントリは、**メニュー テキスト** テキスト ボックスにも表示されます。
- メニュー テキスト** テキスト ボックスで、メニュー項目の名前を入力します。
この名前は、**ツール** メニューに表示されるようになります。たとえば、このメニュー項目によってメモ帳を起動する場合は、「メモ帳」と入力します。
- コマンド** テキスト ボックスで、**参照** をクリックして、メニュー項目を実行するファイルに移動して選択します。
ファイルを選択するには、**プログラムの選択** ダイアログ ボックスでそれを強調表示して、**開く** をクリックします。
ファイルのパスと名前がわかっている場合は、**コマンド** テキスト ボックスにそれを入力することもできます。
- 引数** テキスト ボックスに、**コマンド** テキスト ボックスのファイルを実行するために必要なパラメータを入力します。
コマンドが必要ない場合は、このテキスト ボックスを空のままにします。
- 初期ディレクトリ** テキスト ボックスに、元の項目を含むパスとディレクトリ、またはメニュー項目によって実行されるファイルに必要なすべての関連ファイルを入力します。

プログラムによっては、複数の場所のファイルを使用する必要があります。メニュー項目によって実行されるファイルがこれらのファイルを見つけられるように、これらのファイルが置かれているフォルダを指定する必要があります。

7. **OK** をクリックします。**カスタマイズ** ダイアログ ボックスが閉じ、項目が **ツール** メニューに追加されます。

[ウィンドウ] メニュー

ウィンドウ メニューには以下のコマンドがあります。

コマンド	説明
上下に並べて表示	開いているすべてのウィンドウを上下に並べて表示します。
左右に並べて表示	開いているすべてのウィンドウを左右に並べて表示します。
重ねて表示	開いているすべてのウィンドウを重ねて表示します。
アイコンの整列	最小化したスクリプト、ビジュアル テストなどをウィンドウの最下部に整列します。
開いているファイルのリスト	開いているビジュアル テスト、結果、スクリプト、またはウィンドウのサブメニューを表示します (アセット ブラウザ や 開始画面 など)。

[ヘルプ] メニュー

ヘルプ メニューには以下のコマンドがあります。

コマンド	説明
目次	ヘルプの目次を表示します。
検索	ヘルプの 検索 タブを開きます。
索引	ヘルプの 索引 タブを開きます。
チュートリアル	Silk Test Workbench のチュートリアルを開きます。
製品更新の確認	Silk Test Workbench の新しいバージョンが利用可能かどうかを確認します。
Silk Test Workbench のバージョン情報	Silk Test Workbench の製品情報を表示します。

ツール バー

メニューの **表示** > **ツール バー** を開き、ツール バーの左側にあるチェック ボックスのオン/オフを切り替えて、ツール バーを表示/非表示します。Silk Test Workbench では、以下のツール バーを利用できます。

ファイル 資産管理を行うための以下のアクションが提供されます。

ボタン	説明
新規作成 (Ctrl+N)	新規作成 ウィンドウを表示し、任意の種類の資産を作成することができます。
開く (Ctrl+O)	アセット ブラウザ を開き、資産を開くことができます。前回作業した資産の種類が強調表示されます。
新しいバージョンとして保存 (Ctrl+S)	<ul style="list-style-type: none">資産の現在の状態を資産の新しいバージョンとして保存し、増分の数値識別子をそのバージョンに割り当てます。ショートカット キーの組み

ボタン	説明
	<p>合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。</p> <ul style="list-style-type: none"> 資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
すべて保存 (Ctrl+Shift+L)	<ul style="list-style-type: none"> 開いているすべての資産の現在の状態を資産の新しいバージョンとして保存し、各資産のバージョン識別子をインクリメントします。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。 開いているすべての資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
印刷 (Ctrl+P)	[印刷] ダイアログ ボックスを表示して、選択した資産を印刷できます。
印刷プレビュー	印刷プレビュー ダイアログ ボックスを表示して、現在の資産を印刷する前にそのレイアウトを確認できます。
索引	ヘルプの 索引 タブを開きます。

編集 編集を行うための以下のアクションが提供されます。

ボタン	説明
切り取り (Ctrl+X)	選択した項目を切り取って、クリップボードに貼り付けます。
コピー (Ctrl+C)	選択した項目をコピーします。
貼り付け (Ctrl+V)	選択した項目を貼り付けます。
元に戻す (Ctrl+Z)	テスト スクリプトで直前に加えた編集内容を元に戻します。ビジュアル テストでは利用できません。
やり直し (Ctrl+Y)	テスト スクリプトの直前の 元に戻す 操作を無効にします。ビジュアル テストでは利用できません。

ボタン	説明
行コメント (Ctrl+K)	スクリプトで現在選択されている行 (複数) をコメントアウトして無効化します。コードブロック全体をコメントアウトするために、このショートカットを使用できます。また、 Ctrl+F11 を使用することもできます。
行コメント解除 (Ctrl+Shift+K)	スクリプトで現在選択されている行 (複数) のコメントを解除して有効化します。

アクション

記録操作を行うための以下のアクションが提供されます。

ボタン	説明
記録	すべてのユーザー操作を、アクティブなビジュアルテストまたはテストスクリプトの現在の位置に記録します。
再生 (F5)	アクティブなビジュアルテストまたはスクリプトを再生します。
一時停止	現在再生中のビジュアルテストまたはテストスクリプトを一時停止して、実行が中断された地点を強調表示します。
停止	現在アクティブなビジュアルテストまたはスクリプトの再生を中止します。
オブジェクトの識別 (Ctrl+Shift+I)	選択したオブジェクトのロケータを記録できる オブジェクトの識別 ダイアログボックスを開きます。

表示 テストロジック

アセットブラウザ や **開始画面** を表示します。

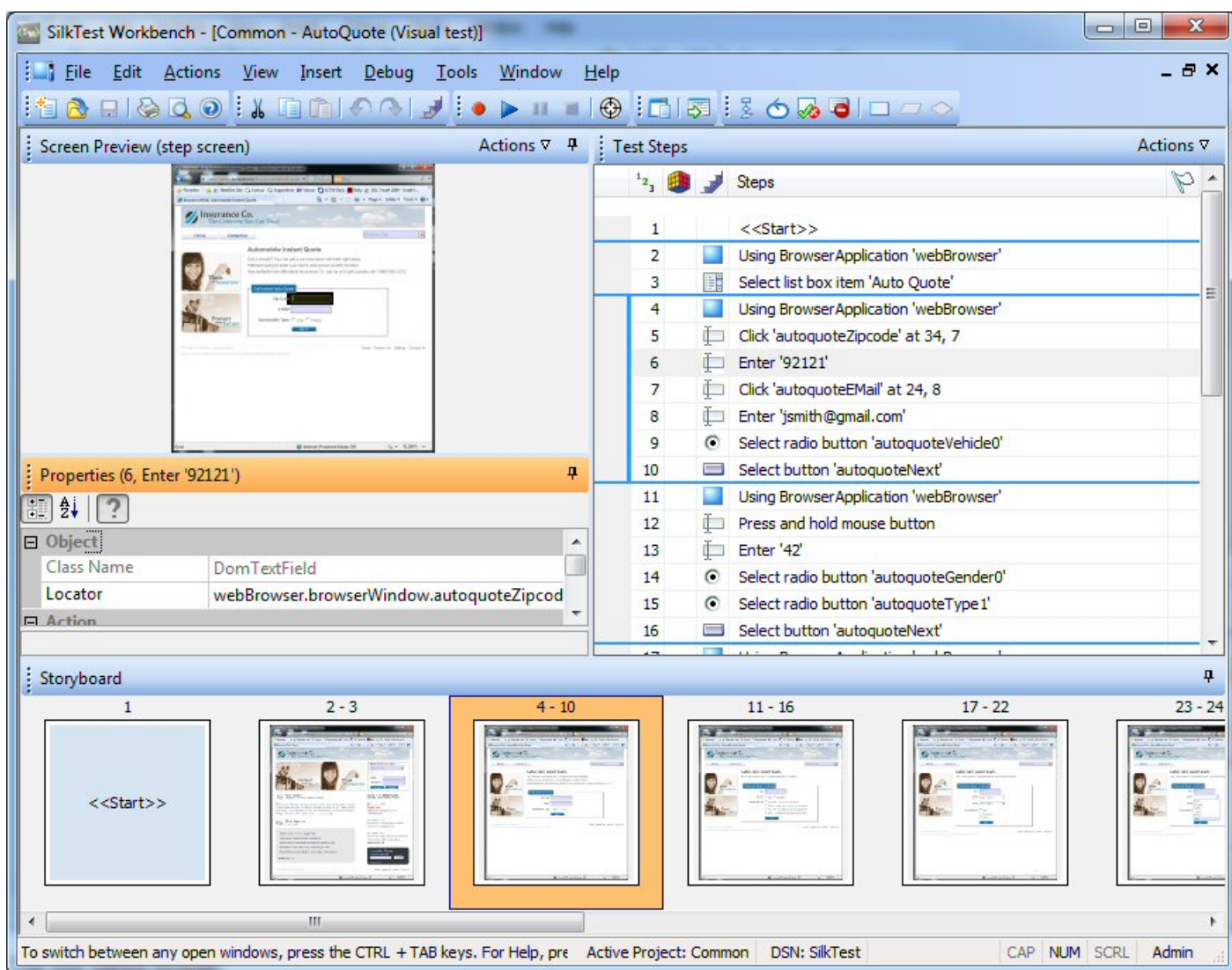
ビジュアルテストにロジックを追加するための以下のアクションが提供されます。

ボタン	説明
判断タイプのロジックの作成	判断ロジックでは、ユーザー定義の条件を評価して、その条件の戻り値に応じて一連のテストステップを実行します。
繰り返しタイプのロジックの作成	繰り返しロジックは、指定した回数、またはユーザー定義の条件が true になるまで、一連のテストステップを繰り返します。
検証タイプのロジックの作成 (Ctrl+Alt)	検証ロジックは、ユーザー定義の条件を評価して、合格/失敗メッセージをビジュアルテストまたはスクリプトの結果に送信します。
再生エラー処理ロジックの作成	エラー処理ロジックを設定すると、再生エラーをデバッグする必要がなくなります。エラー処理を設定するには、ビジュアルテストでエラーの検出ステップとその処理方法を設定します。
If タイプのロジック項目の挿入	作業中のビジュアルテストに条件ロジックを追加します。選択中のテストステップが、If と End If テストステップで囲まれます。テストステップは、条件に合致した場合にのみ実行されます。

ボタン	説明
Else If タイプのロジック項目の挿入	ビジュアルテストの If または Else If テストステップと End If テストステップ間でのみ使用できます。先行する If または Else If テストステップの条件が合致せず、この Else If テストステップの条件に合致した場合に、Else If と End If の間のテストステップが実行されます。
Else タイプのロジック項目の挿入	ビジュアルテストの If または Else If テストステップと End If テストステップ間でのみ使用できます。先行する If または Else If テストステップの条件が合致しない場合に、Else と End If の間のテストステップが実行されます。

ビジュアルナビゲータ

ビジュアルナビゲータは、ビジュアルテストの要素をグラフィカルに表すものであり、ポイントアンドクリックインターフェイスを通して各要素を操作できます。**ビジュアルナビゲータ**では、ビジュアルテストに関する情報が4つのペインに表示されます。4つのペイン全体でビジュアルテストの各ステップを総合的に把握できます。



4つのペインは以下のとおりです。

テストステップ ビジュアルテストの各ステップが技術的ではない明確な言語で示されます。

画面プレビュー ビジュアルテストの再生中に実行されるステップに応じてテスト対象アプリケーションのスナップショットが表示されます。


プロパティ ビジュアルテストのステップのプロパティが表示されます。

ストーリーボード サムネイルイメージを使用してビジュアルテストのフローが表示されます。サムネイルは、ビジュアルテストのステップの論理グループを表します。

 **注:**

Silk Test Workbench では、スナップショットは以下のような状況のときに取得されます。

- 記録中のすべての自動テストステップの前。

 **注:** SAP アプリケーションの場合、スナップショットはすべての自動テストステップの前ではなく、画面が変わったときに発生します。

- ビジュアルテストで Using ステップを実行した結果のスナップショット。

- 再生エラーが発生した場合。

画面プレビュー、**ストーリーボード**、および**プロパティ**ペインは**テストステップ**ペインと同期され、**テストステップ**ペインで選択したステップに関する情報が表示されます。このように、**テストステップ**ペインでステップを選択し、他のペインでそのステップに関する情報を確認することで、ステップのすべての側面を簡単に把握できます。

ビジュアルナビゲータでは、ビジュアルテストを表示するだけでなく、**画面プレビュー**ペインと**プロパティ**ペインを使用して既存のビジュアルテストを拡張または更新することもできます。たとえば、ビジュアルテストを記録したあと、**プロパティ**ペインで、記録したプロパティのリテラル値を変数に置き換えることができます。また、テスト対象アプリケーションに変更が発生したときにビジュアルテストをすばやく更新する場合は、**画面プレビュー**の**画面の更新**機能を使用して、以前にキャプチャした画面を更新できます。

ビジュアルナビゲータでは、ビジュアルテストに使用すると同じペインを使用して、ビジュアルテストの再生結果も表示します。結果の場合、これらのペインには機能が付加され、**結果**ウィンドウ内に表示されます。ウィンドウにはツールバー オプションと複数のタブが表示されます。各タブには結果のさまざまな内容が表示されます。結果に固有の追加機能として、たとえば、**テストステップ**ペインに各ステップのステータス（合格または失敗）を表示する機能があります。さらに、**画面プレビュー**では、記録中にキャプチャされた画面と再生中にキャプチャされた画面の違いを比較し、テストアプリケーションにアクセスすることなく、既存のビジュアルテストを更新できます。

[ビジュアルナビゲータ]: [テストステップ]ペイン

テストステップペインは、ビジュアルテスト、およびビジュアルテストとスクリプトの結果を表示したり、管理したりするための中央の領域です。**テストステップ**ペインには、ビジュアルテストを構成する、記録または挿入された操作が1行ずつ表示されます。各操作はステップと呼ばれる個別の行で表されます。ステップは、テストアプリケーションをテストするために行われる操作を、技術的でない明確な言語で表します。各ステップは、テストアプリケーションに対して実行される操作またはテストを管理するために使用される操作に対応しています。

テストアプリケーションと対話するステップは、**画面プレビュー**で、対応するアプリケーション画面要素のイメージによって表されます。また、各ステップは**ストーリーボード**のサムネイルに対応するようにグループ化されています。

各ステップには、ステップの操作に関する詳細を決定するプロパティが含まれています。ステップのプロパティは**プロパティ**ペインに表示されます。


ビジュアルテストの**テストステップ**ペインから、記録したビジュアルテストを確認し、以下の操作を追加して、テストの信頼性、柔軟性を向上させたり、検証、機能、およびフローを改良したりすることができます。





- テストロジックの追加
- ビジュアルテストの再生方法の指定
- 他のビジュアルテストおよびスクリプトの実行
- ビジュアルテストのデバッグ
- エラー処理の提供

テストステップペインをカスタマイズして、ビジュアルテスト内の必要なステップのみを表示できます。

結果ウィンドウに表示される**テストステップ**ペインには、テスト結果に固有の追加情報が表示されます。**テストステップ**ペインが表示されない**要約**タブを除き、この情報は**基本表示**と**詳細表示**に表示されます。

テストステップペインの標準ビューには以下の列があります。

列	説明
 (番号) *	ステップが記録され、再生される順番を表します。デフォルトでは、ステップはこの順番で表示されます。

列	説明
	<p>ステップにデバッグのためのブレークポイントが追加されている場合は、この列の隣の列にブレークポイントが表示されます。</p> <p>デフォルトのビュー（ステップと画面）から画面のみのビューまたはステップのみのビューにビューを変更した場合、番号付けの方法では、記録の順序で間隔が示されるようにステップまたは画面は順番にスキップされます。</p>
 (ロジック) *	ステップにロジックが含まれている場合、ステップのロジックのタイプを表すアイコンを表示します。
 (ステップ タイプ) *	ステップで実行される操作のタイプを表すアイコンを表示します。
ステップ	ステップで実行される操作について説明します。
 (フラグ) *	<p>割り当てられたフラグ のアイコンを表示します。</p> <p>割り当てられたフラグ のアイコンは、関連付けられたステップが割り当て先のユーザーの テスト ステップ ペインおよびオプションで 開始画面 ペインに表示されるように、フラグが付けられていることを示します。さらに、ポインタをアイコンの上に移動すると、フラグの説明、更新日、および割り当て日を示すツールヒントを表示できます。</p>
 (ステップの説明) *	<p>ユーザー定義のステップの説明を表示します。この列はデフォルトのビューには表示されません。テスト ステップ ペインのタイトルバーから 操作 をクリックし、次に 表示 > ステップの説明 をクリックすると表示できます。</p> <p>ステップの説明を作成するには、ステップを選択します。プロパティ ペインで、ステップの説明 プロパティを更新します。ステップの説明 アイコンがこの列に表示され、ステップに説明があることが示されます。ポインタをアイコンの上に移動して説明を含むツールヒントを表示するか、ステップを選択してプロパティ ペインの説明を読みます。</p>

* この列はサイズ変更できません。

[結果] ウィンドウの [基本表示]

列	説明
結果	ビジュアル テストのステップまたはスクリプトのコード行について、実行ステータスを表示します。
結果の詳細	ビジュアル テストのステップまたはスクリプトのコード行について、実行ステータスの詳細を表示します。














[結果] ウィンドウの [詳細表示]

列	説明
行番号	スクリプトのコード行の行番号またはビジュアル テストのステップ番号を示します。[コマンドの詳細] と組み合わせて使用すると、特定のコード行またはステップ番号を見つけることができます。
コマンドの詳細	実際のコード行テキストを表示します。[行番号] と組み合わせて使用すると、テストスクリプトの特定のコード行またはビジュアル テストのステップ番号を見つけることができます。
日付/時刻	テスト ステップまたはコード行が実行されたときのタイムスタンプを表示します。
ミリ秒	テスト ステップまたはコード行の実行時間を表示します。

列	説明
名前	ビジュアル テストまたはスクリプトの名前を表示します。
ユーザー名	テスト実行者のログオン識別子を表示します。
マシン名	テストを実行するために使用したデバイスの固有の識別子を表示します。

テスト ステップ アイコン

アイコンは、ビジュアル テストのテスト ステップで一般的に実行される操作をグラフィカルに表したものです。グラフィックは、**テスト ステップ** ウィンドウの **テスト ステップ** 列に表示されます。以下の表は、一般的に使用される操作とそのアイコンを示しています。

ステップ タイプ	アイコン	説明
画面キャプチャ		再生中にこのステップに対して画面キャプチャが行われることを示します。
オブジェクト クラス	各種	各 UI オブジェクト クラスには固有のテスト ステップ アイコンがあります。たとえば、Window クラスのアイコンは  であり、TextField クラスは  です。
オブジェクトの待機		待機 プロパティが挿入されたことを示します。
判断ロジック		If ステートメント、Else If ステートメント、または End If ステートメントを実行します。
繰り返しテスト ロジック		一連のステップを指定した回数だけ実行します。
検証ロジック		ステップの 条件 プロパティに規定された条件を検証します。
エラー処理ロジック		エラー処理を実行します。
ビジュアル テスト		指定したビジュアル テストを再生します。
外部プログラム		指定した外部プログラムをステップの属性に従って起動します。
式		式を含みます。
結果コメント		結果コメントを含みます。
ラベル		ラベルを含みます。

テスト ステップの種類

ビジュアル テストの各ステップでは、テスト アプリケーションのコントロールに対する操作の実行、複合条件の評価、または単にビジュアル テストへのブックマークの挿入など、特定の操作が実行されます。**テスト ステップ** ペインの **ステップ** 列では、対応するステップが実行する操作の説明が記述されています。

一般的なテスト ステップは、テスト アプリケーションのコントロールに対して操作を実行しないテスト ステップです。ビジュアル テストで利用できる一般的なテスト ステップの種類を以下の表に示します。

ステップ テキストの内容	説明
<<開始>>および<<終了>>	ビジュアル テストの最初のステップと最後のステップです。ビジュアル テストで使用する受け渡し変数のプロパティは、<<開始>> ステップのプロパティに表示されます。
検証	ステップの 条件 プロパティに規定された条件を検証します。
If/Else/Else If/End If	ユーザー定義の条件を評価して条件の戻り値に応じて一連のテスト ステップを実行する判断ロジックを実行します。

ステップ テキストの内容	説明
	If テスト ステップがビジュアルテストに挿入されると、対応する End If テスト ステップも挿入されます。
ビジュアル テストの再生	ステップに関連付けられている再生設定プロパティを使用して、指定されたビジュアルテストを再生します。
.NET スクリプトの再生	ステップに関連付けられている再生設定プロパティを使用して、指定されたスクリプトを再生します。
プログラムの実行	ステップの コマンド プロパティの設定に基づいて、外部プログラムを起動します。
式	式を実行して評価し、その結果を変数に設定します。
結果コメント	ビジュアル テストの再生時に指定されたコメントを作成して、結果に送信します。コメントには、テキスト、式の値、アクティブ データ ファイル、または式の結果を使用でき、テストの 結果の詳細 タブの コマンドの詳細 列に表示されます。
ラベル'ラベル名'	ステップをラベルとして設定します。ステップのテキストは、ステップの ラベル プロパティに設定されるラベルの名前です。 ラベルを使用すると、テスト内の特定の領域やステップをブックマークし、テスト フローや再生を制御できます。
エラー処理	エラー処理を実行します。ステップの 実行する操作 プロパティによって、エラー処理が有効になっているときにエラーが発生した場合のテストの動作が決まります。エラー処理が有効になっているときにエラーが発生した場合、ステップの 資産 プロパティを使用して、別のテストを再生することができます。
メッセージ ボックス	再生中にテスト担当者に対してメッセージ ボックスを表示します。メッセージ ボックスの内容は、ステップに設定されているプロパティ値で決まります。
再生設定の設定	実行環境の再生設定を特定の値に変更します。この値は、テストステップの 再生設定 プロパティで設定されます。値は、そのビジュアル テストの再生に対してのみ、値がリセットされるまで有効です。
再生設定の取得	実行環境の再生設定を取得して、取得した設定を変数に設定できるようにします。
コントロールの "" プロパティの取得	空のステップを挿入し、そのステップを使用してテスト アプリケーションのコントロールからプロパティをキャプチャして、変数に設定します。 プロパティ ペインの 識別 ボタンを使用してコントロールをキャプチャし、コントロールの特定のプロパティを選択します。
ラベルへ移動します	指定されたラベル ステップにジャンプして、再生します。
待機	コントロールの存在または消滅を監視します。監視するコントロールは、リテラル、変数、式、またはアクティブ データとして表すことができます。タイムアウト プロパティで待機時間を指定します。
停止	再生を完全に停止します。通常、停止は「If」ステップなどの条件ロジックで使用されます。
遅延	ステップの 遅延時間 プロパティで指定されている時間だけ再生を一時停止します。ステップの 遅延タイプ プロパティは、時間の単位（秒またはミリ秒）を決定します。
タイマー番号の開始	タイマーを作成して開始します。 タイマーを使用して、ビジュアル テスト全体またはその一部の再生に要する時間を測ります。1 つのビジュアル テストで複数のタイマーを使用できます。
タイマー番号の停止	ステップの タイマー番号 プロパティで指定されたタイマーを停止します。

ステップ テキストの内容	説明
タイマーの再開	タイマーを使用して、ビジュアル テスト全体またはその一部の再生に要する時間を測ります。1 つのビジュアル テストで複数のタイマーを使用できます。
eCATT 引数の取得	ステップの タイマー番号 プロパティで指定された、停止中のタイマーを再開します。
eCATT 引数の設定	SAP eCATT 引数コンテナから引数の値を取得します。この値は、eCATT スクリプトからビジュアル テストに渡されます。 このステップでは、SAP eCATT スクリプトから渡される、事前定義のインポート引数に変数を渡します。 Silk Test Workbench がインストールされているコンピュータに SAP eCATT をインストールすると、eCATT 変数値が取得可能になります。
アクティブ データ ファイル " に値を直ちに保存します	SAP eCATT スクリプトに渡す引数の値を設定します。 このステップでは、SAP eCATT スクリプトに渡すことのできる、事前定義のエクスポート引数に変数を渡します。 Silk Test Workbench がインストールされているコンピュータに SAP eCATT をインストールすると、eCATT 変数値が設定可能になります。
アクティブ データ ファイル " に値を保存しません	ステップの実行時に、指定されたアクティブ データ資産に対して更新された関連データを保存します。
グローバル変数の設定	指定されたアクティブ データ ファイルから更新されたデータが、ステップの実行時点でファイルに保存されないようにします。このステップのあとで、「アクティブ データ ファイルの" に値を直ちに保存します」ステップが同じアクティブ データ ファイルに対して実行されると、このステップのあとで更新されたデータがファイルに保存されます。
グローバル変数の内容の取得	グローバル変数名を指定し、値を初期化することによって、グローバル変数を定義します。
	グローバル変数の値を取得して、ビジュアル テストで使用するローカル変数に設定します。

[ビジュアル ナビゲータ] : [画面プレビュー]

画面プレビュー には、**テスト ステップ** ペインに表示された、コントロールと対話する各ステップについてキャプチャされたテスト アプリケーションのイメージが表示されます。キャプチャされたイメージとして、デスクトップ全体、アプリケーション ウィンドウ、またはアクティブなウィンドウのみを示すことができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

画面プレビュー と **テスト ステップ** ペインは同期されています。コントロールと対話するステップを選択すると、**画面プレビュー** にテスト アプリケーション画面のイメージが表示され、選択したステップで自動処理されるコントロールが強調表示されます。強調表示されたステップがイメージにあるコントロールと対話すると、そのコントロールの周りに四角形が表示されます。また、キャプチャされたイメージ内のコントロールをクリックすると、そのコントロールと対話する連続ステップが存在する場合は、その最初のステップが強調表示されます。

ビジュアル ナビゲータ でビジュアル テストを操作する場合、テスト アプリケーションにアクセスしなくても、**画面プレビュー** のイメージを使用してステップを作成できます。

ビジュアル ナビゲータ の **テスト ステップ** ペインに何も表示されていない場合、**画面プレビュー** には記録の開始方法を示すイメージが透かし表示されます。

キャプチャされたテスト アプリケーション画面の表示

テスト アプリケーションのキャプチャ イメージを **画面プレビュー** で表示するには、表示するキャプチャ イメージを含むステップを **テスト ステップ** ペインで選択します。キャプチャ イメージが表示されたら、イメージのサイズを変更したり、別のウィンドウにイメージを表示したりできます。

1. 別のサイズで表示するキャプチャ イメージに対応するステップを選択します。
2. **画面プレビュー** から、**操作** をクリックして以下のコマンドのいずれかをクリックします。
 - **ズーム** : 既存の **画面プレビュー** ペインでキャプチャ イメージを表示します。割合 (キャプチャされたサイズの 10% から 300%) を選択するか、**ウィンドウに合わせる** をクリックしてイメージを **画面プレビュー** の現在のサイズに合わせます。
 - **新しいウィンドウで開く** : キャプチャされた画面を **画面プレビュー** ペインとは別の、サイズ変更可能なウィンドウに表示します。

[ビジュアル ナビゲータ] : [プロパティ]

ビジュアル ナビゲータ の **プロパティ** ペインには、ビジュアル テストの **テスト ステップ** ペインで選択したステップのプロパティか、ビジュアル テストまたはテスト スクリプトの結果が表示されます。プロパティは名前と値のペアとして表示されます。プロパティの値は、リテラル値、変数、またはアクティブ データ値です。

ビジュアル テストの **プロパティ** ペインを使用して、選択したステップのプロパティに値を割り当てることができます。これにより、Silk Test Workbench によるテスト アプリケーションとの対話、再生の管理、変数の使用、他の機能テストの呼び出しの方法について、正確な情報を調整できます。結果の **プロパティ** ペインは読み取り専用です。

プロパティ ペインには、以下のオプションを含むツールバーがあります。

カテゴリ	プロパティをプロパティ カテゴリ別にグループ化します。
アルファベット	プロパティをアルファベット順に表示します。
説明	プロパティ ペインの下部にヘルプの説明パネルを表示します。

結果の **プロパティ** ペインには追加のツールバー ボタンがあります。

再生前のビジュアル テストのステップ プロパティを表示/非表示	再生前にビジュアル テストに設定されていたステップのプロパティを表示します。
--	--

テスト ステップに関して表示されるプロパティは、ステップのタイプに基づきます。

ビジュアル テストの [プロパティ] ペインの表示

プロパティ ペインには、資産に関連付けられたプロパティと各プロパティの現在の設定がリストされます。**プロパティ** ペインを使用して、資産のプロパティ値を変更できます。複数の資産またはオブジェクトを選択すると、**プロパティ** ペインには、選択したすべての資産またはオブジェクトに共通のプロパティがリストされます。

プロパティの値を変更するには、プロパティ名に隣接するテキスト ボックスでプロパティの値を変更します。値が事前に定義されているプロパティの場合は、該当するリスト ボックスに表示される値の 1 つを選択します。

1. **テスト ステップ** ペインから、表示するプロパティが含まれるステップを選択し、**操作** > **プロパティ** をクリックします (または F4 を押します)。
2. **アルファベット** ツールバー ボタンをクリックしてプロパティをアルファベット順で表示するか、**カテゴリ** ツールバー ボタンをクリックしてプロパティをカテゴリ タイプ別に表示します。

[ビジュアルナビゲータ]: [ストーリーボード]





ストーリーボード ペインでは、テキストまたはイメージを一連のサムネイルで表示することによって、現在のビジュアル テストの全般構造がグラフィカルに表示されます。ストーリーボード ペインは、ビジュアル テストの概要を視覚的に表示し、ビジュアル テストのフローの概要を示します。




サムネイルとは、テスト アプリケーションからキャプチャされた画面か、1 つのステップまたは一連のステップを説明するテキストを表す縮小サイズのイメージのことです。サムネイルは、アプリケーション画面をキャプチャするテスト ステップに関連付けられています。テキストサムネイルは、テスト アプリケーションと直接対話を実行しないテスト ステップを表します。例として、<<開始>> と <<終了>> のステップや、他のビジュアル テストまたはスクリプトが呼び出されるステップがあります。

テスト ステップ ペインに表示されるステップは、論理グループを示す濃い青の実線で区切られています。このようにグループ化されたステップは、**ストーリーボード** ペインの各サムネイルに対応します。

サムネイルのサイズは、サイズ変更可能な **ビジュアルナビゲータ** ウィンドウのサイズに対して相対的になっています。

ストーリーボード ペインには、以下の情報の要素が表示されます。

要素	説明
サムネイル イメージ	<p>ビジュアル テストのフローをグラフィカルに表します。イメージは、対応するテスト ステップを表す、テスト アプリケーションからキャプチャされたビットマップを表します。</p> <p>ビジュアル テストの開始と終了を示すために、各 [ストーリーボード] には、最初のステップとして <<開始>> サムネイル、最後のステップとして <<終了>> サムネイルが表示されます。</p> <p>ビットマップがないステップまたはステップのグループについては、サムネイルに説明テキストが表示されます。サムネイルにテスト アプリケーションのビットマップが含まれていない場合があります。対応する 1 つまたは複数のステップがテスト アプリケーションと直接対話しない（スクリプトを開くステップなど）場合や、テスト アプリケーションがビットマップでキャプチャされていない場合です。Silk Test Workbench でテスト アプリケーションから画面をキャプチャする方法をカスタマイズできます。詳細については、「記録の出力: テスト オプション」を参照してください。</p> <p>[ストーリーボード] のサムネイルをクリックして、それをアクティブなストーリーボード項目にします。アクティブなストーリーボード項目に対応するステップが テスト ステップ ペインで強調表示されます。対応する画面が 画面プレビュー に表示されます。プロパティ ペインのプロパティ情報も、アクティブなサムネイルを反映して変更されます。</p>
ステップ番号	各サムネイルの上に対応するステップ番号が表示されます。たとえば、番号 5 から 8 がサムネイルの上に表示された場合、そのサムネイルは現在のビジュアル テストのステップ 5 から 8 を表します。
アイコン	<p>以下のアイコンは、サムネイルに関連付けられたステップまたはステップのグループについての情報を提供します。</p> <ul style="list-style-type: none"> フラグ付きステップ ユーザーに割り当てられたフラグ付きステップ ロジックまたはエラー処理ステップ 検証ロジック ステップ

要素	説明
	さらに、 結果 ウィンドウの [ストーリーボード] には以下のアイコンが表示されます。
	 合格した検証ロジック ステップ
	 失敗した検証ロジック ステップ
	 注: [ストーリーボード] のサムネイルに同じタイプのステップが複数含まれている場合は、そのサムネイルに下にフラグが 1 つだけ表示されます。

ストーリーボードでのサムネイルの移動

ストーリーボードでサムネイルを移動するには、以下のキーボード操作を使用します。

- ストーリーボードで次のサムネイルに移動するには、右矢印 キーまたは 下矢印 キーを使用します。
- ストーリーボードで前のサムネイルに戻るには、左矢印 キーまたは 上矢印 キーを使用します。
- 最初のステップに戻るには、Page Up キーを使用します。
- 最後のステップに戻るには、Page Down キーを使用します。
- ビジュアル テストの先頭にある <<開始>> サムネイルに移動するには、Home キーまたは Ctrl +Home キーを使用します。
- ビジュアル テストの最後にある <<終了>> サムネイルに移動するには、End キーまたは Ctrl+End キーを使用します。

ビジュアル ナビゲータのレイアウトのカスタマイズ


ビジュアルナビゲータの画面プレビュー ペイン、プロパティ ペイン、およびストーリーボード ペインは、ドッキングしたり、非表示にしたり、サイズを変更したりできます。


1. ビジュアル ナビゲータのそれぞれのペインおよび関連ウィンドウには、右上隅に押しピンが表示されます。押しピンをクリックするとペインが非表示になります。ペインは非表示になり、**ビジュアル ナビゲータ** ウィンドウの境界線にタブとしてドッキングされます。
2. ポインタをタブの上に置くと、ペインが一時的に表示されます。
3. タブをクリックすると、ペインが一時的に再ドッキングされます。
4. ペインを再表示して、押しピンをクリックすると、ペインが永続的に再ドッキングされます。
5. **ビジュアル ナビゲータ** 内でペインのサイズを調整するには、サイズ変更するペインの間にカーソルを置きます。マウスのカーソルが横方向または縦方向の分割カーソルに変わります。分割カーソルをクリックし、ドラッグして 2 つのペインのサイズを調整します。
6. **表示 > レイアウトの復元** をクリックすると、**ビジュアル ナビゲータ** のレイアウトはデフォルトビューに戻ります。

[結果] ウィンドウ

ビジュアル テストまたはスクリプトを再生したあと、再生結果を **結果** ウィンドウに表示することができます。

Summary **Passed** Failed Flags Details (16 steps)

Overall result for run 1 :  **Passed** No verifications were executed

Browser:  Internet Explorer 11 - Windows 7

Reason for abort: Not applicable

Latest run number: 1

Recent runs: 1

Visual test executed: TestResultOS

Visual test description:

Result description:

Scripts (number of times each ran): [Project A.TestResultOS.version 1 \(1\)](#)

Verifications passed: 0 / 0

Verifications failed: 0 / 0

Flags: 0

Start time: 13.03.2018 14:14:39

End time: 13.03.2018 14:14:46

Total time: 7 s

Steps run: 16

Playback setting: System Defaults Edit...

結果 ウィンドウには以下の機能があります。

ビジュアル ナビゲータ テスト再生の全体の様子がすばやくわかります。詳細については、「[結果] ウィンドウのビジュアル ナビゲータ」を参照してください。

[結果] ウィンドウの ツールバー 結果の内容の表示と種類をカスタマイズすることができます。詳細については、「[結果] ウィンドウのツールバー」を参照してください。

[結果] ウィンドウの タブ タブには、結果の内容を特定の種類にまとめられています。詳細については、「[結果] ウィンドウのタブ」を参照してください。

アセットブラウザ

アセットブラウザ を使用してテスト資産を管理します。**アセットブラウザ** では、データベース内の各種種類の資産を一元的に作成、管理、表示できます。任意の時点で Ctrl+Alt+A を押すか、**開始画面** の **タスク** ペインから **アセットブラウザ** をクリックして、**アセットブラウザ** を表示します。

ある種類の既存の資産項目を表示するには、**資産の種類** でその種類をクリックします。関連する資産のリストが右側に表示されます。資産の名前変更、削除、または複製を行うには、資産を右クリックして、適切なコマンドを選択します。資産プロパティを表示するには、資産を選択して **ファイル > 要約情報** を選択します。

新しい資産を作成するには、資産の種類を右クリックして **新規作成** を選択します。新しい資産が、現在アクティブなプロジェクトに追加されます。

テストの開発中に多数の資産が作成される可能性があります。必要な項目を見つけられるように、資産のビューをさまざまな方法で変更できます。

アセットブラウザ のコントロールには以下のものがあります。

アクティブプロジェクト アクティブ プロジェクトを設定します。これは、新しいテスト資産が追加されるデフォルトのプロジェクトです。アクティブ プロジェクトを変更すると、新しいアクティブ プロジェクトの資産が自動的に表示されます。

表示 表示したい資産のプロジェクト名のチェック ボックスをオンまたはオフにして選択します。アクティブ プロジェクトの資産が常に表示されます。

フィルタ 資産の作成者、最終更新者、作成日時、最終更新日時などの設定した条件に基づいて、資産のフィルタ処理を行います。

資産の種類 リストから資産の種類を選択し、適用されたフィルタ処理に基づいて、選択されているプロジェクトで使用可能なテスト資産を表示します。詳細については、「[資産の種類](#)」を参照してください。

資産リスト 適用されたフィルタ処理に基づいて、選択されているプロジェクトの選択された種類の資産を表示します。

項目を開く 開く資産の名前を入力します。

アセットブラウザ ではこれらのコントロールを使用して資産を管理する他、以下の処理を行うことができます。

- 項目のリストでカテゴリ別に資産を並べ替えるには、基準とする列見出しをクリックします。並べ替えの順序を逆にするには、列見出しを再びクリックします。
- 表示する列を選択したり、列の順序を変更したりするには、列見出しを右クリックし、**列のカスタマイズ** を選択します。
- 項目に直接アクセスするには、**項目を開く** テキスト ボックスにその項目の名前を入力します。


 **注:** 多数の資産を管理するために、**アセットブラウザ** のインスタンスを複数開くことができます。開ける **アセットブラウザ** のインスタンス数は、最大 32 です。

[オブジェクトの識別] ダイアログ ボックス


オブジェクトの識別 ダイアログ ボックスを使用して、テスト対象アプリケーション (AUT) のオブジェクト (コンテナなどの表示されないオブジェクトを含む) のロケータを識別できます。テスト中にオブジェクト マップ エントリーを使用しているかどうかによって、ロケータは、オブジェクト マップ項目の名前か、選択したオブジェクトの XPath ロケータのいずれかになります。テスト中にオブジェクト マップ エントリーを使用している場合、オブジェクト マップの名前がオブジェクトの XPath ロケータの別名として使用されます。**オブジェクトの識別** ダイアログ ボックスを使用してオブジェクトのロケータを識別した場合、そのロケータはビジュアル テストと VB .NET スクリプトで使用できます。また、XPath ロケータまたはオブジェクト マップ エントリーが有効で、期待するオブジェクトに対応していることを検証するために、**オブジェクトの識別** ダイアログ ボックスを使用することもできます。

オブジェクトの識別 ダイアログ ボックスは、次のように複数の方法で起動できます。


- ツールバーで、**オブジェクトの識別** をクリックします。

 **注:** テスト アプリケーションを指定していない場合、**アプリケーションの選択** ダイアログ ボックスが開き、オブジェクトを識別するアプリケーションを指定できます。

- メニューで、**ツール > オブジェクトの識別** をクリックします。


 **注:** テスト アプリケーションを指定していない場合、**アプリケーションの選択** ダイアログ ボックスが開き、オブジェクトを識別するアプリケーションを指定できます。

- **開始画面** で、**入門ガイド** タブを選択して **オブジェクトの識別** をクリックします。

 **注:** **アプリケーションの選択** ダイアログ ボックスが開き、オブジェクトを識別するアプリケーションを指定できます。

- 既存のビジュアル テストで、変更するオブジェクトを参照するテスト ステップを選択します。**プロパティ** ペインを開き、**ロケータ** フィールドをクリックして、**コントロールの識別 > オブジェクトの識別** ダイアログ を選択します。

オブジェクトの識別 ダイアログ ボックスには、以下のコントロールがあります。

識別の開始	識別の開始 をクリックして、テスト アプリケーションでオブジェクト情報の記録を開始します。ロケータを記録する前に、テスト対象アプリケーションを特定の状態にするには、 識別の停止 をクリックします。
選択モード	識別するオブジェクトを選択する方法として、選択モードを定義します。 クリック モードの場合は、識別するオブジェクトをクリックします。 ホット キー モードの場合は、識別するオブジェクトにマウス カーソルを移動して、 キー入力 リスト ボックスで指定したキー入力の組み合わせを使用します。大抵は、メニューやコンボ ボックスなど、ユーザーがクリックしたときにのみ表示されるオブジェクトをキャプチャする場合に ホット キー モードを選択します。 ホット キー モードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャできます。
選択済みロケータ	キャプチャしたロケータが表示されます。 識別の開始 をクリックすると、 選択済みロケータ フィールドには、現在マウス カーソルが指している AUT 上のオブジェクトのロケータが表示されます。  注: ブラウザーでテストしている場合、 選択済みロケータ フィールドに実際にキャプチャするとロケータのみが表示される。
テスト	このボタンをクリックすると、テスト対象アプリケーションで選択したオブジェクトがハイライトされます。このボタンを使用すると、正しいオブジェクトが選択されることを検証します。
ロケータの詳細の非表示 / ロケータの詳細の表示	このリンクをクリックして、 ロケータの詳細 テーブルを表示または非表示にします。
[ロケータの詳細] テーブル	ロケータの詳細 テーブルには、テストしているアプリケーションのオブジェクトとその関連するクラス タイプ、ロケータ値、および属性が一覧表示され、アプリケーションの階層を確認できます。このテーブルを使用して、選択やオブジェクトに割り当てる属性を絞り込むことができます。たとえば、自動生成されたロケータは与えられたテストで使用するのに最適なロケータではない場合があります。 ロケータの詳細 テーブルを使用すると、ロケータで利用するのに、より関係のあるプロパティを見つけることができます。
オブジェクト マップの表示	オブジェクト マップ項目名を ロケータ 列に表示するには、このチェック ボックスをオンにします。オブジェクト マップ項目名は、コントロールまたはウィンドウに対して、コントロールやウィンドウのロケータではなく論理名 (エイリアス) を関連付けます。デフォルトでは、オブジェクト マップ項目名が表示されます。ロケータを使用するには、このチェック ボックスをオフにします。
完全ロケータの表示	完全ロケータ名を表示するには、このチェック ボックスをオンにします。オブジェクトに関連付けられた属性のみを表示するには、このチェック ボックスをオフにします。
プロパティの表示	ロケータの詳細 テーブルで選択したオブジェクトの属性および属性値を表示するには、このチェック ボックスをオンにします。このテーブルの属性を選択して、ロケータの識別に使用できます。[プロパティ] サブツリーを非表示にし、ロケータの詳細のみを表示するには、このチェック ボックスをオフにします。
貼り付け	このボタンをクリックして、 選択済みロケータ を適切なビジュアル テスト、スクリプト、またはオブジェクト マップにコピーします。

オブジェクトの識別 ダイアログ ボックスを使用してオブジェクトをキャプチャすると、以下のことを実行できます。

- **ロケータの詳細** ツリーを展開し、コンテキスト メニューを使用してアプリケーションの階層を表示します。

- コンテキストメニューを使用してサブツリーにあるオブジェクトのロケータを取得します。
- サブツリーのリンクをクリックして、追加のロケータの属性を表示します。
- サブツリーにあるオブジェクトを選択して、**選択済みロケータ** テキストボックスに表示されている既存のオブジェクトを置き換えます。
- ビジュアルテストの**プロパティ** ペインで、ロケータを既存のロケータにコピーします。
- ロケータを .NET スクリプトに直接コピーします。

スクリプトで、ロケータと共に使用するコードを指定する必要があります。ロケータだけがコピーされます。

- ロケータをオブジェクト マップのオブジェクト マップ項目にコピーします。

[コード] ウィンドウ

コード ウィンドウでは、スクリプトの記録、設計、および変更を行います。

コード ウィンドウにアクセスするには、既存のスクリプトを開くか、スクリプトを作成します。

Main ページ

コード ウィンドウを開くと、**Main** ページに基本テンプレートが入力されます。スクリプトの記録後、**Main** ページには記録したすべてのステップが表示されます。このページにコードを手動で入力することもできます。**Main** ページは、**ファイル** カテゴリの**プロパティ** ペインにも表示されます。

[プロパティ] ペイン

プロパティ ペインを使用して、アプリケーション構成、ファイル、パラメータ、およびスクリプト参照を追加、表示、または変更します。**プロパティ** ペインには以下のカテゴリがあります。

- **アプリケーション構成** : スクリプトのアプリケーション構成を表示するには、このカテゴリを展開します。
- **ファイル** : コード ウィンドウの **Main** ページと、追加したその他のファイルを表示するには、このカテゴリを展開します。**Main** ページには、記録したすべてのステップが表示されます。
- **パラメータ** : スクリプトの入力パラメータと出力パラメータを表示するには、このカテゴリを展開します。必要に応じて、入力パラメータと出力パラメータを追加、編集、および削除できます。
- **参照** : スクリプト ライブラリの参照を表示するには、このカテゴリを展開します。必要に応じて、参照を追加および削除できます。

スクリプトの [プロパティ] ペインの表示

プロパティ ペインを使用して、アプリケーション構成、ファイル、およびスクリプト参照を追加、表示、または変更します。コード ウィンドウの**プロパティ** ペインは、ドッキング、非表示、またはサイズ変更が可能です。

1. スクリプトを開きます。コード ウィンドウが開きます。
2. **プロパティ** ペインには、右上隅に押しピンが表示されます。押しピンをクリックするとペインが非表示になります。ペインは非表示になり、コード ウィンドウの境界線にタブとしてドッキングされます。
3. ポインタをタブの上に置くと、ペインが一時的に表示されます。
4. タブをクリックすると、ペインが一時的に再ドッキングされます。
5. ペインを再表示して、押しピンをクリックすると、ペインが永続的に再ドッキングされます。
6. ペインのサイズを調整するには、サイズ変更するペインの間にカーソルを置きます。マウスのカーソルが横方向または縦方向の分割カーソルに変わります。分割カーソルをクリックし、ドラッグして2つのペインのサイズを調整します。

出カウィンドウ

出カ ウィンドウには、スクリプトの再生中に見つかったエラーに関する情報、スクリプトで実行された検索の結果、スクリプトによって書き込まれたテキストなどが表示されます。

出カ ウィンドウには、参照されているオブジェクト マップ エントリが実行中の資産の範囲内でない場合、再生に関する警告メッセージも表示されます。



注: Pushbutton.Select などの NTF メソッド (呼び出し) の結果のみがログに記録されます。したがって、with ステートメントなどの VB.NET コマンドを実行した場合、VB.NET 呼び出しも _desktop.Window() 呼び出しもログに記録されません。

既知の問題

このセクションでは、Silk Test Workbench の既知の問題とその解決策を示します。

全般的な問題

オブジェクト マップを開くのに時間がかかる

.NET 4 を使用している場合、大規模なオブジェクト マップ資産があると読み込みに時間がかかります。.NET 4.5 をインストールすると、この問題を解決できます。

リモート デスクトップまたはリモート デスクトップ接続 (RDC) が最小化されると、Silk Test が機能しない

リモート デスクトップ プロトコル (RDP) 経由でデスクトップに接続している場合、マウスとキーボードを使ってデスクトップに接続することにより、デスクトップの所有権を獲得します。デスクトップの所有権を解放せずにデスクトップを最小化すると、マウス クリックまたはキーストロークの再生がすべて未定義になります。

回避策として、VNC ベースのリモート操作ツールを使用することができます。この場合、クライアントが最小化された場合でも再生を継続できます。

Check Point ファイアウォールがインストールされている場合に Open Agent が起動しない

システムに Check Point ファイアウォールまたは Check Point ZoneAlarm ファイアウォールをインストールしている場合は、ファイアウォールが Agent と infoservice 間の通信を中断するために Open Agent を起動できません。

Open Agent を起動するには、システムから Check Point ファイアウォールをアンインストールする必要があります。

domDoubleClick メソッドの modifiers パラメータが無視される

domDoubleClick メソッドのオーバーロードで修飾キーを指定できません。パラメータが指定されている場合でも、修飾キーはダブルクリックされません。修飾キーを指定できる domDoubleClick メソッドのオーバーロードは、非推奨です。修飾キーを指定するには、doubleClick メソッド (modifiers パラメータを取るオーバーロード メソッドをサポートするクライアントを使用している場合)、または PressKeys および ReleaseKeys メソッドを使用します。

Silk Test が Metro スタイル アプリのテストをサポートしない

Silk Test は、Microsoft Windows 8、Microsoft Windows 8.1、Microsoft Windows 10 上の Metro スタイル アプリのテストをサポートしません。Metro スタイル アプリは、Windows 8 スタイル、Modern UI スタイル、Windows ストア スタイル、ユニバーサル Windows プラットフォーム (UWP) アプリとも呼ばれます。

Microsoft Windows 8 の組み込みスペル チェックがテストの再生に干渉する

Microsoft Windows 8 の組み込みスペル チェックは、Internet Explorer 10 などのアプリケーションで有効にできます。

記録中に単語のスペルを間違え、この単語の入力を再生すると、スペルチェッカーはこの単語をマークするか、間違われやすい単語の場合は自動的に修正します。これは、実際のユーザーに対して行われる動作と同じです。スペル チェック機能が含まれていないオペレーティング システムでテストを作成した場合、

Microsoft Windows 8 でテストを再生すると、予期せぬ結果が生じることがあります。スペルチェックを無効にするには、次の手順を実行します。

1. **Windows キー + C** を押します。
2. チャーム バーで **設定** をクリックします。
3. **PC 設定の変更** を選択します。
4. **全般** を選択すると、スペルチェック セクションが表示されます (使用中の言語によってスペルチェック機能の有無は異なります)。



注: これはシステム全体の設定で、Internet Explorer 固有の設定ではありません。

5. **スペル ミスの語句を自動修正する (Autocorrect misspelled words)** をオフに設定します。
6. **スペル ミスの語句を強調表示する (Highlight misspelled words)** をオフに設定します。

.NET アプリケーションを DevPartner Studio (DPS) から起動すると、Silk Test で認識されないことがある

この問題を解決するには、以下のステップを実行します。

1. Silk Test のインストール フォルダに移動します (デフォルトでは C:\Program Files\Silk\SilkTest)。
2. Windows Forms アプリケーションの場合は、ng\agent\plugins
com.borland.fastxd.techdomain.windowsforms.agent_<バージョン番号> へ移動します。
3. Windows Presentation Foundation (WPF) の場合は、ng\agent\plugins
com.microfocus.silktest.techdomain.wpf.agent_<バージョン番号> へ移動します。
4. メモ帳で、plugin.xml ファイルを開いて、<loadparameters> セクションに以下の行を追加します。

```
<param name="frameworkAssembly">mscoree.dll</param>
```
5. コンピュータからログアウトして、再びログインします。DevPartner Studio によって起動されたアプリケーションに対して、Silk Test が期待どおりの動作をします。

イメージ領域に対するクリックの記録時に矩形領域によってハイライトされる位置がずれる

エリア マップのような複雑なイメージの一部に対するクリックを記録する場合、イメージの適切な領域が緑色の矩形領域によってハイライトされません。ただし、再生時にクリックは正しく実行されます。

Silk Test のインストール中に Windows Defender が有効になっていると Open Agent が起動しない

Silk Test のインストール中にシステムの Windows Defender が有効になっていると、インストールが完了した後に Open Agent を起動できなくなる場合があります。Windows Defender は、ホットフィックス セットアップに必要な操作を妨げる場合があります。回避策として、Silk Test のインストール中は Windows Defender を無効にしてください。

IME エディタが特定の場所から開かれたときにポップアップする

IME エディタを開くと、テキスト フィールド内には開くのではなく、現在の画面の左上隅にポップアップして開きます。

この動作は、次の場所から IME エディタを開いたときに発生します。

- Silk Recorder
- キーワード駆動テスト エディター
- キーワード ウィンドウ

検索ボックスに文字を入力すると Silk Test ヘルプがフリーズする

Microsoft Windows 8.1、Microsoft Windows RT 8.1、または Microsoft Windows Server 2012 R2 上のコンパイルされた HTML ヘルプ (.CHM) ファイルで、**キーワード** タブの **検索** ボックスに文字を入力すると、Silk Test ヘルプがフリーズする場合があります。この問題を解決するには、Windows Update をイ

インストールする必要があります。このアップデートは <https://support.microsoft.com/kb/3080042> から利用できます。

Num Lock キーがオンの状態で Shift + Insert を使用できない

Left Shift + Insert や **Right Shift + Insert** を使用して TypeKeys メソッドでクリップボードの内容を貼り付ける操作は、Num Lock キーがオンの場合に機能しません。

TypeKeys メソッドを呼び出す前にテストスクリプトで Num Lock キーをオフにすることで、この問題を回避できます。

モバイル Web アプリケーション

Apple Safari のフレームを Silk Test Workbench がサポートしない

Silk Test Workbench は、iOS 上の Apple Safari を使用した HTML フレームおよび iframe のテストをサポートしません。

Chrome for Android 43 以降：拡大縮小とスクロールを同時に Silk Test Workbench がサポートしない

Chrome for Android 43 以降でモバイル Web アプリケーションを記録しているとき、モバイル Web アプリケーションが拡大されて左上隅が画面から見えなくなると、期待通り機能しない場合があります。モバイル Web アプリケーションのコントロールに対して緑の矩形領域が記録時に正しく表示されない場合、モバイル Web アプリケーションを完全に縮小し、モバイル Web アプリケーションの左上隅にスクロールして、記録 ウィンドウを更新してください。

Web アプリケーション

100% 以外の拡大レベルを使用して記録すると期待通り機能しない可能性がある

100% 以外の拡大レベルを使用して Web アプリケーションを記録すると、期待通り機能しない可能性があります。Web アプリケーションに対する操作を記録する前に、ブラウザの拡大レベルを 100% に設定してください。

Google Chrome

Windows で記録中のロケーターが Google Chrome で失敗する

Google Chrome で Web アプリケーションをテストするときに、アプリケーションを実行している Google Chrome インスタンスのアプリケーション構成中に複数のウィンドウが開いていると、Windows で記録中のロケーターが失敗します。アプリケーション構成中に他の Google Chrome ウィンドウを閉じると、エラーは発生しなくなります。

Google Chrome のバックグラウンド アプリケーションでオートメーション サポートを読み込むことができない

Google Chrome を使用して Web アプリケーションをテストしている場合に、**Google Chrome を閉じた際にバックグラウンド アプリケーションの処理を続行する** チェックボックスがチェックされていると、Silk Test は Google Chrome を再起動してオートメーション サポートを読み込むことができません。

Windows Aero が無効なときにモーダル ダイアログのロケーターを Silk Test が記録できない

Windows Aero が無効化されている場合、モーダル ダイアログが認識されないため、このようなダイアログのロケーターを選択できません。回避策として、モーダル ダイアログが表示されているときには、

Locator Spy または **オブジェクトの識別** ダイアログ ボックスを使用して、ロケーターを手動で作成および検証してください。

Silk Test が埋め込み PDF を表示しない

Google Chrome 42 以降を使用すると、Google Chrome は、埋め込み PDF を表示するために使用する NPAPI プラグインをデフォルトでブロックします。このため、Silk Test は埋め込み PDF を Google Chrome 42 以降に表示する代わりに、埋め込み PDF をダウンロードします。

- Google Chrome 44 以前を使用している場合、アドレスバーに以下を入力することにより、Google Chrome での NSAPI プラグインのブロックを解除できます。
chrome://flags/#enable-npapi
- Google Chrome 45 以降を使用している場合、NPAPI プラグインは Google Chrome から完全に削除されており、再有効化するオプションも無いため、すべての PDF はダウンロードされます。

テストの実行時に Google Chrome (49 以前) との接続がタイムアウトする

低速なマシンで Google Chrome (49 以前) に対するテストを実行すると、接続タイムアウトが発生し、テストが失敗する場合があります。以下のエラー メッセージが表示されます。

'*' の実行中にエラーが発生しました。ブラウザー オートメーションとの通信がタイムアウトしました。

このような接続タイムアウトを回避するために、十分な処理速度を持つテスト マシンを使用してください。たとえば、低速の仮想マシン (VM) 上でテストしている場合には、VM に CPU コアを追加することで処理速度を向上させることができます。

Internet Explorer

Google ツールバーを使用すると、Web アプリケーションの記録に支障をきたす

Internet Explorer 8 で Google ツールバーを使用すると、Web アプリケーションのロケーターの記録に支障をきたします。

Google ツールバーをオフにしてから、Web アプリケーションを記録してください。

Microsoft Silverlight アプリケーション

一部の Microsoft Silverlight アプリケーションで、Silk Test との通信の際に Internet Explorer がハングします。32 ビットプラットフォームでは、問題の防止に役立つ MS KB 2564958 (Active Accessibility の更新プログラム) を参照してください。

Silk Test 13.5 より前のバージョンの Silk Test を使用して記録したロケーターが Internet Explorer で動作しない

Silk Test 13.5 で、Internet Explorer の textContents 属性における空白文字の標準化を改良しました。この変更は、Silk Test のクロスブラウザ機能を改善するための措置で、textContents 属性を利用しているロケーターに影響を与える可能性があります。この属性は、Silk Test 13.5 以前のリリースを使用して記録されたスクリプトで使用されています。

UAC が Microsoft Windows 8 以降および Internet Explorer 11 で有効化されているとき、Open Agent に対して高い昇格を有効化できない

UAC が有効化され、Internet Explorer と Open Agent の両方を高い昇格で実行するとき、Microsoft Windows 8 以降で Internet Explorer 11 上の Web アプリケーションをテストできません。

入力方式エディター (IME) における既知の問題

- Silk Test は、Internet Explorer 11 での日本語入力において、半角スペースを記録しません。

- Silk Test は、Internet Explorer 11 の互換モードでの IME 入力を記録しません。
- 日本語 IME モードで、**スペース** を押すと、Silk Test は現在の IME の候補を記録することがあります。この問題を避けるには、**変換** を使用してください。

Java 1.7 update 71 よりも新しいバージョンを使ったアプレットのテスト時に Internet Explorer が動作を停止する場合があります

Java 1.7 update 71 (7u71) よりも新しいバージョンを使ってアプレットをテストすると、Internet Explorer が動作を停止する場合があります。

テストの実行中に Internet Explorer が応答しなくなる

Internet Explorer 10 以降でテストを実行していると、Internet Explorer のスレッドが中断され、デッドロックが発生したことが原因で、応答しなくなる場合があります。この問題は、Internet Explorer の新しいセキュリティ機能が原因です。

この問題を解決するには、セキュリティ機能を無効化します。

1. レジストリ エディタ が開きます。
2. **OverrideMemoryProtectionSetting** エントリが **HKEY_CURRENT_USER¥SOFTWARE¥Microsoft¥Internet Explorer¥Main** キーに存在しない場合は、エントリを作成します。
3. レジストリ キー **HKEY_CURRENT_USER¥SOFTWARE¥Microsoft¥Internet Explorer¥Main::OverrideMemoryProtectionSetting** の値を 2 に設定します。

JavaScript 警告処理 API メソッドが埋め込み Internet Explorer で機能しない

BrowserWindow クラスの以下の JavaScript 警告処理メソッドが、埋め込み Internet Explorer のテスト時に機能しません。

- AcceptAlert メソッド
- DismissAlert メソッド
- GetAlertText メソッド
- IsAlertPresent メソッド

Microsoft Edge

Microsoft Edge から開いたウィンドウがサポートされない

タブではなく、Microsoft Edge の新しいウィンドウとして開いたウィンドウはサポートされません。Silk Test Workbench は、このようなウィンドウを正しく閉じることができず、このようなウィンドウを閉じると、エージェントの状態が不正になります。

iframe または frame で開いた JavaScript 警告を閉じることができない

JavaScript 警告 alert()、prompt()、confirm() が、iframe または frame で開かれた場合、Microsoft Edge は閉じることができません。

ネイティブ ブラウザーのコンテキスト メニューを開けない

Microsoft Edge を右クリックして、ネイティブ ブラウザーのコンテキスト メニューを開く操作がテストに含まれていると、そのテストはハングします。Microsoft Edge によって開かれた HTML メニューでは、この問題は発生しません。

UAC を無効にした状態で Microsoft Edge に対するテストを実行できない

UAC を無効にした状態で Microsoft Edge に対してテストを実行すると、次のエラー メッセージが表示されます。

Failed to start application 'Edge'. Unable to parse remote response: Unknown error

この問題を解決するには、UAC を有効にしてください。

「Windows Feature Microsoft WebDriver がインストールされていないため、Edge を開始できません」エラーが表示される理由

Microsoft Windows 10 の Windows 10 October 2018 Update 以降のバージョンを使用している場合、オプション機能「Windows Feature Microsoft WebDriver」がシステムにインストールされていないと、Microsoft Edge 上でのテストの実行時に次のエラーが発生する場合があります。Windows Feature Microsoft WebDriver がインストールされていないため、Edge を開始できません。ヘルプのトピック「既知の問題」 > 「Microsoft Edge」を参照してください。

この問題を解決するには、システムで Windows Update が正しく機能していることを確認してください。内部の Windows Update サーバーを使用している場合は、次の手順を実行する必要があります。

1. IT 管理者に依頼して、Windows Feature Microsoft WebDriver を Windows Update サーバーに追加してもらいます。
2. Windows Feature Microsoft WebDriver をインストールします。オプション機能のインストールについては、「[Microsoft WebDriver](#)」を参照してください。

Mozilla Firefox

Adobe FlashPlayer を使用したアプリケーションの呼び出しが Mozilla Firefox を使用した場合に正しく同期されない

最近のバージョンの Adobe FlashPlayer を Mozilla Firefox で使用している場合、呼び出しが正しく同期されないことがあります。次の問題が発生する可能性があります。

- Mozilla Firefox がスクリプトの実行がハングしたものと誤って認識し、スクリプトが正しく実行されているにもかかわらず、スクリプトの実行を続けるかどうかを確認するダイアログ ボックスが表示される場合があります。
- SetFocus が正しく機能しないため、文字の入力が機能しない場合があります。
- UI には新しい値が表示されているにもかかわらず、Adobe オートメーションが古い値を返す場合があります。

Adobe FlashPlayer を使用するアプリケーションでこのような問題に遭遇した場合は、Adobe FlashPlayer の ProtectedMode をオフにしてください。詳細については、<http://forums.adobe.com/thread/1018071> を参照し、Last Resort に記述された情報をお読みください。

SAP アプリケーション

SAPTree クラスの HierarchyHeaderWidth および ColumnOrder プロパティが書き込み専用になっている

自動化に関するドキュメントに記載されている場合を除いて、SAPTree の HierarchyHeaderWidth および ColumnOrder プロパティは書き込み専用で、読み込むことはできません。

これらのプロパティを使用する場合、スクリプトで、読み込みではなく、書き込みが使用されていることを確認します。

SAPTree クラスの GetColumnIndexFromName() が「特定できないエラー」により失敗する

SapTree クラスの GetColumnIndexFromName() は「特定できないエラー」により失敗することがあります。これは SAP オートメーションの既知の問題です。

SAP Web サイトでこの問題が解決されているかどうか確認してください。

コンテキストメニュー項目の SAPTree クラスの Select() メソッドの呼び出しに失敗する

コンテキストメニュー項目の SAPTree クラスの Select() メソッドの呼び出しに失敗することがあります。

代わりに親コントロールの SelectContextMenuItem を呼び出します。これは SAP オートメーションの既知の問題です。

水平スクロールバーの Position プロパティが常に 1 を返す

水平スクロールバーの Position プロパティは常に 1 を返します。これは SAP オートメーションの既知の問題です。

SAP Web サイトでこの問題が解決されているかどうか確認してください。

SAPNetPlan クラスがサポートされていない

この問題は今後のリリースで解決される予定です。

SAP スクリプトを実行すると、再生エラーが発生する

ある状況下で、SAP テストを記録して再生すると、次のエラーが発生する場合があります：この操作を完了するのに必要なデータは、まだ利用できません。このエラーは、Silk Test Workbench が記録した操作の実行が速すぎたことを意味しています。

この問題を解決するには、テストスクリプトにスリープを追加したり、関数の呼び出しやプロパティの設定の後に Silk Test Workbench が待機する時間を増やすために、再生後の遅延を増やしてください。このオプションについての詳細は、「エージェント オプション」を参照してください。また、xBrowser テクノロジドメインではなく、SAP オートメーションを使用して問題が発生する操作を再生するように、スクリプトを変更します。たとえば、その操作を DomLink.Select から SapHTMLViewer.SapEvent に変更します。

SAPGUI クライアント 7.30 を使用するとメソッド getCurrentRow が誤った値を返す

SAPGUI クライアント 7.30 を使用しているときに、メソッド getCurrentRow を呼び出すと、メソッドは行番号ではなく誤った値 -1 を返す場合があります。

メソッド resizeWorkingPane が SAPGUI クライアント 7.30 で正しく機能しない

SAPGUI クライアント 7.30 を使用しているときに、メソッド resizeWorkingPaneEx を呼び出すと、メソッドは workingPane のサイズを変更せずに、getSapWindow().getWidth() の呼び出し結果は、ウィンドウの幅と異なる値を返します。

Silk Test Workbench を eCATT から開始すると仮想マシン上で記録を開始できない

SAPGUI から eCATT スクリプトを作成したとき、記録 ボタンを押して仮想マシンの操作を記録しようとしても、VB .NET スクリプト、ビジュアルテスト、キーワード駆動テストが Silk Test Workbench エディタで開いていると、動作しません。物理マシン上での記録操作は動作します。

eCATT から Silk Test Workbench データベースに資産インポートするとき、エクスポートした資産の XML よりも新しいバージョンの Silk Test Workbench で作成された eCATT BLOB を使用してもエラーが表示されない

eCATT から Silk Test Workbench データベースに資産をインポートするとき、Silk Test Workbench のバージョンが eCATT BLOB の資産を含む XML のバージョンよりも古い場合、インポート処理はエラーを表示せずに失敗します。

インポートが失敗すると、次に BLOB を編集しようとしたときに、「eCATT に戻る前に現在の BLOB を保存しますか?」というメッセージが表示されます。この場合、BLOB は空のため保存してはいけません。保存せずに、**いいえ** をクリックして eCATT に戻ってください。



ヒント: eCATT を使用して Silk Test Workbench の複数のインスタンスを使用している場合、Micro Focus では、すべてのインスタンスを同じ Silk Test Workbench のバージョンに合わせることを推奨しています。

Oracle Forms

Java 1.7 update 60 よりも新しいバージョンを使った Oracle Forms のテストを Silk Test Workbench がサポートしない

Silk Test Workbench は、Java 1.7 update 60 よりも新しいバージョンを使った Oracle Forms のテストをサポートしません。

Silk Test Workbench

大規模サイトでのビジュアル テストの再生

xBrowser の使用中に大規模サイトでビジュアル テストを実行すると、パフォーマンス問題が発生することがあります。この問題を回避するには、**再生 > 結果 > ビジュアル テスト > コントロールのキャプチャ オプション**を **いいえ** に設定します。

スクリプトの再生時にバックグラウンドにメッセージ ボックスが表示される (31314)

メッセージ ボックスのステートメント (MsgBox ("Hello") など) を .NET スクリプトに含めた場合、スクリプトの再生時にバックグラウンドにメッセージ ボックスが表示されます。

メッセージ ボックスをフォアグラウンドに表示するには、MsgBoxStyle.MsgBoxSetForeground ステートメントに MsgBox を含めます。例：

```
MsgBox ("Hello", MsgBoxStyle.MsgBoxSetForeground)
```

インストーラが SQL Server Native Client のインストールと COM サーバーとしての Silk Test Workbench の登録に失敗する

Silk Test のインストール中またはインストール前に、オペレーティング システムが自動更新を実行するか、またはユーザーが Windows アップデートを行った場合、インストーラは SQL Server Native Client のインストールと COM サーバーとしての Silk Test Workbench の登録に失敗します。

これらの問題は共に、Silk Central Test Manager (SCTM) の統合に影響を与えます。それは、SCTM では SQL Server Express によってインストールされたネイティブ クライアント ドライバを使用して Silk Test Workbench DSN を作成するためです。さらに SCTM は、自動化に COM インターフェイスを使用するには、COM サーバーとして登録された Silk Test Workbench を必要とするためです。

以下の解決策のいずれかを選択します。

1. Windows アップデートの実行中に Silk Test をインストールしない。
2. 使用しているマシンに Windows アップデートが自動的にインストールされないようにする。
3. 保留中の Windows アップデートを Silk Test より先にインストールしてから、システムを再起動し Silk Test をインストールする。

複数の修飾キーを含めるようホット キーの組み合わせを変更すると、追加のキーが記録される

検証の挿入用または記録の開始/停止用に複数の修飾キーを含むホットキーを構成している場合は、修飾キーを押す操作が記録されることがあります。たとえば、Alt+Ctrl+F9 というホット キーの組み合わせを指定した場合、Alt は、ホット キーの組み合わせの一部として無視されるのではなく、キーを押す操作として記録されることがあります。これは、キーから手を離すという対応する操作がないため、再生時に問題になります。

修飾キーに関連するこの操作を手動で削除するか、ホットキーの修飾キーを1つだけ使用します。

データソース名 (DNS) が 64 ビット システム で動作しない

SQL Server または Oracle の 64 ビット DSN を Silk Test Workbench に使用することはできません。64 ビット マシンで DSN を作成するには、**スタート > Silk > Silk Test > 管理 > データソース (ODBC)** をクリックして、32 ビット DSN を作成します。C:\WINDOWS\SysWOW64\odbcad32.exe にある WOW64 ツールを使用することもできます。

Silk Test Workbench を起動するのにネットワークアダプタが必要になる

Silk Test Workbench を開始するとき、Silk Test Workbench と Open Agent の間で通信できるようにするため、Silk Test Workbench をインストールしたマシンでネットワークアダプタが使用可能である必要があります。ネットワークアダプタが使用できない場合、エラー「Open Agent との接続に失敗しました」が発生します。他のクライアント (Silk Test Classic や Silk4J など) を起動する場合には、ネットワーク接続は必要ありません。

Silk Test Workbench を起動する前に、ネットワークアダプタが使用可能であることを確認してください。

SQL Server Express を Windows Vista SP1 マシンにインストールできない

Windows Vista SP1 マシンでは、SQL Server 2008 Express をインストールしようとする場合、.NET 3.5 SP1 の完全バージョンがインストールされていることを確認してください。.NET 3.5 SP1 の完全バージョンをインストールするには、Microsoft の Web サイトからダウンロードします。

[画面プレビューから識別] により、移行されたデータベースに不正なオブジェクト マップ項目が作成される

Silk Test Workbench 2010 から Silk Test Workbench 2010 R2 に移行したあと、**画面プレビューから識別** を使用すると不正なオブジェクト マップ項目が作成されます。

画面プレビューから識別 を使用する前に、**画面の更新** コマンドを使用して、画面を再キャプチャします。

子スクリプトが親スクリプトから呼び出されると、子スクリプトはスコープ外のオブジェクト マップにアクセスできる

親スクリプトと子スクリプトが異なるプロジェクトに含まれており、異なるオブジェクト マップを使用している場合、オブジェクト マップが子スクリプトのスコープ外であっても、子スクリプトは正常に実行されます。

スクリプトが正常に実行されるため、解決策は必要ありません。ただし、この動作は今後のリリースでは正常に実行されなくなります。

複数ユーザーが SQL 2008 データベース上で同じ資産を同時に編集しようすると、資産のロック機構に関する問題が発生する可能性がある

問題かどうかをテストするには、ビジュアルテストなどの資産を開いて、Silk Test Workbench で編集できるようにします。資産が開いている間に、Silk Test Workbench の別のインスタンスから同じ資産を開きます。問題が発生した場合は、Silk Test Workbench の 2 番目のインスタンスで同じ資産を同時に編集できるか、または 2 番目のインスタンスが不完全なデータに関するメッセージを受け取ります。このときユーザー名とマシン名は共に空白になります。

SQL Server で、データベースの管理者権限を、このデータベースを使用する SQL データベースユーザーに割り当てます。

SQL データベースで資産のエクスポート中に予期しないエラーが発生する

SQL データベースで資産のエクスポート中に予期しないエラー「文字列から uniqueidentifier に変換中、変換に失敗しました」が発生します。

エクスポートしたデータベースをインポートし、何らかのエラーメッセージが表示された場合 OK を押し
ます。インポートは正常に終了し、資産を使用できます。

Access データベースの使用中に予期しないエラーが発生する

Silk Test データベース メンテナンス ツールを使用してデータベースをコンパクト化します。(Microsoft
Windows 7) スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス、
または (Microsoft Windows 10) スタート > Silk > データベース メンテナンス をクリックします。次
に、データベースを開き、ツール > データベースのコンパクト化 をクリックします。

構成中に UAC が有効な状態で Silk Test Workbench が SQL Server へ接続できない

構成中に管理者権限を持たずに Silk Test Workbench を使用している場合、UAC が有効な状態で SQL
Server へ接続できません。

構成中に SQL Server へ接続するには、**管理者として実行** オプションを使用して Silk Test Workbench を
起動する必要があります。

リモートの SQL データベースとの接続が失われたときに Silk Test Workbench が異常終了する場 合がある

リモートの SQL データベースを使用している場合、データベースが Silk Test Workbench から切断され
ると (サーバーを再起動した場合など)、Silk Test Workbench がデータベースとの通信を行おうとする際に
異常終了する場合があります。

Silk Test Workbench を再起動して、データベースと再接続してください。

コントロールのキャプチャ オプションが設定されているとビジュアル テストの再生速度が著しく低下 する場がある

再生のパフォーマンスを向上させるには、コントロールのキャプチャ オプションを **いいえ** に設定します。

Microsoft Windows 7 クラシック テーマ : ビジュアル テストの画面プレビューに 記録中 ウィンドウが 含まれる場がある

クラシック テーマを使用した Microsoft Windows 7 マシンで記録したビジュアル テストの画面プレビ
ューに **記録中** ウィンドウが含まれる場合があります。Microsoft Windows 7 で **記録中** ウィンドウを含ま
ずにビジュアル テストを記録するには、クラシック テーマを使用しないでください。

資産名の変更時に参照を更新しない場合に発生する問題

ほかの資産から参照されている資産の名前を変更する時に、参照も更新するようにしてください。同時に
参照を更新しなかった場合、資産の名前を元に戻して、再度名前を変更しなおしても、Silk Test Workbench
は参照を更新を確認するダイアログを表示しないため、参照を更新できません。

名前を変更する資産が .NET スクリプトで、.NET スクリプトを参照している資産がビジュアル テストの
場合は、ビジュアル テストを保存すると、この問題を回避できます。保存することにより、Silk Test
Workbench は .NET スクリプトを見つけ出し、資産の参照が正しくなるよう更新されます。

Silk Test Workbench 言語リファレンス

このセクションでは、Silk Test Workbench がサポートする各テクノロジー ドメインのクラス、メソッド、およびプロパティを示します。

Silk Test Workbench のスクリプト言語は Visual Basic .NET です。

Visual Basic .NET のネイティブのクラス、メソッド、プロパティの詳細については、<http://msdn.microsoft.com/ja-jp/library/2x7h1hfk> を参照してください。

共通クラス リファレンス

Win32、Java SWT、および Windows Forms コントロールのテストに利用可能なクラスを一覧します。

BaseGuiTestObject クラス

説明

コンテキスト メニューを持つことのできる GUI オブジェクトの基本クラスです。

継承階層

- [GuiTestObject](#)
 - BaseGuiTestObject
 - [AccessibleControl](#)
 - [Control](#)
 - [Window](#)

構文

```
'Declaration  
Public Class BaseGuiTestObject _  
Inherits GuiTestObject _  
Implements IScrollable
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

CharSet 列挙

説明

ファイルを読み込む際に使用する文字セットを指定します。

継承階層

- [SystemFunctions](#)
 - CharSet

構文

```
Public Enumeration CharSet {
    Undefined = 100,
    ANSI = 101,
    UTF8 = 102,
    Unicode = 103
}
```

メンバー

名前	説明
Undefined	文字セットが指定されていません。
ANSI	ANSI 文字セットが使用されています。
UTF8	UTF8 文字セットが使用されています。
Unicode	Unicode 文字セットが使用されています。

CheckBox クラス

説明

チェック ボックス コントロールのクラスです。

継承階層

- [Control](#)
 - CheckBox

構文

```
'Declaration
Public Class CheckBox _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
State	チェック ボックスの状態。有効な値は、1=チェックあり、2=チェックなし、3=未設定 です。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	チェック ボックスをチェックします。
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>IsChecked</i>	チェックボックスがチェックされている場合は <i>true</i> を、それ以外の場合は <i>false</i> を返します。
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	チェックボックスを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>SetUndecided</i>	チェックボックスを未設定の状態に設定します。
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Toggle</i>	チェックボックスの状態を切り替えます。チェックボックスがチェックされている場合は、チェックを外し、そうでない場合はチェックします。チェックボックスが未設定の状態にある場合は、未設定のままになります。
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Uncheck</i>	チェックボックスのチェックを外します。

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

CheckBoxToolItem クラス

説明

ツールバー コントロール内のチェック ボックス要素のクラスです。

継承階層

- [ToolItem](#)
 - CheckBoxToolItem

構文

```
'Declaration
Public Class CheckBoxToolItem _
Inherits ToolItem
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
State	チェック ボックスの状態。有効な値は、1=チェックあり、2=チェックなし、3=未設定 です。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ComboBox クラス

説明

ポップアップ リストとテキスト フィールドの両方を兼ね備えたコントロールのクラスです。ユーザーがリストから項目を選択すると、テキスト フィールドにはその文字列が入力されます。また、ユーザーがテキスト フィールドに文字列を直接入力することもできます。

継承階層

- [Control](#)
 - ComboBox

構文

```
'Declaration
Public Class ComboBox _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	コンボ ボックス リストの項目の数。

名前	説明
Items	コンボ ボックス リストの項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
SelectedIndex	選択する項目のインデックス。
SelectedItem	選択する項目。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
ClearText	コンボ ボックスからすべてのテキストを削除します。
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	コンボ ボックスから項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetText	コンボ ボックスのテキスト フィールド内のテキストを置換します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
WaitForObject	<p>ウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

Control クラス

説明

ウィンドウの任意のコントロールのクラスです。

継承階層

- [BaseGuiTestObject](#)
 - Control
 - [CBanner](#)
 - [CheckBox](#)
 - [ComboBox](#)
 - [CoolBar](#)
 - [CTabFolder](#)
 - [DataGrid](#)
 - [DomainUpDown](#)
 - [ElementHost](#)
 - [ExpandBar](#)
 - [FormsHost](#)
 - [Group](#)
 - [Header](#)
 - [Label](#)
 - [Link](#)

- [ListBox](#)
- [ListView](#)
- [MenuStrip](#)
- [MonthCalendar](#)
- [Pager](#)
- [ProgressBar](#)
- [PushButton](#)
- [RadioList](#)
- [Sash](#)
- [SashForm](#)
- [Scale](#)
- [ScrollableControl](#)
- [ScrollBar](#)
- [ScrolledComposite](#)
- [Spinner](#)
- [StatusBar](#)
- [SWTBrowser](#)
- [SWTDateTime](#)
- [TabControl](#)
- [Table](#)
- [TextField](#)
- [ToggleButton](#)
- [ToolBar](#)
- [Tree](#)
- [UpDown](#)
- [ViewForm](#)

構文

```
'Declaration
Public Class Control _
Inherits BaseGuiTestObject
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

名前	説明
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Desktop クラス

説明

Desktop は画面全体を表すクラスです。

継承階層

- [TestObject](#)
 - Desktop

プロパティ

Desktop は [TestObject](#) のすべてのプロパティを継承します。

メソッド

Desktop は [TestObject](#) のすべてのメソッドを継承します。

Dialog クラス

説明

ダイアログのクラスです。

継承階層

- [Window](#)
 - Dialog

構文

```
'Declaration
Public Class Dialog _
Inherits Window
```

プロパティ

名前	説明
Application	このウィンドウが属するアプリケーションの名前。(Window から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)

名前	説明
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。 (IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DropDownToolItem クラス

説明

ツールバー コントロール内のドロップ ダウン要素のクラスです。

継承階層

- [ToolItem](#)
 - DropDownToolItem

構文

```
'Declaration  
Public Class DropDownToolItem _  
Inherits ToolItem
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Open	項目を開きます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FileInfo クラス

説明

ディレクトリ エントリについての情報の構造体。

継承階層

- [SystemFunctions](#)
 - FileInfo

プロパティ

名前	説明
Attributes	ファイルの属性を取得します。
CreationDate	ファイルまたはディレクトリが作成された日時を取得します。
IsDirectory	インスタンスがディレクトリについての情報を保持しているかどうかを取得します。
ModificationTime	ファイルまたはディレクトリが最後に変更された日時を取得します。
Name	ファイルの名前を取得します。
Size	ファイルのサイズを取得します。

FileSizeUnit 列挙

説明

ファイル サイズを計算する際に使用する単位を定義します。

継承階層

- [SystemFunctions](#)
 - FileSizeUnit

構文

```
public enum FileSizeUnit
```

メンバー

名前	説明
Byte	1 バイト。
KiloByte	2 ¹⁰ バイト。
MegaByte	2 ²⁰ バイト。
GigaByte	2 ³⁰ バイト。
TeraByte	2 ⁴⁰ バイト。

Group クラス

説明

コントロールを結合するために使用するクラスで、複数のコントロールが 1 つになっていることを視覚的にユーザーに表現します。このため、グループ コントロールはメソッドで表すことができる他の機能を持ちません。

継承階層

- [Control](#)
 - Group

構文

```
'Declaration  
Public Class Group _  
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

GuiTestObject クラス

説明

すべての GUI オブジェクトの基本クラスです。

継承階層

- [TestObject](#)
 - GuiTestObject
 - [BaseGuiTestObject](#)
 - [Item](#)
 - [Menu](#)

構文

```
'Declaration
Public Class GuiTestObject _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable, INativeWindow
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。
Enabled	GUI オブジェクトが有効かどうか。
Font	GUI オブジェクトのフォントの種類。
Foreground	GUI オブジェクトの前景色。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してくだ

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

HorizontalScrollBar クラス

説明

水平方向のスクロールバーのクラスです。

継承階層

- [ScrollBar](#)
 - HorizontalScrollBar

構文

```
'Declaration
Public Class HorizontalScrollBar _
Inherits ScrollBar
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーの内部ページサイズです。(ScrollBar から継承されました)
Position	スクロールバー キャレットの実際の位置です。(ScrollBar から継承されました)
Range	スクロールバーの範囲です。(ScrollBar から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロール バーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキスト メニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PageLeft</i>	スクロール バーをページ単位に左方向にスクロールします。
<i>PageRight</i>	スクロール バーをページ単位に右方向にスクロールします。
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	スクロール バーをスクロールします。(ScrollBar から継承されました)
ScrollLeft	スクロール バーを左方向にスクロールします。
ScrollRight	スクロール バーを右方向にスクロールします。
ScrollToMax	スクロール バーをその最大の位置にスクロールします。(ScrollBar から継承されました)
ScrollToMin	スクロール バーをその最小の位置にスクロールします。(ScrollBar から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

IClickable インターフェイス

説明

クリックできるオブジェクトのインターフェイスです。

構文

```
'Declaration
Public Class IClickable
```

メソッド

名前	説明
Click	オブジェクトをクリックします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。

IFocusable インターフェイス

説明

フォーカスを持つことのできるオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class IFocusable
```

メソッド

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。
SetFocus	コントロールにフォーカスを与えます。

IKeyable インターフェイス

説明

キーストロークを使用するオブジェクトのインターフェイスです。このインターフェイスのすべてのメソッドとプロパティはモバイル Web アプリケーションではサポートされていません。

構文

```
'Declaration  
Public Class IKeyable
```

メソッド

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。
ReleaseKeys	一連のキーまたはマウス ボタンを離します。
TypeKeys	一連のキーストロークをオブジェクトに送信します。
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。

IMoveable インターフェイス

説明

ウィンドウなど、移動可能なオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class IMoveable
```

プロパティ

名前	説明
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。

メソッド

名前	説明
Close	ウィンドウを閉じます。
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。
GetFocus	入力フォーカスを持つオブジェクトを返します。
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。
Maximize	ウィンドウを最大化します。
Minimize	ウィンドウを最小化します。
Move	ウィンドウを移動します。
Restore	ウィンドウを元のサイズに戻します。
SetActive	ウィンドウをアクティブにします。
Size	ウィンドウのサイズを変更します。

INativeWindow インターフェイス

説明

ネイティブ ウィンドウ ハンドルを提供するオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class INativeWindow
```

プロパティ

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。

IScrollable インターフェイス

説明

スクロールバーを持つことのできるオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class IScrollable
```

メソッド

名前	説明
GetHorizontalBar	このコントロールの水平スクロールバーを返します。
GetVerticalBar	このコントロールの垂直スクロールバーを返します。

Item クラス

説明

項目の基本クラス。

継承階層

- [GuiTestObject](#)
 - Item
 - [CoolItem](#)
 - [CTabItem](#)
 - [DataGridColumn](#)
 - [DataGridItem](#)
 - [DataGridRow](#)
 - [ExpandItem](#)
 - [MenuItem](#)
 - [SeparatorItem](#)
 - [SWTTabItem](#)
 - [SWTTreeColumn](#)
 - [TableColumn](#)
 - [TableRow](#)
 - [ToolItem](#)

構文

```
'Declaration  
Public Class Item _  
Inherits GuiTestObject
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Label クラス

説明

静的テキスト文字列のクラスです (たとえば、ラベルのようなユーザーが変更できないテキスト)。

継承階層

- [Control](#)
 - Label

構文

```
'Declaration
Public Class Label _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Link クラス

説明

ソースで参照したトピックの場所を開きます。

継承階層

- [Control](#)
 - Link

構文

```
'Declaration  
Public Class Link _  
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

ListBox クラス

説明

リスト ボックスのクラスです。

継承階層

- [Control](#)
 - ListBox

- [CheckedListBox](#)

構文

```
'Declaration
Public Class ListBox _
Inherits Control
```

プロパティ

名前	説明
AllowsMultiSelect	コントロールが複数項目の選択をサポートするかどうか。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	リスト ボックスの項目の数。
Items	リスト ボックスの項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	最初の選択された項目のインデックス。
SelectedIndices	選択した項目の索引。
SelectedItem	最初の選択された項目の名前。
SelectedItems	選択した項目の名前。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	リスト ボックスの項目をダブルクリックします。
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>ExtendSelect</i>	リスト ボックスの選択範囲を拡張して項目の範囲を選択します。
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロール バーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロール バーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>MultiSelect</i>	複数または拡張選択リスト ボックスの項目を選択します。
<i>MultiUnselect</i>	複数または拡張選択リスト ボックスの項目の選択を解除します。
<i>OpenContextMenu</i>	指定した位置でコンテキスト メニューを開きます。(<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	リスト ボックスから項目を選択します。
<i>SelectRange</i>	拡張選択リスト ボックスの項目の範囲を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Menu クラス

説明

メニューのクラスです。

継承階層

- [GuiTestObject](#)
 - Menu

構文

```
'Declaration  
Public Class Menu _  
Inherits GuiTestObject
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemCount	このメニューのメニュー項目の数を返します。メニュー項目のセパレーターも含まれます。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	メニューを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。(<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(<i>IKeyable</i> から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

MenuItem クラス

説明

メニュー項目のクラスです。

継承階層

- [Item](#)
 - MenuItem

構文

```
'Declaration
Public Class MenuItem _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Checked	メニュー項目がチェックされているかどうか。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	メニューから項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	ムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

PushButton クラス

説明

ボタンのクラスです。

継承階層

- [Control](#)
 - PushButton

構文

```
'Declaration
Public Class PushButton _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。(INativeWindow から継承されました)
Pressed	プッシュ ボタンが押されているかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	プッシュ ボタンを選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

PushToolItem クラス

説明

ツールバー コントロール内のプッシュ ボタン要素のクラスです。

継承階層

- [ToolItem](#)
 - PushToolItem

構文

```
'Declaration
Public Class PushToolItem _
Inherits ToolItem
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

RadioList クラス

説明

テキスト エディター アプリケーションの検索ダイアログの検索方向を指定するために使用される 2 つのラジオ ボタンのような、ラジオ ボタンのグループのクラスです。ラジオ リスト内の 1 つのボタンだけがある時点で選択状態になることができます。

継承階層

- [Control](#)
 - [RadioList](#)

構文

```
'Declaration  
Public Class RadioList _  
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。 (GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。 (GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。 (GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。 (GuiTestObject から継承されました)
ItemCount	ラジオ リストの項目の数。
Items	ラジオ リストの項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
SelectedIndex	選択する項目のインデックス。
SelectedItem	選択する項目。
State	チェック ボックスの状態。 有効な値は、1=チェックあり、2=チェックなし、3=未設定 です。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。 デフォルトでは値は常に <code>true</code> です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。 (GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ラジオ リストから項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

RadioListItem クラス

説明

RadioListItem はツールバー コントロール上のラジオ ボタン要素のクラスです。

継承階層

- [ToolItem](#)
 - RadioListItem

構文

```
'Declaration  
Public Class RadioListItem _  
Inherits ToolItem
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
State	チェック ボックスの状態。有効な値は、1=チェックあり、2=チェックなし、3=未設定 です。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

RegistryCategory 列挙

説明

レジストリのルート キーを指定します。詳細については、「[Predefined Keys](#)」を参照してください。

継承階層

- [SystemFunctions](#)
 - RegistryCategory

構文

```
public enum RegistryCategory
```

メンバー

名前	説明
HKEY_CLASSES_ROOT	このキーの下位にあるレジストリ エントリは、ドキュメントやプロパティの種類（またはクラス）とそれに関連するタイプを定義します。
HKEY_CURRENT_USER	このキーの下位にあるレジストリ エントリは、現在のユーザーの設定を定義します。
HKEY_LOCAL_MACHINE	このキーの下位にあるレジストリ エントリは、コンピュータの物理的な状態を定義します。
HKEY_USERS	このキーの下位にあるレジストリ エントリは、ローカル コンピュータ上の新しいユーザーに対するデフォルトのユーザー設定と、現在のユーザーに対するユーザー設定を定義します。

RegistryView 列挙

説明

32 ビットまたは 64 ビット レジストリのどちらを使うか定義します。

継承階層

- [SystemFunctions](#)
 - RegistryView

構文

```
public enum RegistryView
```

メンバー

名前	説明
RegistryView_32Bit	32 ビット レジストリを使用します。
RegistryView_64Bit	64 ビット レジストリを使用します。

Scale クラス

説明

スケールのクラスです。

継承階層

- [Control](#)
 - Scale

構文

```
'Declaration  
Public Class Scale _  
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
<i>Position</i>	スケール キャレットの実際の位置です。
<i>Range</i>	スケールの範囲です。
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Visible</i>	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(<i>GuiTestObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetPosition	スケールの位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetToMax	スケールをその最大の位置に設定します。
SetToMin	スケールをその最小の位置に設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ScrollBar クラス

説明

スクロールバーのクラスです。リスト ボックス上のスクロールバーなどのコントロールの一部のスクロールバーも含まれます。

継承階層

- [Control](#)
 - ScrollBar
 - [HorizontalScrollBar](#)
 - [VerticalScrollBar](#)

構文

```
'Declaration
Public Class ScrollBar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
PageSize	スクロールバーの内部ページ サイズです。
Position	スクロールバー キャレットの実際の位置です。
Range	スクロールバーの範囲です。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	スクロールバーをスクロールします。
ScrollToMax	スクロールバーをその最大の位置にスクロールします。
ScrollToMin	スクロールバーをその最小の位置にスクロールします。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SeparatorItem クラス

説明

メニュー セパレーターのクラスです。

継承階層

- [Item](#)
 - SeparatorItem

構文

```
'Declaration
Public Class SeparatorItem _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

TabControl クラス

説明

タブ付きまたは複数ページのダイアログ、およびボタンのクラスです。

継承階層

- [Control](#)
 - TabControl
 - [SWTTabControl](#)

構文

```
'Declaration
Public Class TabControl _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	タブ コントロールの項目の数。
Items	タブ コントロールの項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択する項目のインデックス。
SelectedItem	選択する項目。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。た

名前	説明
	だし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Select	現在のページを指定したページに設定します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイ

名前	説明
	タイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Table クラス

説明

テーブルのクラスです。

継承階層

- [Control](#)
 - Table
 - [SWTTable](#)

構文

```
'Declaration
Public Class Table _
Inherits Control
```

プロパティ

名前	説明
AllowsCheck	コントロールがチェック マークを表示できるかどうか。
AllowsMultiSelect	テーブルが複数項目の選択をサポートするかどうか。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
ColumnCount	テーブルの列の数。
ColumnItems	列のすべての項目のリスト。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
RowCount	テーブルの行の数。
RowItems	テーブルのすべての行のリスト。
SelectedIndices	選択した項目の索引。
SelectedItems	選択した項目。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Check	テーブルの定義された行のチェック ボックスをチェックします。
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClick	項目をダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExtendSelectRow	行の範囲を選択します。
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
FocusRow	テーブルの行にフォーカスを設定します。
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MultiSelectRow	テーブルの行を選択行の集合に追加します。
MultiUnselectRow	行を選択行の集合から削除します。

名前	説明
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SelectRow	テーブルの行を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Uncheck	テーブルの定義された行のチェック ボックスのチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

TableColumn クラス

説明

テーブルにおける列のクラスです。

継承階層

- [Item](#)
 - TableColumn
 - [SWTTableColumn](#)

構文

```
'Declaration
Public Class TableColumn _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)

名前	説明
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
Width	列の幅。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Resize	列のサイズを変更します。
Select	列を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステ

名前	説明
WaitForProperty	<p>ートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

TableRow クラス

説明

テーブルにおける行のクラスです。

継承階層

- [Item](#)
 - TableRow
 - [SWTTableRow](#)

構文

```
'Declaration
Public Class TableRow _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Checked	行のチェックボックスがチェックされているかどうか。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	行の項目の数。
Items	行の項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

TestObject クラス

説明

すべての UI オブジェクトの基本クラスです。

構文

```
'Declaration  
Public Class TestObject
```

プロパティ

名前	説明
<i>Text</i>	コントロールのテキスト。
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。
<i>GetProperty</i>	指定したプロパティの値を返します。
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。
HighlightObject	このオブジェクトをハイライトします。
ImageClick	指定したイメージ資産をクリックします。
ImageClickFile	指定したイメージをクリックします。
ImageExists	指定したイメージが存在するかどうかを返します。
ImageExistsFile	指定したイメージが存在するかどうかを返します。
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。
SetProperty	指定したプロパティの値を設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。
TextClick	指定したテキストの中央をクリックします。
TextExists	指定したテキストが存在するかどうかを返します。
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。

TextField クラス

説明

ユーザーが変更可能なテキストを保持する、単一行および複数行フィールドのクラスです。このクラスは、Windows RichEdit コントロールもサポートします。

継承階層

- [Control](#)
 - TextField
 - [StyledText](#)

構文

```
'Declaration
Public Class TextField _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。 (GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。 (GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。 (GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。 (GuiTestObject から継承されました)
IsPassword	コントロールがパスワード テキスト フィールドであるかどうか。
MultiLine	コントロールが複数行かどうか。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Position	テキスト フィールド内のキャレットの位置。
SelectedRange	テキスト フィールド内の選択範囲。
SelectedText	テキスト フィールド内の選択テキスト。テキストが選択されていない場合は空の文字列。
Text	コントロールのテキスト。 (TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
ClearText	すべてのテキストをテキスト フィールドから削除します
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
GetMultiText	複数行テキスト フィールドの指定された行のテキストを返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetMultiText	複数行テキスト フィールドの行のすべて、または一部を置換します。
SetPosition	テキスト フィールド内の挿入位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	単一行または複数行フィールドの指定した範囲を選択します。
SetText	テキスト フィールド内のテキストのすべて、または一部を新しいテキストに置換します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForObject	値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ToggleButton クラス

説明

ボタンを選択することで変更できる状態を持つボタンのクラスです。

継承階層

- [Control](#)
 - ToggleButton

構文

```
'Declaration
Public Class ToggleButton _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。(INativeWindow から継承されました)
State	トグル ボタンの状態。有効な値は、1=チェックあり、2=チェックなし、3=未設定 です。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForProperty	<p>に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

ToolBar クラス

説明

ツールバー コントロールのクラスです。 ツールバーはボタンや他のコントロールの集合のコンテナです。

継承階層

- [Control](#)
 - ToolBar

構文

```
'Declaration
Public Class ToolBar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが

名前	説明
	再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ToolItem クラス

説明

ツールバー コントロール内のツール項目のクラスです。

継承階層

- [Item](#)
 - ToolItem
 - [CheckBoxToolItem](#)
 - [DropDownToolItem](#)
 - [PushToolItem](#)
 - [RadioListToolItem](#)

構文

```
'Declaration
Public Class ToolItem _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Tree クラス

説明

図式的階層に構成された項目のクラスです。

継承階層

- [Control](#)
 - Tree
 - [SWTTree](#)

構文

```
'Declaration  
Public Class Tree _  
Inherits Control
```

プロパティ

名前	説明
AllowsCheck	コントロールがチェック マークを表示できるかどうか。
AllowsMultiSelect	コントロールが複数項目の選択をサポートするかどうか。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	ツリー上の項目数 (すべての子を含む)。
ItemPaths	ツリー上の項目のリスト (すべての子を含む)。
Items	ツリー上の項目のリスト (すべての子を含む)。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	最初の選択された項目のインデックス。
SelectedIndices	選択した項目の索引。
SelectedItem	最初の選択された項目の名前。
SelectedItems	選択した項目の名前。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
VisibleItemPaths	ツリー上の可視の項目のリスト (すべての子を含む)。
VisibleItems	ツリー上の可視の項目のリスト (すべての子を含む)。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Check	チェック ボックスをチェックします。
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Collapse	ツリービュー コントロール上で項目を折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClick	項目をダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Expand	ツリー ビュー コントロール上でオブジェクトを展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetItemPaths	ツリーのすべての項目を項目パスのリストとして返します。
GetItemRect	TreeView コントロールに対して相対的な項目のサイズと位置を返します。

名前	説明
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロール バーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsExpandable</i>	指定した項目が展開可能かどうかを検査します。
<i>IsExpanded</i>	指定した項目が展開されているかどうかを検査します。
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキスト メニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ツリーから項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	チェック ボックスのチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForProperty	<p>値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

VerticalScrollBar クラス

説明

垂直方向のスクロールバーのクラスです。

継承階層

- [ScrollBar](#)
 - VerticalScrollBar

構文

```
'Declaration
Public Class VerticalScrollBar _
Inherits ScrollBar
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーの内部ページサイズです。(ScrollBar から継承されました)
Position	スクロールバー キャレットの実際の位置です。(ScrollBar から継承されました)

名前	説明
Range	スクロールバーの範囲です。(ScrollBar から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PageDown	スクロールバーをページ単位に下方向にスクロールします。
PageUp	スクロールバーをページ単位に上方向にスクロールします。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	スクロールバーをスクロールします。(ScrollBar から継承されました)
ScrollDown	スクロールバーを下に移動します。
ScrollToMax	スクロールバーをその最大の位置にスクロールします。(ScrollBar から継承されました)
ScrollToMin	スクロールバーをその最小の位置にスクロールします。(ScrollBar から継承されました)
ScrollUp	スクロールバーを上移動します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Window クラス

説明

移動可能な典型的なウィンドウのクラスです。

継承階層

- [BaseGuiTestObject](#)
 - Window
 - [BrowserApplication](#)
 - [Dialog](#)
 - [FlexStandalonePlayer](#)
 - [FormsWindow](#)
 - [Shell](#)
 - [UIAWindow](#)

構文

```
'Declaration
Public Class Window _
Inherits BaseGuiTestObject _
Implements IMoveable
```

プロパティ

名前	説明
Application	このウィンドウが属するアプリケーションの名前。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetCloseWindows</i>	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(<i>IMoveable</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetFocus</i>	入力フォーカスを持つオブジェクトを返します。(<i>IMoveable</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロール バーを返します。(<i>IScrollable</i> から継承されました)
<i>GetNextCloseWindow</i>	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(<i>IMoveable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロール バーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ExecutionMode 列挙

説明

関数が実行するコマンドの完了を待機するかどうかを定義します。

継承階層

- [SystemFunctions](#)
 - executionMode

構文

```
public enum ExecutionMode
```

メンバー

名前	説明
WaitUntilFinished	Execute の実行をプログラムの終了まで 待機します。
ContinueRunning	実行可能ファイルを開始したら直ちにに戻ります。ExecutionResult.ExitCode は常に 0 で、ExecutionResult.RawOutput には何も 出力されません。

ExecutionResult クラス

説明

SystemFunctions.Execute の呼び出し結果をカプセル化します。

継承階層

- [SystemFunctions](#)
 - ExecutionResult

メソッド

名前	説明
GetExitCode	Execute の呼び出し結果を返します。エラーが発生していない場合は 0。
GetRawOutput	stdout と stderr の内容を返します。
GetOutput	stdout と stderr の内容を文字列で返します。
ToString	終了コードを文字列で返します。

FileHandle クラス

説明

論理ファイルハンドル。

継承階層

- [SystemFunctions](#)
 - FileHandle

メソッド

名前	説明
GetHandle	ハンドルの内部値を返します。
GetName	ハンドルに対応するファイルまたはディレクトリ の名前を返します。

FileOpenMode 列挙

説明

ファイルを開く方法を定義します。ファイルのモードは、[FileOpen](#) 関数 の必須パラメータです。

継承階層

- [SystemFunctions](#)
 - FileOpenMode

構文

```
public enum FileOpenMode
```

メンバー

名前	説明
Read	ファイルを読み取り用に開きます。
Write	ファイルを書き込み用に開きます。ファイルが存在しない場合は作成します。ファイルが存在する場合は、内容は削除され、0 バイトのファイルとして扱われます。
ReadWrite	ファイルを読み書き用に開きます。
Update	ファイルを書き込み用に開きますが、ファイルの内容は削除されません。デフォルトで、ほかのユーザーからの書き込みアクセスは禁止されます。ファイルのポインターは、ファイルの先頭を指します。

FilePointerMode 列挙

説明

ファイルの読み書き時に使用する位置を指定します。

継承階層

- [SystemFunctions](#)
 - FilePointerMode

構文

```
public enum FilePointerMode
```

メンバー

名前	説明
Start	ファイルの先頭にポインターを設定します。
Relative	ファイルの現在の位置にポインターを設定します。
End	ファイルの終端にポインターを設定します。

FileShareMode 列挙

説明

Silk Test がファイルを開いている間に、他のプログラムに与えるアクセス許可の種類を定義します。

継承階層

- [SystemFunctions](#)
 - FileShareMode

構文

```
public enum FileShareMode
```

メンバー

名前	説明
DenyNone	他のプログラムは Open Agent と同時に ファイルを読み書きできます。
DenyWrite	他のプログラムは Open Agent と同時に ファイルを読み込めますが、書き込みはできません。
DenyAll	他のプログラムは Open Agent と同時に ファイルにアクセスできません。

SystemFunctions クラス

説明

さまざまなオペレーションを（リモート）Open Agent のホスト上で実行できます。

継承階層

- SystemFunctions
 - [CharSet](#)
 - [ExecutionMode](#)
 - [ExecutionResult](#)
 - [FileHandle](#)

- [FileInfo](#)
- [FileOpenMode](#)
- [FilePointerMode](#)
- [FileShareMode](#)
- [FileSizeUnit](#)
- [RegistryCategory](#)
- [RegistryView](#)

構文

```
'Declaration
Public Class SystemFunctions _
```

プロパティ

名前	説明
AgentDirectory	Open Agent を実行しているディレクトリを取得します。
AgentVersion	Open Agent のバージョンを取得します。
ClipboardText	テキストをクリップボードから取得または設定します。
CurrentDrive	Open Agent をホストしているマシンの現在のドライブを取得または設定します。
CurrentRegistryView	現在のレジストリ ビューを取得または設定します。
CurrentWorkingDirectory	Open Agent の現在の作業ディレクトリを取得または設定します。
CursorPosition	Open Agent を実行しているマシンの現在のカーソルの位置を取得します。
CursorType	Open Agent を実行しているマシンの現在のカーソルの種類を取得します。
FreeMemory	Open Agent をホストしているマシンの 空きメモリの割合を取得します。
HostName	Open Agent をホストしているマシンの ホスト名を取得します。
Locale	Open Agent をホストしているマシンのロケールを取得します (en-US など)。
OSName	Open Agent をホストしているマシンの OS の名前を取得します。
OSVersionType	Open Agent をホストしているマシンの OS のバージョンの種類 (サーバーまたはクライアント) を取得します。
OSVersion	Open Agent をホストしているマシンの OS のメジャー バージョンと マイナー バージョンを取得します。

メソッド

名前	説明
CompareBinaryFiles	2 つのファイルのバイナリ比較を実行します。
CompareBitmaps	2 つのビットマップを比較します。
CompareTextFiles	2 つのファイルを辞書的に比較します。

名前	説明
<i>CopyFile</i>	ファイルをコピーします。
<i>CreateDirectory</i>	指定した名前の新しいディレクトリを作成します。
<i>CreateRegistryKey</i>	レジストリに新しいキーを作成します。指定した keyPath に存在しないすべてのレジストリ キーを再帰的に作成します。
<i>CreateRegistryValue</i>	新しい値を作成します。
<i>DeleteRegistryKey</i>	レジストリからキーを削除します。
<i>DeleteRegistryValue</i>	値を削除します。
<i>DirectoryExists</i>	指定したディレクトリが存在するかどうかを確認します。
<i>Execute</i>	Open Agent のホスト上で指定したコマンドを実行します。
<i>ExistsRegistryKey</i>	指定したキーがレジストリに存在するかどうかを確認します。
<i>ExistsRegistryValue</i>	レジストリに値が存在するかどうかを確認します。
<i>FileClose</i>	開いたファイルを閉じます。
<i>FileExists</i>	指定したファイルが存在するかどうかを確認します。
<i>FileOpen</i>	ファイルを開きます。ファイルが存在しない場合は作成します。
<i>FileReadLine</i>	ファイルから 1 行読み込みます。ファイルポインタがある現在の位置から読み込みます。
<i>FileSetPointer</i>	読み書きオペレーション用にファイル内のポインタ位置を設定します。
<i>FileWriteLine</i>	ファイルポインタがある現在の位置に 1 行書き込みます。
<i>GetBitmapCRC</i>	Open Agent を実行しているディレクトリを取得します。
<i>GetDirectoryContents</i>	ディレクトリの内容を取得します。
<i>GetEnvironmentVariable</i>	指定した環境変数の値を取得します。
<i>GetFileInfo</i>	指定したファイルについての情報を取得します。
<i>GetFreeDiskSpace</i>	指定したドライブで利用可能な空き容量を返します。
<i>GetRegistryKeyNames</i>	指定したキー以下のすべてのレジストリ キーの名前を返します。
<i>GetRegistryValueNames</i>	指定したキー以下のすべてのレジストリ キーの値を返します。
<i>GetRegistryValue</i>	指定したレジストリ キーの値を返します。
<i>MoveFile</i>	ファイルを移動します。
<i>RemoveDirectory</i>	指定したディレクトリを削除します。
<i>RemoveFile</i>	指定したファイルを削除します。
<i>SetEnvironmentVariable</i>	指定した環境変数の値を設定します。
<i>SetRegistryValue</i>	指定したレジストリ キーの値を設定します。

コア クラス リファレンス

アプリケーションを設定すると、Silk Test Workbench は標準のエージェントおよびテスト オブジェクト コントロールのテストの組み込みサポートを自動的に提供します。

ActiveData クラス

説明

ActiveData は、ActiveData オブジェクトをスクリプトで使用するための機能を提供するクラスです。このクラスは内部コンストラクターです。ActiveData を呼び出すには、このクラスを使用する代わりに、Workbench.LoadActiveData を呼び出します。

継承

ActiveData はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class ActiveData
```

プロパティ

名前	説明
ColumnCount	基になる ActiveData オブジェクトの列の数。
Count	ActiveData 行の列の数。

メソッド

名前	説明
AddColumn	基になる ActiveData オブジェクトに列を追加します。
AddRow	基になる ActiveData オブジェクトに行を追加します。
Dispose	オブジェクトで使用されるリソースをクリーン アップします。
GetEnumerator	行の列挙子を返します。
Item	ActiveDataRow をインデックス別に返します。
RemoveColumn	基になる ActiveData オブジェクトから列を削除します。
RemoveRow	基になる ActiveData オブジェクトから行を削除します。
RenameColumn	ActiveData オブジェクトの列の名前を変更します。
Reset	フォーカスを ActiveData ファイルの先頭行に移動します。
Save	ActiveData オブジェクトを保存します。

AddColumn メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる `ActiveData` オブジェクトに列を追加し、成功または失敗を示す値を返します。

構文

```
AddColumn (defaultValue, columnName)
```

変数	説明
<code>defaultValue</code>	新しい列のデフォルト値。Double。
<code>columnName</code>	列の名前。String。

使用例

PhoneBookData 資産に「Occupation」という新しい列を追加するには、以下のように入力します。

```
Public Sub Main()  
    AddColumnToDataFile( "occupation" )  
End Sub  
  
Public Sub AddColumnToDataFile( name As String )  
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )  
    data.AddColumn( "Occupation", name )  
    data.Save()  
End Sub  
End Module
```

AddRow メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる `ActiveData` オブジェクトに行を追加し、成功または失敗を示す値を返します。

構文

```
AddRow ()
```

使用例

特定の ID 番号の新しいデータ行を追加するには、以下のように入力します。

```
Public Sub Main()  
    InsertNewNumberToDataFileWithOccupation( 108, "Jay", "Jones", "5551219", _  
        "QA Analyst" )  
End Sub  
  
Public Sub InsertNewNumberToDataFileWithOccupation( id As Integer, _  
    firstName As String, lastName As String, phoneNumber As String, _  
    occupation As String )  
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )  
    Dim row As ActiveDataRow = data.AddRow()
```

```
row.SetLong( "id", id )
row.SetString( "fname", firstName )
row.SetString( "lname", lastName )
row.SetString( "number", phoneNumber )
row.SetString( "occupation", occupation )

data.Save()
```

```
End Sub
End Module
```

Dispose メソッド (ActiveData)

クラス

[ActiveData](#)。

処理


オブジェクトで使用されるリソースをクリーン アップします。

構文

```
Dispose ()
```

メモ

このメソッドは、複数のスクリプトからファイルを読み書きしている場合のみ使用する必要があります。

 **注意:** 上記以外のどの場合でも、このメソッドは使用しないでください。Silk Test Workbench は必要に応じてこのメソッドを自動的に呼び出します。

例

```
Dim a As ActiveData
a = Workbench.LoadActiveData("ActiveDataAsset")
' Do your work here
a.Dispose()
```

GetEnumerator メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

行の列挙子を返します。

構文

```
GetEnumerator ()
```

Item メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

ActiveDataRow をインデックス別に返します。

構文

Item(*index*)

変数	説明
<i>index</i>	返す行のインデックス。Integer。

RemoveRow メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトから行を削除します。

構文

RemoveRow (*rowNumber*)

変数	説明
<i>rowNumber</i>	削除する行の数 (1 から始まる)。Integer。

RemoveColumn メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトから列を削除し、成功または失敗を示す値を返します。

構文

RemoveColumn (*columnName*)

変数	説明
<i>columnName</i>	削除する行の数 (1 から始まる)。Integer。

RenameColumn メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

ActiveData オブジェクトの列の名前を変更し、列の名前が正常に変更されたかどうかを示す値を返します。

構文

RenameColumn (*columnIndex*, *newColumnName*)

変数	説明
<i>columnIndex</i>	削除する列のインデックス (1 から始まる)。Integer。
<i>newColumnName</i>	列の新しい名前。String。

Reset メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

フォーカスを ActiveData ファイルの先頭行に移動します。

構文

Reset ()

Save メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

ActiveData オブジェクトを保存します。

構文

Save ()

ColumnCount プロパティ

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトの列の数を返します。

構文

```
'Declaration  
Protected Read Property ColumnCount As Integer
```

アクセス

読み取り専用。

Count プロパティ

クラス

[ActiveData](#)。

処理

ActiveData 行の列の数を返します。

構文

```
'Declaration  
Protected Read Property Count As Integer
```

アクセス

読み取り専用。

ActiveDataRow クラス

説明

ActiveDataRow は、ActiveData オブジェクトの行を使用するための機能を提供するクラスです。このクラスは内部コンストラクターです。

継承

ActiveDataRow はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class ActiveDataRow
```

プロパティ

名前	説明
Count	ActiveData 行の列の数。
ExternalRowNumber	読み込まれた一連の行における現在のオブジェクト行番号を取得します。
RowNumber	ActiveData ファイル内における現在のオブジェクトの行番号を取得します。

メソッド

名前	説明
Dispose	行で使用されるリソースをクリーン アップします。
GetDouble	列の double 値を取得します。
GetLong	列の integer 値を取得します。
GetLongLong	列の long 値を取得します。

名前	説明
GetString	列の string 値を取得します。
SetDouble	列の double 値を設定します。
SetLong	列の integer 値を設定します。
SetLongLong	列の long 値を設定します。
SetString	列の string 値を設定します。

Dispose メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。


処理

行で使用されるリソースをクリーン アップします。

構文

```
Dispose ()
```

メモ

 **注意:** このメソッドは使用しないでください。 Silk Test Workbench は必要に応じてこのメソッドを自動的に呼び出します。

GetDouble メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の double 値を取得します。

構文

```
GetDouble (columnIndex)
```

または

```
GetDouble (columnName)
```

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer.。
<i>columnName</i>	列の名前。String。

GetLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の integer 値を取得します。

構文

GetLong (*columnIndex*)

または

GetLong (*columnName*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。

GetLongLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の long 値を取得します。

構文

GetLongLong (*columnIndex*)

または

GetLongLong (*columnName*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。

GetString メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の string 値を取得します。

構文

GetString (*columnIndex*)

または

GetString (*columnName*)

変数	説明
<code>columnIndex</code>	列のインデックス (1 から開始)。Integer。
<code>columnName</code>	列の名前。String。

SetDouble メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の double 値を設定します。

構文

`SetDouble (columnIndex, value)`

または

`SetDouble (columnName, value)`

変数	説明
<code>columnIndex</code>	列のインデックス (1 から開始)。Integer。
<code>columnName</code>	列の名前。String。
<code>value</code>	新しい値。Integer または Double。

SetLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の integer 値を設定します。

構文

`SetLong (columnIndex, value)`

または

`SetLong (columnName, value)`

変数	説明
<code>columnIndex</code>	列のインデックス (1 から開始)。Integer。
<code>columnName</code>	列の名前。String。
<code>value</code>	新しい値。Integer。

SetLongLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の long 値を設定します。

構文

```
SetLongLong (columnIndex, value)
```

または

```
SetLongLong (columnName, value)
```

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。
<i>value</i>	新しい値。Long。

SetString メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の string 値を設定します。

構文

```
SetString (columnIndex, value)
```

または

```
SetString (columnName, value )
```

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。
<i>value</i>	新しい値。String。

Count プロパティ

クラス

[ActiveData](#)。

処理

ActiveData 行の列の数を返します。

構文

```
'Declaration  
Protected Read Property Count As Integer
```

アクセス

読み取り専用。

ExternalRowNumber プロパティ

クラス

[ActiveDataRow](#)。

処理

読み込まれた一連の行における現在のオブジェクトの行番号を取得します。

構文

```
'Declaration  
Protected Read Property ExternalRowNumber As Integer
```

アクセス

読み取り専用。

RowNumber プロパティ

クラス

[ActiveDataRow](#)。

処理

実際のファイル内の現在のオブジェクトの行番号を取得します。

構文

```
'Declaration  
Protected Read Property RowNumber As Integer
```

アクセス

読み取り専用。

Agent クラス

説明

Agent は、グラフィカル ユーザー インターフェイスを操作する Silk Test Workbench のコンポーネントである Open Agent のクラスです。

Agent では識別子 Agent が事前に定義されています。Agent のメソッドを呼び出すには、以下の構文を使用します。

```
Agent.method()
```


継承

Agent はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class Agent
```

プロパティ

名前	説明
Desktop	ルート GUI オブジェクトを取得します。
SystemFunctions	SystemFunctions オブジェクトを取得します。このオブジェクトを使うと、Open Agent がインストールされたローカル マシンやリモート マシン上で、OS 固有の操作を実行できます。

メソッド

名前	説明
Attach	指定したアプリケーションにアタッチします。コマンドラインパターンおよびテクノロジードメインを追加で指定することができます。
Connect	指定したホストとポートを使用して指定したマシン上のエージェントに接続します。
Decrypt	指定された文字列を復号化します。 Encrypt メソッドで暗号化された文字列のみを復号化することができます。
DetachAll	現在アタッチされているすべてのアプリケーションを解除します。
Encrypt	指定された文字列を暗号化します。
ExecuteBaseState	テスト対象アプリケーションが実行中でテストの準備ができていることを保証します。
GetOption	エージェント オプションの値を取得します。
ResetOptions	エージェントのすべてのオプションをデフォルト値に戻します。
SetOption	エージェント オプションを設定します。このオプションは、すべてのテクノロジードメインに対して利用されます。
Shutdown	エージェントを停止します。テスト実行が完了したときにのみ呼び出されるべきです。

Attach メソッド

クラス

[Agent](#)。

処理

指定したアプリケーションにアタッチします。コマンドラインパターンおよびテクノロジードメインを追加で指定することができます。

構文

```
Attach (executablePattern, commandLinePattern, techDomains)
```

変数	説明
<code>executablePattern</code>	アプリケーションの実行可能名。パターンには、ワイルドカード '?' および '*' を使用できます (それぞれ、1 文字、または複数文字に一致)。例： <code>myApplication.exe</code> 、 <code>myApplica?ion.exe</code> 、 <code>myApp*.exe</code> 。String。
<code>commandLinePattern</code>	省略可能：このパターンはアプリケーションのコマンドライン引数に対して照合します。これは、アプリケーションの複数のインスタンスが実行中で、その中のいくつかだけがテスト対象である場合に有効です。Java アプリケーションの場合、特徴的な jar や main クラスの名前をコマンドラインパターンに含めることができます。例： <code>*org.MyMainClass</code> 。特定のテクノロジードメインをロードしたいときに、コマンドラインパターンが無いことを指定する場合には、 <code>""</code> を使用できます。String。
<code>techDomains</code>	省略可能：アプリケーションに読み込むテクノロジードメイン。通常は、このパラメータはテクニカルサポートによってのみ使用されます。テクノロジードメインを指定しないことをお勧めします。テクノロジードメインを指定しない場合は、すべての利用可能なテクノロジードメインが読み込まれます。TechDomain。

使用例

次の例では、メモ帳にアタッチします。

```
Agent.Attach("notepad.exe")
```

次の例では、アプリケーションの複数のインスタンスが実行されている場合に、コマンドラインパターンを使用してプロセスの特定のインスタンスにアタッチする方法を示します。

```
Agent.Attach("javaw.exe", "*org.MyMainClass")
```

以下の例は、Internet Explorer にアタッチして Win32 テクノロジードメインをロードする方法を示しています。

```
Agent.Attach("iexplore.exe", "", {"WIN32"})
```

Connect メソッド

クラス

[Agent](#)。

処理

指定したホストとポートを使用して指定したマシン上のエージェントに接続します。

エージェントが実行されていない場合、またはエージェントがこのマシンに到達できない場合は、例外がスローされます。

指定したホストがローカルホストの場合は、エージェントが起動します。

このエージェントとの接続を最初に確立した後に、すべてのエージェント オプションをデフォルト値にリセットするには、`Agent.ResetOptions()` を呼び出します。

構文

```
Connect (host, [port])
```

変数	説明
<i>host</i>	Open Agent のホスト。String。
<i>port</i>	省略可能: Open Agent のポート。ポートが指定されていない場合、デフォルトのポートが使用されます。このポートは、Information Service のポートで、デフォルト値は 22901 です。たとえば、別の Information Service のポートでエージェントに接続するには、 「Agent.Connect("myRemoteMachine", portnumber) 」と入力します。 リモート エージェントに接続している場合に、ローカル エージェントに接続するには、 「Agent.Connect("localhost")」と入力します。 Integer。

Decrypt メソッド

クラス

[Agent](#)。

処理

指定された文字列を復号化します。Encrypt メソッドで暗号化された文字列のみ復号化することができます。

構文

```
Decrypt(toDecrypt)
```

変数	説明
<i>toDecrypt</i>	復号化する文字列。String。

DetachAll メソッド

クラス

[Agent](#)。

処理

現在アタッチされているすべてのアプリケーションを解除します。

構文

```
DetachAll( )
```

Encrypt メソッド

クラス

[Agent](#)。

処理

指定された文字列を暗号化します。

構文

Encrypt (*toEncrypt*)

変数	説明
<i>toEncrypt</i>	暗号化する文字列。String。

ExecuteBaseState メソッド

クラス

[Agent](#)。

処理

テスト対象アプリケーションが実行中でテストの準備ができていることを保証します。

構文

TestObject ExecuteBaseState(*baseState*)

変数	説明
<i>baseState</i>	基本状態を実行するために必要なすべての情報を含みます。IBaseState。

GetOption メソッド

クラス

[Agent](#)。

処理

エージェント オプションの値を取得します。

構文

GetOption(*name*)

変数	説明
<i>name</i>	オプションの名前。使用できるオプションのリストは Options クラスで定義されています。String。

例

たとえば、以下のように入力します。

```
desktop.getOption(Options.ObjectResolveTimeout)
```

ResetOptions メソッド

クラス

[Agent](#)。

処理

エージェントのすべてのオプションをデフォルト値に戻します。

構文

```
ResetOptions()
```

SetOption メソッド

クラス

[Agent](#)。

処理

エージェント オプションを設定します。このオプションは、すべてのテクノロジー ドメインに対して利用されます。

構文

```
SetOption(name, value)
```

変数	説明
<i>name</i>	オプションの名前。使用できるオプションのリストは Options クラスで定義されています。String。
<i>value</i>	オプションの値。Object。

例

たとえば、以下のように入力します。

```
Agent.setOption(Options.ObjectResolveTimeout, true)
```

Shutdown メソッド (Agent)

クラス

[Agent](#)。

処理

夜間テスト実行後などのテスト実行後に Open Agent を停止する場合にこのメソッドを呼び出します。このメソッドは、テスト実行の最後にのみ呼び出されるべきで、テストの操作が実行される前に呼び出されるべきではありません。シャットダウン後に操作が実行されると、エージェントは再起動します。Open Agent を停止する方法として、次のコマンドをコマンド ラインから実行する方法もあります：

```
openAgent.exe -shutDown
```

構文

VB

```
Shutdown()
```

例 : Shutdown メソッドの使用

```
// VB code  
<AssemblyCleanup(>
```

```
Public Shared Sub ShutdownAgentAfterTestRun()  
    Agent.Shutdown()  
End Sub
```

Desktop プロパティ

クラス

[Agent](#)。

処理

ルート GUI オブジェクトを取得します。

構文

```
'Declaration  
Protected Read Property Desktop As Desktop
```

アクセス

読み取り専用。

SystemFunctions プロパティ (Agent)

クラス

[Agent](#)。

処理

SystemFunctions オブジェクトを取得します。このオブジェクトを使うと、Open Agent がインストールされたローカル マシンやリモート マシン上で、OS 固有の操作を実行できます。利用可能なシステム関数についての情報は、「SystemFunctions クラス」トピックを参照してください。

構文

```
'Declaration  
Public Read Property SystemFunctions As SystemFunctions
```

アクセス

読み取り専用。

例

たとえば、メモ帳をリモート マシン上で開始する場合は、次のコードをスクリプトに記述します。

```
Agent.SystemFunctions.Execute("notepad.exe")
```

BaseState クラス

説明

BaseState クラスを使用すると、テスト対象アプリケーションが実行中でテストの準備ができていたことが保証されます。また、テスト対象アプリケーションが前面に配置されます。

エージェントをアプリケーションにアタッチしようとした場合に、BaseState.Locator で指定されたテスト オブジェクトが見つかったら、基本状態としてこのテスト オブジェクトを含むウィンドウが前面に配置さ

れ、見つかったテスト オブジェクトが即座に返されます。テスト オブジェクトが見つからない場合、baseStateInfo で指定したとおりにアプリケーションがコマンド ラインから起動し、エージェントがブラウザにアタッチされます。テスト オブジェクトが見つかるまで、エージェントは待機します。待機するためのタイムアウトは、オプション Options.ObjectResolveTimeout.タイムアウトのデフォルト値は、30 秒です。タイムアウトを変更するには、Desktop.SetOption (String, Object) を参照してください。**オプション** ダイアログ ボックスの **再生 > タイミング** にある **アプリケーション準備完了タイムアウト** を編集して UI 上からタイムアウトを変更することもできます。指定したタイムアウト内にテスト オブジェクトが見つかったら、このテスト オブジェクトを含むウィンドウが前面に配置され、見つかったテスト オブジェクトが返されます。テスト オブジェクトが見つからずにタイムアウトした場合は、例外がスローされます。

継承

BaseState はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration
Public Class BaseState _
Implements IBaseState
```

プロパティ

名前	説明
CommandLineArguments	コマンド ライン引数を取得または設定します。
CommandLinePattern	コマンド ライン パターンを取得または設定します。
ExecutablePattern	実行可能パターンを取得または設定します。
Executable	実行可能ファイルの名前を取得します
Locator	実行可能ファイルのロケータを返します
TechDomains	テストに使用するテクノロジー ドメインのリストを取得または設定します
WorkingDirectory	作業ディレクトリを取得または設定します。

CommandLineArguments プロパティ

クラス

[BaseState](#)。

処理

コマンド ライン引数を取得または設定します。

構文

```
'Declaration
Public ReadWrite Property CommandLineArguments As String
```

アクセス

読み書き。

CommandLinePattern プロパティ

クラス

[BaseState](#)。

処理

コマンド ライン パターンを取得または設定します。

構文

```
'Declaration  
Public ReadWrite Property CommandLinePattern As String
```

アクセス

読み書き。

ExecutablePattern プロパティ

クラス

[BaseState](#)。

処理

実行可能パターンを取得または設定します。

構文

```
'Declaration  
Public ReadWrite property ExecutablePattern As String
```

アクセス

読み書き。

Executable プロパティ

クラス

[BaseState](#)。

処理

実行可能ファイルの名前を返します。

構文

```
'Declaration  
Public Read Property CommandLineArguments As String
```

アクセス

読み取り専用。

Locator プロパティ

クラス

[BaseState](#)。

処理

実行可能ファイルのロケーターを取得します

構文

```
'Declaration  
Public Read Property Locator As String
```

アクセス

読み取り専用。

TechDomains プロパティ

クラス

[BaseState](#)。

処理

実行可能ファイルに必要なテクノロジー ドメインを取得します。

構文

```
'Declaration  
Public Read Property TechDomains As List of TechDomain
```

アクセス

読み取り専用。

WorkingDirectory プロパティ

クラス

[BaseState](#)。

処理

作業ディレクトリを取得または設定します。Windows 環境変数を含めることができます。

構文

```
'Declaration  
Public ReadWrite Property WorkingDirectory As String
```

アクセス

読み書き。

BrowserBaseState クラス

説明

Executable で指定されたブラウザが適切に実行されていて、テストの準備ができていることを保証します。さらに、基本状態として Url によって指定された URL に移動し、ブラウザを前面に持ってきます。ブラウザの基本状態は、以下のように実行されます。

- エージェントをブラウザにアタッチします。
 - Locator で適切に指定されたテスト オブジェクトが見つかり、基本状態として所定の URL に移動し、ブラウザが前面に配置され、見つかったテスト オブジェクトが返されます。
 - テスト オブジェクトが見つからない場合、browserBaseStateInfo で指定したとおりにブラウザがコマンドラインから起動し、エージェントがブラウザにアタッチされます。
- テスト オブジェクトが見つかるまで、エージェントは待機します。タイムアウトは、オプション Options.ObjectResolveTimeout.タイムアウトを変更するには、Desktop クラスの SetOption メソッドを参照してください。
 - 指定したタイムアウト内にテスト オブジェクトが見つかり、基本状態として所定の URL に移動し、ブラウザが前面に配置され、見つかったテスト オブジェクトが返されます。
 - テスト オブジェクトが見つからずにタイムアウトした場合は、例外がスローされます。

継承

BrowserBaseState はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class BrowserBaseState _  
Implements IBaseState
```

プロパティ

名前	説明
CommandLineArguments	コマンドライン引数を取得または設定します。(BaseState から継承)。
CommandLinePattern	コマンドラインパターンを取得または設定します。(BaseState から継承)。
ExecutablePattern	実行可能パターンを取得または設定します。(BaseState から継承)。
Executable	実行可能ファイルの名前を取得します(BaseState から継承)。
Locator	実行可能ファイルのロケータを返します(BaseState から継承)。
TechDomains	テストに使用するテクノロジードメインのリストを取得または設定します(BaseState から継承)。
Url	移動する URL を取得します。
WorkingDirectory	作業ディレクトリを取得または設定します。(BaseState から継承)。

URL プロパティ

クラス

[BrowserBaseState](#)。

処理

移動する URL を取得します。

構文

```
'Declaration  
Public Read Property Url As String
```

アクセス

読み取り専用。

ConsoleWindow クラス

説明

ConsoleWindow はコンソールのクラスです。

継承

ConsoleWindow はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class ConsoleWindow
```

メソッド

名前	説明
Close	コンソールを閉じます。
CloseAll	すべての開いているコンソールまたは特定のキャプションを持つすべてのコンソールを閉じます。
GetContents	コンソールの内容を返します。
Open	新しいコンソールを開きます。
TypeKeys	一連のキーストロークをコンソールに送信します。

Close メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

コンソールを閉じます。

構文

```
consoleWindow.Close ([sItemIdentifier])
```

変数	説明
<i>sItemIdentifier</i>	コンソールウィンドウのインデックスまたはウィンドウのキャプション。このパラメータが指定されていない場合、Silk Test Workbench は最初に見つけたコンソールウィンドウを閉じます。STRING。

例 1

次の例では、インデックスを使用して 2 番目のコンソール ウィンドウを閉じる方法を示します。

```
consoleWindow.Close(2)
```

例 2

次の例では、コンソール ウィンドウの完全識別子を使用してコンソール ウィンドウを閉じる方法を示します。

```
consoleWindow.Close("C:¥¥Windows¥¥system32¥¥cmd.exe[1]")
```

CloseAll メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

すべての開いているコンソールまたは特定のキャプションを持つすべてのコンソールを閉じます。

構文

VB

```
consoleWindow.CloseAll ([sCaption])
```

変数	説明
<i>sCaption</i>	このパラメータを設定すると、このキャプションを持つコンソールだけが閉じられます。STRING

例

次の例では、キャプション MyCommandWindow を持つすべてのコンソール ウィンドウを閉じる方法を示します：

```
// VB  
consoleWindow.CloseAll("MyCommandWindow")
```

GetContents メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

コンソールの内容を返します。

構文

```
IsContents = consoleWindow.GetContents ([sItemIdentifier])
```

変数	説明
<i>IsContents</i>	コンソールの内容。LIST OF STRING。
<i>sItemIdentifier</i>	コンソール ウィンドウのインデックスまたはウィンドウのキャプション。このパラメータが指定されていない場合、Silk Test Workbench は最初に見つけたコンソール ウィンドウの内容を返します。STRING。

メモ

- 返された文字列リスト内の各文字列は、コンソール内の 1 行に対応しています。

例 1

次の例では、インデックスを使用して 2 番目のコンソール ウィンドウの内容を取得する方法を示します。

```
IsContents = consoleWindow.GetContents(2)
```

例 2

次の例では、コンソール ウィンドウの完全識別子を使用してコンソール ウィンドウの内容を取得する方法を示します。

```
IsContents = consoleWindow.GetContents("C:¥¥Windows¥¥system32¥¥cmd.exe[1]")
```

Open メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

新しいコンソールを開きます。

使用可能な資産

この機能がサポートされるのは、Open Agent を使用している場合のみです。

構文

VB

```
ConsoleWindow.Open ([caption])
```

変数	説明
<i>caption</i>	新しいコンソール ウィンドウのキャプション。String。

例

次の例では、キャプション "My Console" の新しいコンソールを開く方法について示します。

```
// VB  
ConsoleWindow.Open("My Console")
```

TypeKeys メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

一連のキーストロークをコンソールに送信します。

構文

```
ConsoleWindow.TypeKeys (sEvents [, nDelay, ensureFocus, sItemIdentifier])
```

変数	説明
<i>sEvents</i>	入力するキーストローク。STRING。
<i>nDelay</i>	省略可能 : キーストローク間の遅延 (秒単位、小数可)。NUMBER。
<i>ensureFocus</i>	省略可能 : アプリケーションを自動的に前面に移動し、メソッドを実行したコントロールにキーボードのフォーカスを設定するかどうか。この処理は、最初のキーを入力する前に実行されるため、入力するキーを複数指定した場合、キーストロークごとに再確認されることはありません。この変数は、デフォルトで TRUE に設定されています。BOOLEAN。
<i>sItemIdentifier</i>	コンソール ウィンドウのインデックスまたはウィンドウのキャプション。このパラメータが指定されていない場合、Silk Test Workbench は最初に見つけたコンソール ウィンドウにキーストロークを送信します。STRING。

メモ

- 開いているコンソールがない場合は、TypeKeys によってコンソールが開きます。
- Windows の場合は、入力する前に TypeKeys によってコンソールに入力フォーカスが設定されます。
- *nDelay* を指定した場合、この値は、SetOption で設定された、OPT_KEYBOARD_DELAY オプションを使用した設定値よりも優先します。
- キーストロークが認識されない場合は、使用する遅延の値を大きくすることができます。
- キーの後に、そのキーの入力頻度を定義するカウンターを配置できます。カウンターは遅延と組み合わせることができます。
- 遅延時間中は画面がロックされ、マウスおよびキーボードとの対話は不可能になります。

- DOM に拡張機能を使用して記録している場合、TypeKeys("<Tab>") はキャプチャされません。入力するオブジェクトはスクリプトで直接参照されているため、この手動の Tab を記録する必要はありません。必要に応じて、スクリプトに TypeKeys("<Tab>") を手動で入力できますが、この Tab は記録されません。
- **Ctrl-Shift** などのキー グループを指定した場合、プラスおよびマイナス記号はサポートされます。

例

次のサンプル コードは、コンソール ウィンドウで dir コマンドを実行し、ディレクトリのファイルとフォルダをリストします。

```
// VB
ConsoleWindow.TypeKeys ("dir<Enter>")
```

DllCall クラス

説明

DllCall クラスを使用すると、NTF ユーザーはエージェントまたは AUT のいずれかで、任意の DLL の公開された DLL 機能呼び出すことができます。

継承

DllCall はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration
Public Class Agent
```

メソッド

名前	説明
UnloadAllDllsFromAgent	エージェントからすべての DLL をアンロードします。
UnloadDllFromAgent	指定した DLL をエージェントからアンロードします。

UnloadAllDllsFromAgent メソッド

クラス

[DllCall](#)。

処理

エージェントからすべての DLL をアンロードします。

構文

```
Agent.UnloadAllDllsFromAgent ()
```

UnloadDllFromAgent メソッド

クラス

[DllCall](#)。

処理

指定した DLL をエージェントからアンロードします。

構文

```
bSuccessful = Agent.UnloadDll (sDllName)
```

変数	説明
<i>bSuccessful</i>	DLL が正常にアンロードされた場合は true、失敗した場合は false。BOOLEAN。
<i>sDllName</i>	DLL ファイルの名前。STRING。

メモ

UnloadDllFromAgent は、標準 DLL をアンロードする場合のみ使用できます。混在アセンブリをアンロードする場合は、使用できません。混在アセンブリをアンロードするには、UnloadAllDllsFromAgent を使用します。

IBaseState インターフェイス

説明

基本状態のインターフェイスです。基本状態は、アプリケーションをテスト可能な状態にします。必要に応じてアプリケーションを起動して、あるロケーターが見つかるまで待機し、アプリケーションを前面に持ってきます。

構文

```
'Declaration  
Public Interface IBaseState
```

メソッド

名前	説明
Execute	デスクトップで指定したマシン上で基本状態を実行します。

Execute メソッド

クラス

[BaseState](#)。

処理

デスクトップで指定したマシン上で基本状態を実行します。

構文

```
TestObject = BaseState.Execute()
```

変数	説明
<i>TestObject</i>	ロケーターが待機するオブジェクト。

例：基本状態でのブラウザと Web アプリケーションの URL の指定

Web アプリケーションをテストするブラウザとして Internet Explorer を、Web アプリケーションの URL として `www.borland.com` を指定するには、次のコードを使用できます。

```
Dim baseState = New BrowserBaseState(BrowserType.InternetExplorer,
"www.borland.com")
baseState.Execute()
```

さらに、基本状態を変数に格納するには：

```
Dim baseState = New BrowserBaseState(BrowserType.InternetExplorer,
"www.borland.com")
Dim browserApplication As BrowserApplication = baseState.Execute()
```

モバイルデバイス上のモバイルブラウザ（iOS デバイス上の Safari など）を指定するには、次のコードを使用できます。

```
Dim baseState = New BrowserBaseState(BrowserType.Safari,
"www.borland.com")
baseState.MobileDeviceName = "My iPhone"
baseState.Execute()
```

例：実行可能ファイルへのパスと AUT のメイン ウィンドウのロケータの指定

AUT の実行可能ファイルへのパスとして `C:/windows/system32/notepad.exe` を、AUT のメイン ウィンドウのロケータとして `//Window[@caption='*Notepad*']` を指定するには、次のコードを使用できます。

```
Dim baseState = New BaseState("C:/windows/system32/notepad.exe", "//
Window[@caption='*Notepad*']")
baseState.Execute()
```

さらに、基本状態を変数に格納するには：

```
Dim baseState = New BaseState("C:/windows/system32/notepad.exe", "//
Window[@caption='*Notepad*']")
Dim mainWindow As Window = baseState.Execute()
```

Timer クラス

説明

Timer クラスを使用すると、経過時間を正確に計測できます。

特に、Timer クラスのメソッドとプロパティは、Silk Performer から呼び出されるテスト実行の計測に使用できるという利点があります。Silk Test Workbench と Silk Performer の統合についての詳細は、『[Silk Performer ヘルプ](#)』を参照してください。

継承

BaseState はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration
Public Class Timer
```

メソッド

名前	説明
Pause	タイマーの計測を一時停止します。
Resume	タイマーの計測を再開します。
Start	タイマーの計測を開始または再開します。
StartNew	新しい Timer インスタンスを初期化し、経過時間を 0 に設定してから、経過時間の計測を開始します。
Stop	タイマーの計測を停止します。

プロパティ

名前	説明
Elapsed	タイマーの経過時間を取得します。
ElapsedMilliseconds	タイマーの経過時間をミリ秒で取得します。
IsRunning	タイマーが現在実行されているかどうかを取得します。
Name	Silk Performer とのやり取りに使用するタイマーの名前を取得または設定します。

Pause メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーの計測を一時停止します。

構文

```
timer.Pause ()
```

Resume メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーの計測を再開します。

構文

```
timer.Resume ()
```

Start メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーが現在停止している場合、タイマーの計測を開始します。タイマーが現在一時停止している場合、タイマーの計測を再開します。

構文

```
timer.Start ()
```

StartNew メソッド (Timer)

クラス

[Timer](#)。

処理

新しい Timer インスタンスを初期化し、経過時間を 0 に設定してから、経過時間の計測を開始します。名前は Silk Performer とやり取りするために提供されます。

構文

```
timer=StartNew ([timerName])
```

変数	説明
<i>timer</i>	経過時間の測定を開始した新しい Timer インスタンス。 Timer。
<i>timerName</i>	省略可能：新しいタイマーの名前。名前は Silk Performer とやり取りするために提供されます。 String。

Stop メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーの計測を停止します。

構文

```
timer.Stop()
```

Elapsed プロパティ (Timer)

クラス

[Timer](#)。

処理

タイマーの経過時間を取得します。

構文

```
'Declaration  
Public Read Property Elapsed As TimeSpan
```

アクセス

読み取り

ElapsedMilliseconds プロパティ (Timer)

クラス

[Timer](#)。

処理

タイマーの経過時間をミリ秒で取得します。

構文

```
'Declaration  
Public Read Property ElapsedMilliseconds As Long
```

アクセス

読み取り

IsRunning プロパティ (Timer)

クラス

[Timer](#)。

処理

タイマーが現在実行されているかどうかを取得します。タイマーが実行している場合は True を返します。

構文

```
'Declaration  
Public Read Property IsRunning As Boolean
```

アクセス

読み取り

Name プロパティ (Timer)

クラス

[Timer](#)。

処理

Silk Performer とのやり取りに使用するタイマーの名前を取得または設定します。

構文

```
'Declaration  
Public ReadWrite Property Name As String
```

アクセス

読み書き。

Workbench クラス

説明

Workbench は、検証を追加したり、子スクリプトを実行するための機能を提供するクラスです。

継承

Workbench はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class Workbench
```

Workbench のメソッドを呼び出すには、以下の構文を使用します。

```
Workbench.method()
```

メソッド

名前	説明
LoadActiveData	ActiveData オブジェクトを読み込んで、スクリプトで使用します。
ResultComment Method	結果 ウィンドウの 詳細 タブに、コメントをステップとして追加します。
RunScript	名前によって識別される子スクリプトを、パラメータを渡して実行します。
Verify	予想した値と実際の値が等しいかどうかを検証します。コメントを指定することもできます。
VerifyAsset	検証資産を実行します。

LoadActiveData メソッド

クラス


[Workbench](#)。

処理

スクリプトで使用する ActiveData 資産を読み込みます。

構文

```
Workbench.LoadActiveData("activeDataName", [sheetName, startRow, endRow,  
randomCount, readOnly])
```

変数	説明
<code>activeDataName</code>	<p>スクリプトで使用する <code>ActiveData</code> 資産の名前。STRING。</p> <p> 注: 環境内で複数のプロジェクトを使用する場合、Common プロジェクト内にある、名前が一意でない <code>ActiveData</code> 資産を挿入するには、プロジェクト修飾子を使用します。詳細については、例を参照してください。</p>
<code>sheetName</code>	<p>省略可能：使用するシートの名前。ここで指定したシートは、<code>ActiveData</code> 資産で指定したシートを上書きします。デフォルトでは、資産で指定したシートが使用されます。STRING</p>
<code>startRow</code>	<p>省略可能：最初に読み込む <code>ActiveData</code> 資産内の行のインデックスです。INTEGER。</p>
<code>endRow</code>	<p>省略可能：最後に読み込む <code>ActiveData</code> 資産内の行のインデックスです。INTEGER。</p>
<code>randomCount</code>	<p>省略可能：資産に関連付けられた <code>ActiveData</code> ファイルのレコードを <code>ActiveData</code> テストで使用するかどうか、およびその方法を決定します。値が 0 または <code>False</code> の場合は、ランダム モードがオフになります。値が -1 または <code>True</code> の場合は、値が <code>StartRow</code> から <code>EndRow</code> のすべての行が <code>ActiveData</code> テストでランダムな順番で使用されます。値が 1 から N (N はテストデータ資産の行数) の場合は、<code>StartRow</code> から <code>EndRow</code> の範囲でのランダムな行数として使用されます。<code>RandomCount</code> パラメータの許容値とともに、さまざまな <code>StartRow</code> 値および <code>EndRow</code> 値を使用する場合の詳細については、「アクティブデータに使用するデータの指定」を参照してください。INTEGER。</p>
<code>readOnly</code>	<p>省略可能：<code>ActiveData</code> 資産を読み取り/書き込み可能で開くか、読み取り専用で開くかを決定します。読み取り/書き込み可能で <code>ActiveData</code> 資産を開くには、<code>False</code> に設定します。<code>ActiveData</code> 資産を読み取り専用として開くには、<code>True</code> に設定します。読み取り専用を設定すると、スクリプトの <code>ActiveData</code> テストの部分で <code>ActiveData</code> 資産に書き込みを行おうとした場合に実行時エラーが生成されます。デフォルトでは、<code>ActiveData</code> 資産は読み取り/書き込み可能で開かれます。BOOLEAN。</p>

使用例

スクリプトに `ActiveData` 内のランダムな行を含めるには、次のように入力します。

```
Public Sub Main()
    AddRandomToPhoneBook( 3 )    'Reads 3 rows in a random order
End Sub

Public Sub AddRandomToPhoneBook( howMany As Integer )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData", 1, -1,
howMany )
    Dim row As ActiveDataRow

    With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
        For Each row In data
            Dim FirstName As String = row.GetString("fname")
            Dim LastName As String = row.GetString("lname")
            Dim PhoneNumber As String = row.GetString("number")

            .DomTextField("@id='txtFirstName']").SetText(FirstName)
            .DomTextField("@id='txtLastName']").SetText(LastName)
            .DomTextField("@id='txtPhoneNumber']").SetText(PhoneNumber)

            .DomButton("@id='btnAdd']").Select()
        End For
    End With
End Sub
```

```
Next
End With
End Sub
```

スクリプトに ActiveData ファイル内のデータの一部を含めるには、次のように入力します。

```
Public Sub Main()
    AddSomeToPhoneBook( 3, 5 )    'Reads rows 3, 4 and 5 only
End Sub

Public Sub AddSomeToPhoneBook( firstToAdd As Integer, lastToAdd As Integer )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData", firstToAdd,
lastToAdd )
    Dim row As ActiveDataRow

    With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
        For Each row In data
            Dim FirstName As String = row.GetString("fname")
            Dim LastName As String = row.GetString("lname")
            Dim PhoneNumber As String = row.GetString("number")

            .DomTextField("@id='txtFirstName']").SetText(FirstName)
            .DomTextField("@id='txtLastName']").SetText(LastName)
            .DomTextField("@id='txtPhoneNumber']").SetText(PhoneNumber)

            .DomButton("@id='btnAdd']").Select()
        Next
    End With
End Sub
```

スクリプトに ActiveData ファイル内のすべてのデータを読み取り専用として含めるには、次のように入力します。

```
Public Sub Main()
    AddAllToPhoneBookReadOnly()    'Opens the data file as read only
End Sub

Public Sub AddAllToPhoneBookReadOnly()
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData", 1, -1, 0,
True )
    Dim row As ActiveDataRow

    With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
        For Each row In data
            Dim FirstName As String = row.GetString("fname")
            Dim LastName As String = row.GetString("lname")
            Dim PhoneNumber As String = row.GetString("number")

            .DomTextField("@id='txtFirstName']").SetText(FirstName)
            .DomTextField("@id='txtLastName']").SetText(LastName)
            .DomTextField("@id='txtPhoneNumber']").SetText(PhoneNumber)

            .DomButton("@id='btnAdd']").Select()
        Next
    End With
End Sub
```

環境内で複数のプロジェクトを使用する場合、Common プロジェクト内にある、名前が一意でない ActiveData ファイルを挿入するには、プロジェクト修飾子を使用します。たとえば、以下のように入力します。


```
Public Sub Main()
    AddAllToPhoneBookReadOnly() 'Opens the data file as read only
End Sub

Public Sub AddAllToPhoneBookReadOnly()
    Dim data As ActiveData = Workbench.LoadActiveData( "Common.PhoneBookData", 1,
-1, 0, True )
    Dim row As ActiveDataRow

    With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
        For Each row In data
            Dim FirstName As String = row.GetString("fname")
            Dim LastName As String = row.GetString("lname")
            Dim PhoneNumber As String = row.GetString("number")

            .DomTextField("@id='txtFirstName']").SetText(FirstName)
            .DomTextField("@id='txtLastName']").SetText(LastName)
            .DomTextField("@id='txtPhoneNumber']").SetText(PhoneNumber)

            .DomButton("@id='btnAdd']").Select()
        Next
    End With
End Sub
```

 **注:** 任意のプロジェクト内の資産から、Common プロジェクト内の資産にアクセスできます。ただし、その逆はできません。現在処理中のプロジェクト内の ActiveData ファイルを使用する場合、プロジェクト修飾子を指定する必要はありません。別のプロジェクト内の Common プロジェクトにある ActiveData ファイルを使用する場合は、「Common」プロジェクト修飾子を指定します。

RunScript メソッド

クラス


[Workbench](#)。

処理


名前によって識別される子スクリプトを、パラメータを渡して実行します。

構文

```
Workbench.RunScript("scriptname", [parameters])
```

変数	説明
<code>scriptname</code>	実行する子スクリプトの名前。STRING。  注: 任意のプロジェクト内の資産から、Common プロジェクト内の資産にアクセスできます。ただし、その逆はできません。現在処理中のプロジェクト内のスクリプトを使用する場合、プロジェクト修飾子を指定する必要はありません。別のプロジェクト内の Common プロジェクトにあるスクリプトを使用する場合は、「Common」プロジェクト修飾子を指定します。

変数	説明
パラメータ	省略可能：子スクリプトからメイン スクリプトに渡すパラメーター。STRING または OBJECT。

 **注:** Oracle データベースに格納されるスクリプト名では、大文字と小文字が区別されます。Oracle データベースから挿入するスクリプトの名前は、データベースでの名前と正確に一致する必要があります。一致していないと、Silk Test Workbench でスクリプトの読み込みに失敗します。

使用例

環境内で複数のプロジェクトを使用する場合、Common プロジェクト内にある、名前が一意でないスクリプトを挿入するには、プロジェクト修飾子を使用します。たとえば、以下のように入力します。

```
Workbench.RunScript("Common.AddAccount")
```

ここで、*AddAccount* は Common プロジェクトおよび現在使用中のプロジェクト内にあるスクリプトです。ユーザーは現在のプロジェクトでなく、Common プロジェクト内のスクリプトを使用することができます。

Verify メソッド

クラス

[Workbench](#)。

処理

予想した値と実際の値が等しいかどうかを検証します。コメントを指定することもできます。

構文

このメソッドには多くのオーバーロードがあります。

```
result = Workbench.Verify(condition[, VerifyFlags])
```

または

```
result = Workbench.Verify(condition[, comment, VerifyFlags])
```

または

```
result = Workbench.Verify(expected, actual[, VerifyFlags])
```

または

```
result = Workbench.Verify(expected, actual[, VerifyFlags, comment])
```

変数	説明
<i>result</i>	検証が成功したかどうか。BOOLEAN。
<i>condition</i>	満たす必要がある条件。BOOLEAN。
<i>expected</i>	スクリプトから返されると予測される値。OBJECT。
<i>actual</i>	スクリプトから実際に返される値。OBJECT。
<i>comment</i>	省略可能：追加するコメント。STRING。
<i>VerifyFlags</i>	省略可能：検証の失敗時にスクリーンショットをキャプチャするかどうか。この列挙には、次の値を指定できます。 <ul style="list-style-type: none"> None

変数	説明
	<ul style="list-style-type: none"> ScreenShotOnFailure VerifyFlags。

使用例

期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object, comment As String)
```

たとえば、`Workbench.Verify("red", "green", "checking colors")` は検証に失敗し、メッセージ「checking colors - 実際: [green]; 予想: [red]」が表示されます。

期待値と実際の値を比較して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object, comment As String, verifyFlags As VerifyFlags)
```

たとえば、`Workbench.Verify("red", "green", "checking colors", verifyFlags.ScreenShotOnFailure)` は検証に失敗し、メッセージ「checking colors - 実際: [green]; 予想: [red]」が表示され、スクリーンショットが結果ファイルに追加されます。

予想結果によって返された値を検証して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean, comment As String)
```

たとえば、`Workbench.Verify(True, "Test Passed")` は検証に合格します。一方、`Workbench.Verify(False, "Test Failed")` は検証に失敗します。

予想結果によって返された値を検証して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean, comment As String, verifyFlags As VerifyFlags)
```

たとえば、`Workbench.Verify(True, "Test Passed", verifyFlags.ScreenShotOnFailure)` は検証に合格し、スクリーンショットは追加されません。一方、`Workbench.Verify(False, "Test Failed", ScreenShotOnFailure)` は検証に失敗し、スクリーンショットが結果ファイルに追加されます。

`IEnumerable` オブジェクト（リストや配列など）について実際の値と期待値を比較するには、以下のように入力します。


```
Workbench.Verify(expectedEnumerable, actualEnumerable)
```


例：

```
Dim selectedItemsList = listBox.SelectedItems ' we assume that a list with the items
"red" and "blue" is returned
Dim expectedItemsList = New List(Of String)()
expectedItemsList.Add("red")
expectedItemsList.Add("blue")
```

```
Workbench.Verify(selectedItemsList, expectedItemsList) ' verification passes
```

```
Dim expectedItemsArray = New String() { "red", "blue" }
Workbench.Verify(selectedItemsList, expectedItemsArray) ' verification passes
```

 **注:** 2つの `IEnumerable` オブジェクトは、要素の数が同じであり、それらの要素が等しく、かつ順序が同じである場合に、等しいとみなされます。

 **注:** 浮動小数点数を含む数学演算では、2つの数値が、ユーザーの視点では等しいにもかかわらず、内部表現のために完全には同一ではないとみなされる場合があります。このため、浮動小数点数（単精度、倍精度）は、差異が 0.00001 未満である場合は等しいとみなされます。この値は、状況によ

っては正しくない場合があります。その場合は、代わりに `Verify(result As Boolean)` を使用して 2 つの値を比較します。

VerifyAsset メソッド (Workbench)

クラス

[Workbench](#)。

処理

検証資産を実行します。検証する UI オブジェクトが見つからない場合は、`ObjectNotFoundException` をスローします。

構文

```
result = Verification.VerifyAsset(verificationAsset[, verifyFlags])
```

変数	説明
<code>result</code>	検証が成功したかどうか。BOOLEAN。
<code>verificationAsset</code>	実行する検証資産の名前。STRING。
<code>VerifyFlags</code>	省略可能：検証の失敗時にスクリーンショットをキャプチャするかどうか。この列挙には、次の値を指定できます。 <ul style="list-style-type: none">• None• ScreenShotOnFailure VerifyFlags。

使用例

イメージ資産 `myImageAsset` の検証を実行するには、次のように入力します。

```
Workbench.VerifyAsset("myImageAsset")
```

イメージ資産 `myImageAsset` の検証を実行して、検証の失敗時にスクリーンショットを結果ファイルに追加するには、次のように入力します。

```
Workbench.VerifyAsset("myImageAsset", verifyFlags.ScreenShotOnFailure)
```

ResultComment メソッド

クラス

[Workbench](#)。

処理

結果 ウィンドウの **詳細** タブに、コメントをステップとして追加します。結果コメントを使用して、テストに関する情報をテスト結果に追加できます。テストを再生すると、**結果** ウィンドウの **詳細** タブにコメントがステップとして表示されます。

構文

```
Workbench.ResultComment(comment)
```

変数	説明
<code>comment</code>	結果に追加されるコメント。STRING。


使用例

テストでいくつかの挿入された値のフォーマット検証が実行されなかったという結果を指摘する場合、次のような行をスクリプトに追加できます：

```
Workbench.ResultComment("Note: This test did not verify that the format of the inserted value was correct.")
```

データ型

このセクションでは、さまざまなデータ型を示します。

 **注:** .NET Framework のデータ型についての情報は、MSDN を参照してください（たとえば、Integer、Boolean、String など）。ビジュアル テストで使用することのできるデータ型についての情報は、「ビジュアル テストにおける変数」を参照してください。

ClickType 列挙

説明

TextClick の再生中にテキストをクリックする方法を指定します。

構文

Public Enumeration ClickType

メンバー

名前	説明
Left	左マウス ボタンを 1 回クリックします。
Right	右マウス ボタンを 1 回クリックします。
Middle	中央マウス ボタンを 1 回クリックします。
LeftDouble	左マウス ボタンをダブル クリックします。
Press	左マウス ボタンをクリックして、その状態を保持します。
Release	左マウス ボタンを離します。

Color クラス

説明

レッド、グリーン、ブルー、アルファ チャンネルの点から色について説明します。

継承階層

Color はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

Public Class Color

フィールド

名前	説明
<code>BLACK</code>	#FF000000 の ARGB 値を持つ色を取得します。
<code>WHITE</code>	#FFFFFFFF の ARGB 値を持つ色を取得します。
<code>GREEN</code>	#FF00FF00 の ARGB 値を持つ色を取得します。
<code>RED</code>	#FFFF0000 の ARGB 値を持つ色を取得します。
<code>BLUE</code>	#FF0000FF の ARGB 値を持つ色を取得します。
<code>YELLOW</code>	#FFFFFF00 の ARGB 値を持つ色を取得します。
<code>MAGENTA</code>	#FFFF00FF の ARGB 値を持つ色を取得します。
<code>CYAN</code>	#FF00FFFF の ARGB 値を持つ色を取得します。

プロパティ

名前	説明
<code>R</code>	色のレッド ARGB チャンネルの値を取得します。 Integer
<code>G</code>	色のグリーン ARGB チャンネルの値を取得します。 Integer
<code>B</code>	色のブルー ARGB チャンネルの値を取得します。 Integer
<code>A</code>	色のアルファ ARGB チャンネルの値を取得します。 Integer

ItemIdentifier データ型

説明

リストボックス、タブコントロール、コンボボックスなどのコントロール内の項目を指定します。項目は次のようにして指定することができます。

- テキスト (String)。
- インデックス (Integer)。
- テキストとインデックスの組み合わせ (String)。

文字列値は項目の名前を指定します。文字列には、ワイルドカード文字を含めることができます。また、文字列として項目のインデックスを "[n]" という形式で指定することもできます。ここで、n は項目のインデックス番号です。

例

次の例では、ListBox 内の項目を選択する方法を示します。

```
listbox.select(0) ' Selects the first item, which has the index 0.
listbox.select("[0]") ' Also selects the first item.
listbox.select("apple") ' Selects the first item with text "apple".
listbox.select("apple[1]") ' Selects the second item with the text "apple".
```

ItemPath データ型

説明

ツリーのルートから特定の項目への階層的パスです。常に階層全体を指定する必要があります。項目パスの階層の各部分は、名前、インデックス、または両方の組み合わせで指定できます。複数の属性を指定することによって、階層パス内の各項目を一意に指定することができます。

ツリー内のパスは次の属性を持ちます。

- 項目と他の項目間の階層的関係。
- 項目のタイトルまたはキャプション。
- 同レベルの項目に関して項目の順序を表す 0 ベースの数。

例

次の例では、パス、インデックス、またはパスとインデックスの組み合わせによって、ツリー内の項目を一意に指定する方法を示します。

指定方法	項目
"/Desktop/ Applications/ Games"	ツリーの外部レベルにある一意な項目 "Desktop" の下の、一意な項目 "Applications"、さらにその下の一意な項目 "Games"。
"/ Desktop/[2]/ [4]"	ツリーの外部レベルにある一意な項目 "Desktop" の下の、3 番目の項目の下の 5 番目の項目。
"/Desktop/ Applications[2]/Games"	ツリーの外部レベルにある一意な項目 "Desktop" の下の、テキスト "Applications" を持つ 3 番目の項目、さらにその下の一意な項目 "Games"。

ModifierKeys 列挙

説明

再生中に押下するキー修飾子を指定します。

構文

Public Enumeration ModifierKeys

メンバー

名前	説明
None	キー修飾子はありません。
Shift	Shift 修飾子。
Alt	Alt 修飾子。
コントロール	Ctrl 修飾子。
ControlAlt	Ctrl+Alt 修飾子。
ShiftControl	Shift+Ctrl 修飾子。

名前	説明
ShiftAlt	Shift+Alt 修飾子。
ShiftControlAlt	Ctrl+Shift+Alt 修飾子。

MouseButton 列挙

説明

マウス デバイスのボタンを指定する値を定義します。

構文

Public Enumeration MouseButton

メンバー

名前	説明
Left	左マウス ボタン。 Left = 1。
Middle	中央マウス ボタン。 Middle = 3。
Right	右マウス ボタン。 Right = 2。

Point クラス

説明

画面上のポイントを指定します。

継承階層

Point はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

Public Class Point

プロパティ

名前	説明
X	X 座標を取得します。
Y	Y 座標を取得します。

Range クラス

説明

最小値と最大値を持つ範囲を指定します。

継承階層

Range はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

Public Class Range

プロパティ

名前	説明
<i>Minimum</i>	最小値を取得します。
<i>Maximum</i>	最大値を取得します。

Rectangle クラス

説明

画面上の矩形領域を指定します。

継承階層

Rectangle はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

Public Class Rectangle

プロパティ

名前	説明
<i>X</i>	矩形領域の X 座標を取得します。
<i>Y</i>	矩形領域の Y 座標を取得します。
<i>Width</i>	矩形領域の幅を取得します。
<i>Height</i>	矩形領域の高さを取得します。
<i>Center</i>	矩形領域の中央を取得します。
<i>IsEmpty</i>	矩形が空かどうかを判定します。

メソッド

名前	説明
<i>Contains(Rectangle)</i>	現在の Rectangle が、指定した Rectangle を含むかどうかを調べます。 Boolean
<i>Contains(Point)</i>	現在の Rectangle が、指定した Point を含むかどうかを調べます。 Boolean
<i>Contains(X,Y)</i>	現在の Rectangle が、指定した座標を含むかどうかを調べます。 Boolean

TextPosition クラス

説明

テキスト フィールド内の位置を指定します。

継承階層

TextPosition はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
Public Class TextPosition
```

プロパティ

名前	説明
Line	行を取得します。
Column	列を取得します。

TextRange クラス

説明

テキスト フィールド内の範囲を指定します。

継承階層

TextRange はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
Public Class TextRange
```

プロパティ

名前	説明
StartLine	範囲の最初の行を取得します。
StartColumn	範囲の最初の列を取得します。
EndLine	範囲の最後の行を取得します。
EndColumn	範囲の最後の列を取得します。

TreeContent クラス

説明

ツリー コントロールの内容を表します。 ツリー内のすべての項目が `TreeNode` として表現されます。

継承階層

TreeContent はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
Public Class TreeContent
```

プロパティ

名前	説明
RootNodes	対応するツリーのルート子ノードを取得します。

TreeNode クラス

説明

TreeContent 内のノードを表します。

継承階層

TreeNode はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
Public Class TreeNode
```

プロパティ

名前	説明
Children	TreeNode の子ノードを取得します。
Text	TreeNode のテキストを取得します。

Flex クラス リファレンス

Flex コントロールのテストに利用可能なクラスを一覧します。さらに、カスタム コントロールをテストするために Silk Test をカスタマイズすることもできます。

クラスの詳細については、『*Apache Flex 開発者ガイド*』を参照してください。



注: Flex および Flex クラス使用時の既知の問題は、Silk Test でも再現します。たとえば、FlexChart クラスを使用した要素のドラッグ&ドロップは、要素のドラッグを開始する前にグラフ要素の端近くをクリックすると機能しない場合があります。これによって、ドラッグの開始座標が無効になるため、再生に失敗します。これは Flex のドラッグ&ドロップの既知の問題です。Flex の既知の問題の詳細については、『*Apache Flex リリース ノート*』を参照してください。



注: FlexTitleWindow クラスは、Silk Test によって FlexPanel として識別され記録されます（別のクラスとして定義されていますが）。これは Flex の既知の問題です。Flex の既知の問題の詳細については、『*Apache Flex リリース ノート*』を参照してください。

FlexAccordion クラス

説明

Accordion ナビゲーター コンテナーは子コンテナーのコレクションを持ちますが、その 1 つだけを一度に可視にすることができます。子の間を移動するために使用するナビゲーター ボタン(アコーディオン ヘッダー) を作成および管理します。

子のコンテナーそれぞれに 1 つのナビゲーター ボタンが関連付けられており、それぞれのナビゲーター ボタンは、子ではなく Accordion コンテナーに属しています。ナビゲーター ボタンをクリックすると、関連付けられた子のコンテナーが表示されます。新しい子への切り替えは、ある子が非表示になり、別の子が表示されたことをユーザーにわかりやすくするためのアニメーションが使用されます。

Accordion コンテナは ViewStack コンテナを形容しませんが、selectedIndex や selectedChild などの ViewStack コンテナのすべてのプロパティ、メソッド、スタイル、およびイベントを実装します。

継承階層

- [FlexContainer](#)
 - FlexAccordion

構文

```
'Declaration
Public Class FlexAccordion _
Inherits FlexContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)

名前	説明
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (<i>FlexContainer</i> から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>headerHeight</i>	Accordion ヘッダーの高さ (ピクセル単位)。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalGap</i>	子同士の水平方向の間隔を示すピクセル数。デフォルト値は 8 です。
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)

名前	説明
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。(FlexContainer から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>paddingBottom</i>	コンテナの下境界線とコンテンツ領域との間にあるピクセル数。
<i>paddingTop</i>	コンテナの上境界線とコンテンツ領域との間にあるピクセル数。
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>selectedChild</i>	現在表示されている子コンテナへの参照。
<i>selectedIndex</i>	現在表示されている子コンテナの 0 を基準としたインデックス。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>textSelectedColor</i>	選択したテキストの色。デフォルト値は 0x2B333C です。
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
<i>verticalGap</i>	子同士の垂直方向の間隔を示すピクセル数。
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)

名前	説明
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Change	選択した子コンテナーが変更されたときにディスパッチされます。
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。 (FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexAdvancedDataGrid クラス

説明

標準の DataGrid コントロールの機能を拡張したもので、Apache Flex アプリケーションにデータ視覚化機能が追加されます。これらの機能によって、データ表示やデータ集計、データフォーマットについて、より高度な操作が可能になります。

AdvancedDataGrid コントロールは、List コントロールに似ていますが、複数の列のデータを表示でき、複数のプロパティを持つオブジェクトの表示に適している点が異なります。

継承階層

- [FlexListBase](#)
 - FlexAdvancedDataGrid
 - [FlexOLAPDataGrid](#)

構文

```
'Declaration  
Public Class FlexAdvancedDataGrid _  
Inherits FlexListBase
```

プロパティ

名前	説明
allowDragSelection	ドラッグによる選択が有効かどうか。通常の選択では、選択する項目上にマウスを移動させてもマウス ボタンを押すことはできませんが、ドラッグによる選択では、項目にドラッグすることによって、その項目を選択できます。
allowMultipleSelection	1 つ以上の項目を同時に選択することが許可されているかどうか。 (FlexListBase から継承されました)
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
alternatingItemColors	リストの項目の背景に使用する色。 (FlexListBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。 (FlexListBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnCount	TileList コントロールまたは HorizontalList コントロールの項目に表示される列の数。 (FlexListBase から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。 (FlexListBase から継承されました)
columnWidth	コントロールの列の幅。 (FlexListBase から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
dataTipField	データ ヒントとして表示するデータ プロバイダー項目内のフィールドの名前。 (FlexListBase から継承されました)
depthColors	AdvancedDataGrid コントロールのナビゲーション ツリーの各レベルの行で使用する色の配列 (降順)。デフォルト値は未定義です。
disabledColor	コンポーネントが無効な場合のテキストの色。
displayItemsExpanded	ナビゲーション ツリーが展開されてすべての項目が表示されているかどうか。
editable	データ プロバイダーの項目を編集できるかどうか。

名前	説明
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>focusAlpha</i>	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。 (<i>FlexListBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>headerColors</i>	ヘッダーの背景にグラデーションを描画するために使用する 2 色の配列。
<i>headerHeight</i>	列のヘッダー セルの高さ (ピクセル単位)。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalGridLineColor</i>	水平グリッド線の色。
<i>iconField</i>	アイコンとして表示するものを決定する、データ プロバイダー オブジェクト内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>indentation</i>	ナビゲーション ツリーの各ノードのインデント (ピクセル単位)。デフォルト値は 17 です。
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>lockedColumnCount</i>	スクロールするコントロールの最初の列のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>lockedRowCount</i>	スクロールするコントロールの最初の行のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>minColumnWidth</i>	列の最小幅 (ピクセル単位)。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexListBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexListBase から継承されました)
openDuration	ナビゲーション ツリーの開いた状態や閉じた状態への遷移にかかる時間の長さ (ミリ秒単位)。デフォルト値は 250 です。
paddingBottom	行の下端と行内のレンダラの下端との間のピクセル数。
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。
paddingTop	行の上端と行内のレンダラの上端との間のピクセル数。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizableColumns	列のサイズをユーザーが変更できるかどうか。
rollOverColor	ユーザーがレンダラの上にマウスを置いたときのレンダラの背景色。デフォルト値は 0xEEFEE6 です。(FlexListBase から継承されました)
rowCount	表示する行の数。(FlexListBase から継承されました)
rowHeight	行の高さ (ピクセル単位)。(FlexListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectable	選択された項目を選択状態でリストに表示するかどうか。 (FlexListBase から継承されました)
selectedCells	行インデックスおよび列インデックスとしてセル位置の配列。
selectedIndex	データ プロバイダー内で選択された項目のインデックス。 (FlexListBase から継承されました)
selectedIndices	データ プロバイダーで選択された項目のインデックスの配列。 (FlexListBase から継承されました)
selectedItem	データ プロバイダー内で選択された項目への参照。(FlexListBase から継承されました)

名前	説明
<i>selectedItems</i>	データプロバイダーで選択された項目への参照の配列。 (<i>FlexListBase</i> から継承されました)
<i>selectionColor</i>	ユーザーが選択したときのレンダラの背景色。デフォルト値は 0x7FCEFF です。 (<i>FlexListBase</i> から継承されました)
<i>selectionDisabledColor</i>	コンポーネントが無効化されたときのレンダラの背景色。デフォルト値は 0xDDDDDD です。 (<i>FlexListBase</i> から継承されました)
<i>selectionMode</i>	コントロールの選択モード。
<i>showDataTips</i>	行内のテキストに対して dataTips を表示するかどうか。 (<i>FlexListBase</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>textRollOverColor</i>	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。 (<i>FlexListBase</i> から継承されました)
<i>textSelectedColor</i>	ユーザーが選択したときのレンダラのテキスト色。デフォルト値は 0x2B333C です。 (<i>FlexListBase</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (<i>FlexObject</i> から継承されました)
<i>useRollOver</i>	項目の上にマウスを置いたときに、その項目をハイライト表示するかどうか。 (<i>FlexListBase</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>variableRowHeight</i>	個々の行が別々の高さを持つことができるかどうか。 (<i>FlexListBase</i> から継承されました)
<i>verticalAlign</i>	行のレンダラの垂直方向の配置。 (<i>FlexListBase</i> から継承されました)
<i>verticalGridLineColor</i>	垂直グリッド線の色。デフォルト値は 0x666666 です。
<i>visible</i>	表示オブジェクトが可視かどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
wordWrap	行内のテキストを折り返すかどうか。(FlexListBase から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	AdvancedDataGrid の分岐を閉じるまたは折りたたみます。
ColumnGroupedADGHeaderShift	ユーザーが列ヘッダーを新しい場所にドラッグし、マウス ボタンを離れたときにディスパッチされます。その結果、その列は新しいインデックスに移されます。
ColumnStretch	ユーザーが列の幅を変更したときにディスパッチされます。これは、列に表示されるデータ量を変更されたことを意味します。
Deselect	以前に選択した項目の選択を解除するときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClick	コントロールの可視項目上でマウスがダブルクリックされたことを示す itemDoubleClick イベントにおける ListEvent オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)

名前	説明
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetCellData	グリッド内のセルのデータを返します。
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetGroupedItemChildrenCount	グループの最初の項目内にある子の数を返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetRowData	データ プロバイダーの項目の行のデータを返します。
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexListBase から継承されました)
HeaderClick	列の内容に基づいてグリッドの内容の並べ替えを行うために、ユーザーが列ヘッダーでマウス ボタンを離れたときにディスパッチされます。

名前	説明
HeaderShift	ユーザーが列ヘッダーを新しい場所にドラッグし、マウス ボタンを離れたときにディスパッチされます。その結果、その列は新しいインデックスに移されます。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsGroupedItem	項目がグループのメンバーかどうかを指定する値を返します。
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
MultiSelect	複数項目を選択するアクションの一部として項目が選択されたときにディスパッチされるイベントにおけるイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
Open	AdvancedDataGrid の分岐を開くまたは展開します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
ScrollToIndex	指定したインデックス位置にあるデータ プロバイダー項目を表示状態にします。(FlexListBase から継承されました)
Select	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
SelectIndex	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
WaitForObject	<p>ウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexAlert クラス

説明

メッセージ、タイトル、ボタン（「OK」、「キャンセル」、「はい」、「いいえ」の任意の組み合わせ）およびアイコンを含むことができるポップアップ ダイアログ ボックスです。

Alert コントロールはモーダルです。これは、ユーザーが閉じるまでフォーカスを保持することを意味します。

継承階層

- [FlexPanel](#)
 - FlexAlert

構文

```
'Declaration
Public Class FlexAlert _
Inherits FlexPanel
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。 (<i>FlexContainer</i> から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。 (<i>FlexContainer</i> から継承されました)
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。 (<i>FlexContainer</i> から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。 (<i>FlexContainer</i> から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。 (<i>FlexContainer</i> から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>FlexContainer</i> から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。 (<i>FlexContainer</i> から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。 (<i>FlexContainer</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>defaultButtonFlag</i>	デフォルト ボタンを指定する Alert.OK、Alert.CANCEL、Alert.YES または Alert.NO を含むビット フラグ。デフォルト値は Alert.OK です。
<i>dropShadowEnabled</i>	Panel コンテナのドロップ シャドウが可視かどうか。デフォルト値は true です。 (<i>FlexPanel</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (<i>FlexContainer</i> から継承されました)

名前	説明
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>footerColors</i>	フッター (ControlBar コンテナの領域) の背景の描画に使用される 2色の配列。 (<i>FlexPanel</i> から継承されました)
<i>headerColors</i>	ヘッダーを描画するために使用する 2色の配列。 (<i>FlexPanel</i> から継承されました)
<i>headerHeight</i>	ヘッダーの高さ。デフォルト値はタイトル テキストのスタイルに基づいて計算されます。 (<i>FlexPanel</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)

名前	説明
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
status	タイトル バーのステータス領域に表示されるテキスト。デフォルト値は "" です。 (FlexPanel から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
text	この警告ダイアログ ボックスに表示するテキスト。デフォルト値は "" です。
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
title	タイトル バーに表示されるタイトルまたはキャプション。デフォルト値は "" です。 (FlexPanel から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値（たとえば、テキスト コントロールのテキスト）。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexApplication クラス

説明

デフォルトのコンテナ (Application コンテナ) が定義されるため、別のコンテナを明示的に定義することなく、アプリケーションにコンテンツを追加することができます。

継承階層

- [FlexContainer](#)
 - FlexApplication
 - [FlexWindowedApplication](#)

構文

```
'Declaration
Public Class FlexApplication _
Inherits FlexContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)

名前	説明
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。 (<i>FlexContainer</i> から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。 (<i>FlexContainer</i> から継承されました)
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。 (<i>FlexContainer</i> から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。 (<i>FlexContainer</i> から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。 (<i>FlexContainer</i> から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>FlexContainer</i> から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。 (<i>FlexContainer</i> から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。 (<i>FlexContainer</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (<i>FlexContainer</i> から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)

名前	説明
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
url	このアプリケーションの SWF ファイルがロードされた URL。
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)

名前	説明
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexAreaChart クラス

説明

データ内の値をつなぐ線で囲まれた面としてデータを表します。AreaChart コントロールを使用すると、単純面グラフ、積み上げ面グラフ、構成比面グラフ、高低面グラフなど、各種の面グラフを表すことができます。

継承階層

- [FlexCartesianChart](#)
 - FlexAreaChart

構文

```
'Declaration
Public Class FlexAreaChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipContent	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dataTipMode	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
description	グラフのデータの簡単な説明。(FlexChart から継承されました)
dragEnabled	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
dropEnabled	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)

名前	説明
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。 (FlexChart から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
<code>fontFamily</code>	使用するフォントの名前。 (FlexChart から継承されました)
<code>fontSize</code>	テキストの高さ (ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。 (FlexChart から継承されました)
<code>gutterBottom</code>	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
<code>gutterLeft</code>	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
<code>gutterRight</code>	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
<code>gutterTop</code>	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
<code>horizontalAxisRatio</code>	水平軸の高さの制限。 (FlexCartesianChart から継承されました)
<code>id</code>	コンポーネントの ID。 (FlexObject から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。 (FlexChart から継承されました)
<code>maximumDataTipCount</code>	グラフが表示するデータ ヒントの最大数。 (FlexChart から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)

名前	説明
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。 (FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。 (FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。 (FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データ ポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)

名前	説明
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。 (FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。 (FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexAreaSeries クラス

説明

AreaChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexAreaSeries

構文

```
'Declaration  
Public Class FlexAreaSeries _  
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。 (FlexChartSeries から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。 (FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
form	領域の境界のタイプ。

名前	説明
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (FlexChartSeries から継承されました)
minField	領域の下部の境界を決定する dataProvider のフィールド。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。 (FlexChartSeries から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
radius	データ ポイントのグラフ要素の半径 (ピクセル単位)。
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。 (FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。 (FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。 (FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。 (FlexChartSeries から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)

名前	説明
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
xField	水平軸上のデータ ポイントの位置を決定するデータ プロバイダーのフィールド。
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)
yField	垂直軸上のデータ ポイントの位置を決定するデータ プロバイダーのフィールド。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、 Apache Flex SDK ドキュメント を参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインターを動かしたインデックスです。(FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexAxisRenderer クラス

説明

グラフの水平軸および垂直軸を表すために使用します。軸は、その長さに沿って、ラベル、目盛り、およびタイトルを表示する役割を持ちます。

AxisRenderer オブジェクトは、それを囲んでいるグラフ オブジェクトからビジュアル プロパティの一部を継承します。ラベルおよびタイトルのテキスト フォーマットには、レンダラの CSS テキストのプロパティがデフォルトとして設定されます。これとは別に、軸のタイトルのフォーマットは、AxisRenderer または囲んでいるグラフで `axisTitleStyleName` を指定することで制御できます。

継承階層

- [FlexObject](#)
 - FlexAxisRenderer

構文

```
'Declaration  
Public Class FlexAxisRenderer _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
canDropLabels	AxisRenderer が正しくレイアウトすることが必要な場合にラベルを落とすかどうか。
canStagger	2つのラベル行でラベルを交互に配置するかどうか。この設定を使用すると、ラベルに必要な領域が最小化されます。デフォルト値は true で、ラベルは交互に配置されます。
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>heightLimit</code>	軸レンダラがグラフから取得するスペースの最大値 (ピクセル単位)。
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>labelGap</code>	目盛りの末端からラベルの上端までの間隔の幅 (ピクセル単位)。 デフォルト値は 3 です。
<code>labelRotation</code>	ラベルの回転。
<code>lastVisibleRow</code>	最後の可視の子のインデックス。
<code>length</code>	軸の長さ (スクリーン座標)。
<code>minorTickLength</code>	軸上の小さな目盛りの長さ (ピクセル単位)。 デフォルト値は 0 です。
<code>minorTickPlacement</code>	小さな目盛りを描画する場所を示す値。 有効な値は、 <i>inside</i> 、 <i>outside</i> 、 <i>cross</i> 、 および <i>none</i> です。
<code>minorTicks</code>	軸に沿った小さな目盛りの描画位置を指定する配列。 各配列要素には、 0 から 1 までの値が含まれています。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は <i>true</i> であり、これは表示リスト上の <i>InteractiveObject</i> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <i>false</i> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。
<code>numRows</code>	利用可能なデータ中の行の合計数。
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は <i>NaN</i> です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは <i>NaN</i> にリセットされます。 (<i>FlexObject</i> から継承されました)
<code>placement</code>	グラフで <code>axisRenderer</code> が表示される側。

名前	説明
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
showLabels	ラベルを軸に沿って表示するかどうか。デフォルト値は true です。
showLine	軸を表示するかどうか。デフォルト値は true です。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TextObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
tickLength	軸上の目盛りの長さ (ピクセル単位)。デフォルト値は 3 です。
tickPlacement	目盛りを描画する場所を示す値。有効な値は、 <i>inside</i> 、 <i>outside</i> 、 <i>cross</i> 、および <i>none</i> です。
ticks	軸に沿った目盛りの描画位置を指定する配列。各配列要素には、0 から 1 までの値が含まれています。
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。 <code>useHandCursor</code> が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TextObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 ToString 	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexBarChart クラス

説明

データ内の値によって長さが決まる一連の横棒でデータを表します。BarChart コントロールは、単純横棒グラフ、集合横棒グラフ、積み上げ横棒グラフ、構成比横棒グラフ、高低横棒グラフなど、各種の横棒グラフを表すことができます。

継承階層

- [FlexCartesianChart](#)
 - FlexBarChart

構文

```
'Declaration
Public Class FlexBarChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
barWidthRatio	棒グラフを描く際のカテゴリ幅に相対の広さ (0 から 1 の範囲の割合)。
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipContent	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には <code>false</code> を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)

名前	説明
dataTipMode	グラフの DataTip コントロールの表示方法。 (FlexChart から継承されました)
description	グラフのデータの簡単な説明。 (FlexChart から継承されました)
dragEnabled	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。 (FlexChart から継承されました)
dropEnabled	ドラッグした項目をグラフ上にドロップできるかどうか。 (FlexChart から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。 (FlexChart から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
fontFamily	使用するフォントの名前。 (FlexChart から継承されました)
fontSize	テキストの高さ (ピクセル単位)。 ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。 ColorPicker コントロールのデフォルト値は 11 です。 (FlexChart から継承されました)
gutterBottom	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
gutterLeft	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
gutterRight	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
gutterTop	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。 (FlexCartesianChart から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
horizontalAxisRatio	水平軸の高さの制限。 (FlexCartesianChart から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (FlexChart から継承されました)
maxBarWidth	描画する横棒の幅 (ピクセル単位)。
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。 (FlexChart から継承されました)

名前	説明
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)

名前	説明
showDataTips	グラフの DataTip コントロールが表示されるかどうか。 (FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。 (FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。 (FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。 (FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル)

名前	説明
	はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexChart から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexChart から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 (FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexChart から継承されました)

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexBarSeries クラス

説明

BarChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexBarSeries

構文

```
'Declaration
Public Class FlexBarSeries _
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
barWidthRatio	カテゴリ幅に相対的な棒グラフをレンダリングするための幅。
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(<i>FlexObject</i> から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。(<i>FlexChartSeries</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。(<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。(<i>FlexChartSeries</i> から継承されました)
<code>maxBarWidth</code>	横棒の幅 (ピクセル単位)。
<code>minField</code>	各横棒の下部を決定する データ プロバイダーのフィールド。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。(<i>FlexObject</i> から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。(<i>FlexChartSeries</i> から継承されました)
<code>numRows</code>	利用可能なデータ中の行の合計数。(<i>FlexChartSeries</i> から継承されました)
<code>offset</code>	使用可能な領域の中心から横棒の中心をずらす度合い (カテゴリの幅に対する割合)。
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<code>selectedIndex</code>	データ プロバイダーのデータ内で選択された項目のインデックス。(<i>FlexChartSeries</i> から継承されました)

名前	説明
<i>selectedIndices</i>	データプロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
<i>selectedItem</i>	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
<i>selectedItems</i>	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>xField</i>	各横棒の先端の x 軸の位置を決定するデータプロバイダーのフィールド。
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)
<i>yField</i>	グラフ内の各横棒の終端の y 軸の位置を決定するデータプロバイダーのフィールド。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)

名前	説明
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。(FlexChartSeries から継承されました)

名前	説明
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウスポインターを動かしたインデックスです。(FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキストフィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexBox クラス

説明

Box コンテナでは、子が単一の垂直列または水平行にレイアウトされます。

継承階層

- [FlexContainer](#)
 - FlexBox
 - [FlexDividedBox](#)
 - [FlexNavigationBar](#)

構文

```
'Declaration
Public Class FlexBox _
Inherits FlexContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)

名前	説明
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>direction</i>	垂直レイアウト (デフォルト)、水平レイアウトのどちらを使用するか。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexContainer から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexContainer から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexContainer から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>horizontalAlign</i>	このコンテナ内の子の水平方向の配置。

名前	説明
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。(FlexContainer から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
<i>verticalAlign</i>	このコンテナ内の子の垂直方向の配置。
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(FlexObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>DragCancel</i>	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
<i>DragDrop</i>	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
<i>DragStart</i>	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>GetValues</i>	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(<i>FlexContainer</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。 (FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

FlexBubbleChart クラス

説明

各データ ポイントにつき 3 つの値でデータを表します。 各データ ポイントは、水平軸に沿った位置を決定する値、垂直軸に沿った位置を決定する値、およびグラフ上の他のデータ ポイントとの比較でグラフ要素のサイズを決定する値で定義されます。

継承階層

- [FlexCartesianChart](#)

- FlexBubbleChart

構文

```
'Declaration
Public Class FlexBubbleChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>clipContent</i>	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipMode</i>	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
<i>description</i>	グラフのデータの簡単な説明。(FlexChart から継承されました)
<i>dragEnabled</i>	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
<i>dropEnabled</i>	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChart から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)

名前	説明
fontFamily	使用するフォントの名前。(FlexChart から継承されました)
fontSize	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。(FlexChart から継承されました)
gutterBottom	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterLeft	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterRight	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterTop	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
horizontalAxisRatio	水平軸の高さの制限。(FlexCartesianChart から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChart から継承されました)
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。(FlexChart から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)

名前	説明
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。(FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。(FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>GetValues</i>	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(<i>FlexChart</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (<i>FlexObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (<i>FlexChart</i> から継承されました)
<i>SetFocus</i>	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (<i>FlexObject</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ToString</i>	コントロールの文字列表現を返します。 (<i>FlexDisplayObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexBubbleSeries クラス

説明

BubbleChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexBubbleSeries

構文

```
'Declaration  
Public Class FlexBubbleSeries _  
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChartSeries から継承されました)
maxRadius	この系列にレンダリングされる最大の項目の半径。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべて

名前	説明
	の子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(<i>FlexChartSeries</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(<i>FlexChartSeries</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<i>radiusField</i>	グラフ内の他のデータ ポイントを基準にして各シンボルの半径を決定するデータ プロバイダーのフィールド。
<i>scaleX</i>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>selectedIndex</i>	データ プロバイダーのデータ内で選択された項目のインデックス。(<i>FlexChartSeries</i> から継承されました)
<i>selectedIndices</i>	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(<i>FlexChartSeries</i> から継承されました)
<i>selectedItem</i>	系列内で選択されたグラフ項目。(<i>FlexChartSeries</i> から継承されました)
<i>selectedItems</i>	系列内で選択されたグラフ項目の配列。(<i>FlexChartSeries</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は <code>Nothing</code> です。(<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は <code>true</code> です。 <code>useHandCursor</code> が <code>false</code> の

名前	説明
	場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
xField	各データ ポイントの x 軸の位置を決定するデータ プロバイダーのフィールド。
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)
yField	各データ ポイントの y 軸の位置を決定するデータ プロバイダーのフィールド。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインタを動かしたインデックスです。 (FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexButton クラス

説明

Button コントロールはきわめて使用頻度の高い四角形のボタンです。 Button コントロールは押し下げ可能に見えます。 表面には、テキスト ラベル、アイコンのいずれか、または両方を表示できます。

通常、ボタンはイベントリスナを使用して、ユーザーがコントロールを選択したときのアクションを実行します。 Button コントロール上でマウスをクリックしたときに Button コントロールが有効である場合は、click イベントと buttonDown イベントがディスパッチされます。 ボタンは、有効か無効かにかかわらず、mouseMove、mouseover、mouseout、rollover、rollout、mousedown、mouseup などのイベントを必ずディスパッチします。

Button コントロールは、外観のカスタマイズや、プッシュ ボタンからトグル ボタンへの機能変更などが可能です。 ボタンの各状態に対応するスキンを使用することによって、ボタンの外観を変更できます。

Button コントロールのラベルには太字の書体を使用されます。Button コントロールのラベルに使用するフォントを埋め込む場合は、太字の書体を埋め込む必要があります。

継承階層

- [FlexObject](#)
 - FlexButton
 - [FlexCheckBox](#)
 - [FlexPopUpButton](#)
 - [FlexRadioButton](#)

構文

```
'Declaration  
Public Class FlexButton _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
disabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コントロールの背景の塗りつぶしに使用するアルファ値。
fillColors	コントロールの背景を濃淡色化するのに使用する色。
focusAlpha	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。

名前	説明
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。
<i>fontSize</i>	テキストの高さ (ピクセル単位)。
<i>fontStyle</i>	テキストがイタリック フォントかどうか。
<i>fontWeight</i>	テキストがボールド体かどうか。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	Button コントロールに表示されるテキスト。
<i>labelPlacement</i>	指定されたアイコンを基準にしたラベルの向き。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selected</i>	トグルボタンがオン (true) またはオフ (false) のどちらであるか。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>textAlign</i>	コンテナ内のテキストの配置。

名前	説明
textDecoration	テキストに下線を表示するかどうか。
textIndent	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。
textRollOverColor	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。
textSelectedColor	ユーザーがラベルを選択したときのラベルのテキストの色。デフォルト値は 0x000000 です。
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toggle	Button を toggle 状態にするかどうか。
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexButtonBar クラス

説明

共通の外観と操作性を持つ論理的に関連するプッシュ ボタンの水平または垂直方向のグループを定義します。

継承階層

- [FlexNavigationBar](#)
 - FlexButtonBar
 - [FlexToggleButtonBar](#)

構文

```
'Declaration  
Public Class FlexButtonBar _  
Inherits FlexNavigationBar
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
direction	垂直レイアウト (デフォルト)、水平レイアウトのどちらを使用するか。(FlexBox から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (<i>FlexContainer</i> から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalAlign</i>	このコンテナ内の子の水平方向の配置。 (<i>FlexBox</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)

名前	説明
numRows	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	アクティブなナビゲーション項目のインデックス。最初の項目のインデックスは 0 です。デフォルト値は -1 です。(FlexNavigationBar から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAlign	このコンテナ内の子の垂直方向の配置。(FlexBox から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)

名前	説明
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (<code>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps</code>)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Change	<code>itemClick</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexNavigationBar から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)

名前	説明
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexNavigationBar から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexCandlestickChart クラス

説明

データ系列の高値、安値、始値、終値などの財務データを一連のローソク足で示します。

各ローソク足の垂直線の上部和下部は、データポイントの高値と安値を示しています。一方、塗りつぶされたボックスの上部和下部は始値と終値を示しています。ローソク足の塗りつぶされ方は、データポイントの終値が始値よりも高いかどうかによって異なります。

継承階層

- [FlexCartesianChart](#)
 - FlexCandlestickChart

構文

```
'Declaration
Public Class FlexCandlestickChart _
Inherits FlexCartesianChart
```


プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>clipContent</i>	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
<i>columnWidthRatio</i>	ロウソク足グラフを描く際の水平軸のカテゴリ幅に相対の広さ (0 から 1 の範囲の割合)。
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipMode</i>	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
<i>description</i>	グラフのデータの簡単な説明。(FlexChart から継承されました)
<i>dragEnabled</i>	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
<i>dropEnabled</i>	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChart から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexChart から継承されました)
<i>fontSize</i>	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コント

名前	説明
	ロールのデフォルト値は 11 です。(FlexChart から継承されました)
<code>gutterBottom</code>	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
<code>gutterLeft</code>	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
<code>gutterRight</code>	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
<code>gutterTop</code>	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<code>horizontalAxisRatio</code>	水平軸の高さの制限。(FlexCartesianChart から継承されました)
<code>id</code>	コンポーネントの ID。(FlexObject から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。(FlexChart から継承されました)
<code>maxColumnWidth</code>	描画するロウソク足の幅 (ピクセル単位)。
<code>maximumDataTipCount</code>	グラフが表示するデータ ヒントの最大数。(FlexChart から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<code>mouseSensitivity</code>	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
<code>numRows</code>	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
<code>paddingBottom</code>	グラフの下境界線とコンテンツ領域とにあるピクセル数。(FlexChart から継承されました)

名前	説明
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。(FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。(FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>GetValues</i>	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(<i>FlexChart</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexCandlestickSeries クラス

説明

データ系列の高値、安値、始値、終値などの財務データを一連のローソク足で示します。

各ローソク足の垂直線の上部和下部は、データ ポイントの高値と安値を示しています。一方、塗りつぶされたボックスの上部和下部は始値と終値を示しています。 ローソク足の塗りつぶされ方は、データ ポイントの終値が始値よりも高いかどうかによって異なります。

継承階層

- [FlexHLOCSeriesBase](#)

- FlexCandlestickSeries

構文

```
'Declaration
Public Class FlexCandlestickSeries _
Inherits FlexHLOCSeriesBase
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
<i>columnWidthRatio</i>	カテゴリ幅に相対的な要素の幅。(FlexHLOCSeriesBase から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChartSeries から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexChartSeries から継承されました)
<i>maxColumnWidth</i>	要素の幅 (ピクセル単位)。(FlexHLOCSeriesBase から継承されました)

名前	説明
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChartSeries から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。(FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)

名前	説明
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
xField	要素の x 軸の位置を決定するデータプロバイダーのフィールド。(FlexHLOCSeriesBase から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウスボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインタを動かしたインデックスです。 (FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

FlexCanvas クラス

説明

Canvas レイアウト コンテナは、子コンテナおよびコントロールを配置する矩形領域を定義します。このコンテナは、その内部における子の位置を、各子の x および y プロパティを使用して明示的に指定できる唯一のコンテナです。

Canvas レイアウト コンテナの子は、それぞれの推奨されている幅および高さ設定されます。子の width プロパティを固定ピクセル値またはコンテナ サイズのパーセント値に設定することにより、その子の推奨幅の値をオーバーライドできます。推奨高さも同様の方法で設定できます。

Canvas コンテナ内でパーセントによるサイズ指定を使用している場合、コンポーネントの一部が重なり合う場合があります。これが期待する状況でない場合は、コンポーネントの位置とサイズを慎重に計画してください。

継承階層

- [FlexContainer](#)
 - FlexCanvas

構文

```
'Declaration  
Public Class FlexCanvas _  
Inherits FlexContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)

名前	説明
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (<i>FlexContainer</i> から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナーを表すために、一部のナビゲーター コンテナーによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナーに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナーに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)

名前	説明
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexCartesianChart クラス

説明

一般的なグラフ タイプの基本クラスです。CartesianChart は、標準的な長方形の 2 次元グラフの基本的なレイアウト動作を定義します。

継承階層

- [FlexChart](#)
 - FlexCartesianChart
 - [FlexAreaChart](#)
 - [FlexBarChart](#)
 - [FlexBubbleChart](#)
 - [FlexCandlestickChart](#)
 - [FlexColumnChart](#)
 - [FlexHLOCChart](#)
 - [FlexLineChart](#)
 - [FlexPlotChart](#)

構文

```
'Declaration
Public Class FlexCartesianChart _
Inherits FlexChart
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>clipContent</i>	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipMode</i>	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
<i>description</i>	グラフのデータの簡単な説明。(FlexChart から継承されました)
<i>dragEnabled</i>	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
<i>dropEnabled</i>	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChart から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexChart から継承されました)
<i>fontSize</i>	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。(FlexChart から継承されました)

名前	説明
gutterBottom	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。
gutterLeft	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。
gutterRight	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。
gutterTop	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
horizontalAxisRatio	水平軸の高さの制限。
id	コンポーネントの ID。 (FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (FlexChart から継承されました)
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。 (FlexChart から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。 (FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。 (FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。 デフォルト値は 2 です。 (FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。 デフォルト値は 0 です。 (FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)

名前	説明
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。(FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナー内のテキストの配置。
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。(TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)

名前	説明
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexChart クラス

説明

Flex コンポーネント アーキテクチャと DualStyleObject アーキテクチャの間のリンクです。ChartBase クラスは Flex UIComponent の基本クラスを拡張しますが、DualStyleObject の各クラスは ChartBase クラスに含まれています。

通常、ChartBase クラスを直接使用することはありません。代わりに、そのサブクラス (PlotChart または BubbleChart など) の 1 つを使用します。このクラスは、mx.charts パッケージで提供される一般的なグラフ タイプの基本クラスとして機能します。

このクラスは、フレームワークの一般的な機能に簡単にアクセスするための CSS スタイルおよび CSS プロパティの数を定義します。

グラフの最小サイズは 20 × 20 ピクセルです。グラフの最大サイズに制限はありません。グラフの推奨サイズは 400 × 400 ピクセルです。

継承階層

- [FlexObject](#)
 - FlexChart
 - [FlexCartesianChart](#)
 - [FlexPieChart](#)

構文

```
'Declaration  
Public Class FlexChart _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipContent	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。
columnNames	データ内のすべての列の名前を含むリスト。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dataTipMode	グラフの DataTip コントロールの表示方法。
description	グラフのデータの簡単な説明。
dragEnabled	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。
dropEnabled	ドラッグした項目をグラフ上にドロップできるかどうか。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
fontFamily	使用するフォントの名前。
fontSize	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。

名前	説明
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。
numRows	利用可能なデータ中の行の合計数。
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。 デフォルト値は 2 です。
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。 デフォルト値は 0 です。
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。 複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。
selectionMode	ChartItem オブジェクトが選択できるかどうか。

名前	説明
showDataTips	グラフの DataTip コントロールが表示されるかどうか。
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)

名前	説明
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 ToString 	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexChartLegend クラス

説明

グラフに凡例を追加できます。凡例にはグラフ内の各データ系列のラベルや、データ系列のグラフ要素を示すキーが表示されます。

継承階層

- [FlexObject](#)
 - FlexChartLegend

構文

```
'Declaration
Public Class FlexChartLegend _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。

名前	説明
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	凡例要素の隣に表示するテキスト。
<i>labelPlacement</i>	凡例要素のラベルの配置。有効な阿多灰、 <i>top</i> 、 <i>bottom</i> 、 <i>right</i> 、および <i>left</i> です。
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>markerAspectRatio</i>	この凡例項目に関連付けられたマーカーのアスペクト比。
<i>markerHeight</i>	凡例要素の高さ。
<i>markerWidth</i>	凡例要素の幅。
<i>mouseEnabled</i>	このオブジェクトがマウスメッセージを受け取るかどうか。デフォルト値は <i>true</i> であり、これは表示リスト上の <i>InteractiveObject</i> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <i>mouseEnabled</i> を <i>false</i> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <i>mouseEnabled</i> 動作を変更するには、 <i>flash.display.DisplayObjectContainer.mouseChildren</i> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。
<i>numRows</i>	利用可能なデータ中の行の合計数。
<i>paddingLeft</i>	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。
<i>paddingRight</i>	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <i>width</i> または <i>explicitWidth</i> プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)

名前	説明
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalGap	子同士の垂直方向の間隔を示すピクセル数。デフォルト値はコンポーネントのクラスに応じて異なります。クラスでオーバーライドされない場合のデフォルト値は 6 です。
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
LegendItemClick	ユーザーが Legend コントロール内の LegendItem をクリックしたときにディスパッチされます。ユーザーが凡例項目の上でマウス ボタンをクリックしたことを示します。
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexChartSeries クラス

説明

グラフ コントロールのデータ系列を指定するクラスの基本クラスです。

Series クラスのサブクラスは、関連付けられたグラフ コントロールと共に使用します。Series クラスを使用すると、データ系列に関連するグラフ要素の塗りつぶしパターンや線の属性を指定できます。

継承階層

- [FlexObject](#)
 - FlexChartSeries
 - [FlexAreaSeries](#)
 - [FlexBarSeries](#)
 - [FlexBubbleSeries](#)
 - [FlexColumnSeries](#)
 - [FlexHLOCSeriesBase](#)
 - [FlexLineSeries](#)
 - [FlexPieSeries](#)
 - [FlexPlotSeries](#)

構文

```
'Declaration  
Public Class FlexChartSeries _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。

名前	説明
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。
numRows	利用可能なデータ中の行の合計数。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。
selectedItem	系列内で選択されたグラフ項目。
selectedItems	系列内で選択されたグラフ項目の配列。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)

名前	説明
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインタを動かしたインデックスです。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

FlexCheckBox クラス

説明

CheckBox コントロールは、オプションのラベルと、チェック マークの表示と非表示の状態を保持する小さなボックスから構成されます。

オプションのテキスト ラベルは、CheckBox の左、右、上、または下に配置できます。 ユーザーが CheckBox コントロールまたは対応するテキストをクリックすると、チェック マークが表示された状態から表示されていない状態へ、またはチェック マークが表示されていない状態から表示された状態へ変化します。 CheckBox コントロールは、相互排他的でない一連のブール値 (true または false) を収集します。

継承階層

- [FlexButton](#)
 - FlexCheckBox

構文

```
'Declaration  
Public Class FlexCheckBox _  
Inherits FlexButton
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。(FlexButton から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
disabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。(FlexButton から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コントロールの背景の塗りつぶしに使用するアルファ値。(FlexButton から継承されました)
fillColors	コントロールの背景を濃淡色化するのに使用する色。(FlexButton から継承されました)
focusAlpha	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。(FlexButton から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
fontFamily	使用するフォントの名前。(FlexButton から継承されました)

名前	説明
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexButton</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexButton</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexButton</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	Button コントロールに表示されるテキスト。 (<i>FlexButton</i> から継承されました)
<i>labelPlacement</i>	指定されたアイコンを基準にしたラベルの向き。 (<i>FlexButton</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selected</i>	トグルボタンがオン (true) またはオフ (false) のどちらであるか。 (<i>FlexButton</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexButton から継承されました)
textDecoration	テキストに下線を表示するかどうか。(FlexButton から継承されました)
textIndent	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。(FlexButton から継承されました)
textRollOverColor	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。(FlexButton から継承されました)
textSelectedColor	ユーザーがラベルを選択したときのラベルのテキストの色。デフォルト値は 0x000000 です。(FlexButton から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toggle	Button を toggle 状態にするかどうか。(FlexButton から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(<code>TestObject</code> から継承されました)
<code>ChangeFocus</code>	現在のフォーカスを変更します。(<code>FlexObject</code> から継承されました)
<code>Click</code>	オブジェクトをクリックします。(<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	マウス ボタンをオブジェクト上でダブルクリックします。(<code>IClickable</code> から継承されました)
<code>Exists</code>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(<code>TestObject</code> から継承されました)
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<code>TestObject</code> から継承されました)
<code>Find</code>	XPath ロケーターで指定したオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>FindAll</code>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。(<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。(<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。(<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetRect</code>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<code>TestObject</code> から継承されました)
<code>GetStyle</code>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<code>FlexDisplayObject</code> から継承されました)
<code>HighlightObject</code>	このオブジェクトをハイライトします。(<code>TestObject</code> から継承されました)
<code>ImageClick</code>	指定したイメージ資産をクリックします。(<code>TestObject</code> から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (FlexButton から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexColorPicker クラス

説明

ユーザーが色見本リストから色を選択する手段を提供します。

コンポーネントのデフォルトのモードでは、正方形のボタンに単一の色見本が表示されます。ユーザーが色見本のボタンをクリックすると、色見本パネルが開き、色見本リスト全体が表示されます。

継承階層

- [FlexComboBase](#)
 - FlexColorPicker

構文

```
'Declaration  
Public Class FlexColorPicker _  
Inherits FlexComboBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。(FlexComboBase から継承されました)
backgroundColor	コンポーネントの背景色。(FlexComboBase から継承されました)
backgroundImage	コンポーネントの背景画像。(FlexComboBase から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexComboBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。(FlexComboBase から継承されました)
columnCount	色見本グリッドの列の数。デフォルト値は 20 です。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)

名前	説明
<i>disabledColor</i>	コンポーネントが無効な場合の背景色。(FlexComboBase から継承されました)
<i>dropShadowEnabled</i>	ComboBase コンテナのドロップ シャドウが可視かどうか。デフォルト値は true です。(FlexComboBase から継承されました)
<i>editable</i>	コントロールが編集可能であるかどうか。編集可能な場合は、dataProvider で指定されていないエントリをユーザーが直接入力できます。編集できない場合は、dataProvider 内の項目から選択する必要があります。(FlexComboBase から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コントロールの背景の塗りつぶしに使用するアルファ値。(FlexComboBase から継承されました)
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。(FlexComboBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexComboBase から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexComboBase から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexComboBase から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexComboBase から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>highlightColor</i>	ドロップダウン リストのハイライト色。(FlexComboBase から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)

名前	説明
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>restrict</i>	ユーザーがテキストフィールドに入力できる文字、または入力できない文字のセット。 (<i>FlexComboBase</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selectedColor</i>	SwatchPanel オブジェクト内で現在選択されている色の値。
<i>selectedIndex</i>	データプロバイダー内で選択された項目のインデックス。 (<i>FlexComboBase</i> から継承されました)
<i>showTextField</i>	カラー ラベルまたは 16 進数のカラー値を示すテキスト ボックスを表示するかどうか。デフォルト値は true です。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>text</i>	テキスト フィールドの内容。 (<i>FlexComboBase</i> から継承されました)
<i>textAlign</i>	コンテナ内のテキストの配置。有効な値は left、right または center です。 (<i>FlexComboBase</i> から継承されました)
<i>textDecoration</i>	テキストに下線を表示するかどうか。 (<i>FlexComboBase</i> から継承されました)
<i>textIndent</i>	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。 (<i>FlexComboBase</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の

名前	説明
	場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
Change	ユーザーによる操作の結果として選択した色が変更されたときにディスプレイパッチされます。
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	close イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexComboBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Input	テキスト イベントについての情報を含むイベント オブジェクトを作成します。 イベント オブジェクトは、パラメーターとしてイベント リスナーに渡されます。(FlexComboBase から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
Open	open イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexComboBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexComboBase から継承されました)
SelectText	テキスト フィールド内の選択範囲を追跡します。 (FlexComboBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (FlexComboBase から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

FlexColumnChart クラス

説明

データ内の値によって高さが決まる一連の縦棒でデータを表します。

ColumnChart を使用すると、単純縦棒グラフ、集合縦棒グラフ、積み上げ縦棒グラフ、構成比縦棒グラフ、高低縦棒グラフなど、各種の縦棒グラフを表すことができます。

継承階層

- [FlexCartesianChart](#)
 - FlexColumnChart

構文

```
'Declaration
Public Class FlexColumnChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipContent	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
columnWidthRatio	縦棒グラフを描く際のカテゴリ幅に相対の広さ (0 から 1 の範囲の割合)。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dataTipMode	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
description	グラフのデータの簡単な説明。(FlexChart から継承されました)
dragEnabled	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
dropEnabled	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChart から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexChart から継承されました)
<i>fontSize</i>	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。(FlexChart から継承されました)
<i>gutterBottom</i>	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ(ピクセル単位)。(FlexCartesianChart から継承されました)
<i>gutterLeft</i>	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ(ピクセル単位)。(FlexCartesianChart から継承されました)
<i>gutterRight</i>	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ(ピクセル単位)。(FlexCartesianChart から継承されました)
<i>gutterTop</i>	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ(ピクセル単位)。(FlexCartesianChart から継承されました)
<i>height</i>	表示オブジェクトの高さ(ピクセル単位)。(FlexDisplayObject から継承されました)
<i>horizontalAxisRatio</i>	水平軸の高さの制限。(FlexCartesianChart から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexChart から継承されました)
<i>maxColumnWidth</i>	描画する縦棒の幅(ピクセル単位)。
<i>maximumDataTipCount</i>	グラフが表示するデータ ヒントの最大数。(FlexChart から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>mouseSensitivity</i>	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離(ピクセル単位)。(FlexChart から継承されました)

名前	説明
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。 (FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。 (FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。 (FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。 (FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。 (FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。 (FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。 (FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。 (FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。(FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)

名前	説明
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexColumnSeries クラス

説明

ColumnChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexColumnSeries

構文

```
'Declaration  
Public Class FlexColumnSeries _  
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
columnWidthRatio	カテゴリ幅に相対的な縦棒グラフの幅。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChartSeries から継承されました)
maxColumnWidth	縦棒の幅 (ピクセル単位)。

名前	説明
minField	縦棒下端の y 軸の位置を決定するデータ プロバイダーのフィールド。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChartSeries から継承されました)
offset	使用可能な領域の中心から縦棒の中心をずらす度合い (カテゴリの幅に対する割合)。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。(FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
sortOnXField	レンダリングの前に縦棒を左から右にソートするかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
xField	縦棒の x 軸の位置を決定するデータ プロバイダーのフィールド。
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)
yField	縦棒上端の y 軸の位置を決定するデータ プロバイダーのフィールド。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>GetValues</i>	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(<i>FlexChartSeries</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインターを動かしたインデックスです。 (FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexComboBase クラス

説明

テキストを表示するフィールドのほか、ユーザーが選択するテキストを表示するドロップダウン リストを表示するためのボタンを持つコントロールの基本クラスです。

継承階層

- *FlexObject*
 - FlexComboBox
 - *FlexColorPicker*
 - *FlexComboBox*
 - *FlexDateField*

構文

```
'Declaration  
Public Class FlexComboBox _  
Inherits FlexObject
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(<i>FlexDisplayObject</i> から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(<i>FlexObject</i> から継承されました)
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。
<i>backgroundColor</i>	コンポーネントの背景色。
<i>backgroundImage</i>	コンポーネントの背景画像。
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(<i>FlexObject</i> から継承されました)
<i>color</i>	コンポーネント ラベルを含むコンポーネントのテキストの色。
<i>currentState</i>	コンポーネントの現在の表示状態。(<i>FlexObject</i> から継承されました)
<i>disabledColor</i>	コンポーネントが無効な場合の背景色。
<i>dropShadowEnabled</i>	ComboBox コンテナのドロップ シャドウが可視かどうか。デフォルト値は true です。
<i>editable</i>	コントロールが編集可能であるかどうか。編集可能な場合は、dataProvider で指定されていないエントリをユーザーが直接入力できます。編集できない場合は、dataProvider 内の項目から選択する必要があります。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(<i>FlexObject</i> から継承されました)

名前	説明
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
fillAlphas	コントロールの背景の塗りつぶしに使用するアルファ値。
fillColors	コントロールの背景を濃淡色化するのに使用する色。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
fontFamily	使用するフォントの名前。
fontSize	テキストの高さ (ピクセル単位)。
fontStyle	テキストがイタリック フォントかどうか。
fontWeight	テキストがボールド体かどうか。
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
highlightColor	ドロップダウン リストのハイライト色。
id	コンポーネントの ID。 (FlexObject から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
restrict	ユーザーがテキスト フィールドに入力できる文字、または入力できない文字のセット。
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedIndex	データ プロバイダー内で選択された項目のインデックス。

名前	説明
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TextObject から継承されました)
text	テキスト フィールドの内容。
textAlign	コンテナ内のテキストの配置。有効な値は left、right または center です。
textDecoration	テキストに下線を表示するかどうか。
textIndent	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TextObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	close イベントのイベント オブジェクトの type プロパティの値を定義します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Input</i>	テキスト イベントについての情報を含むイベント オブジェクトを作成します。 イベント オブジェクトは、パラメーターとしてイベント リスナーに渡されます。
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(<i>FlexObject</i> から継承されました)
<i>Open</i>	open イベントのイベント オブジェクトの type プロパティの値を定義します。
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Scroll</i>	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。
<i>SelectText</i>	テキスト フィールド内の選択範囲を追跡します。

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexComboBox クラス

説明

ユーザーが 1 つの値を選択できるドロップダウン リストが含まれています。

ComboBox は編集可能にできます。この場合、リストに含まれないエントリを、ユーザーが ComboBox の TextInput 部分に入力できます。

継承階層

- [FlexComboBase](#)
 - FlexComboBox

構文

```
'Declaration
Public Class FlexComboBox _
Inherits FlexComboBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alternatingItemColors	ドロップダウン リストの背景色の代替パターン配列。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。(FlexComboBase から継承されました)
backgroundColor	コンポーネントの背景色。(FlexComboBase から継承されました)
backgroundImage	コンポーネントの背景画像。(FlexComboBase から継承されました)

名前	説明
<i>backgroundSize</i>	指定した <i>backgroundImage</i> の画像サイズを変更する割合。 (<i>FlexComboBase</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>color</i>	コンポーネント ラベルを含むコンポーネントのテキストの色。 (<i>FlexComboBase</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>disabledColor</i>	コンポーネントが無効な場合の背景色。 (<i>FlexComboBase</i> から継承されました)
<i>dropShadowEnabled</i>	<i>ComboBase</i> コンテナのドロップ シャドウが可視かどうか。デフォルト値は true です。 (<i>FlexComboBase</i> から継承されました)
<i>editable</i>	コントロールが編集可能であるかどうか。編集可能な場合は、 <i>dataProvider</i> で指定されていないエントリをユーザーが直接入力できます。編集できない場合は、 <i>dataProvider</i> 内の項目から選択する必要があります。 (<i>FlexComboBase</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コントロールの背景の塗りつぶしに使用するアルファ値。 (<i>FlexComboBase</i> から継承されました)
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。 (<i>FlexComboBase</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexComboBase</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexComboBase</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexComboBase</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexComboBase</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>highlightColor</i>	ドロップダウン リストのハイライト色。 (<i>FlexComboBase</i> から継承されました)

名前	説明
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。
<i>numRows</i>	利用可能なデータ中の行の合計数。
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>restrict</i>	ユーザーがテキスト フィールドに入力できる文字、または入力できない文字のセット。 (<i>FlexComboBase</i> から継承されました)
<i>rollOverColor</i>	ドロップダウン リストの rollOverColor。
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selectedIndex</i>	データ プロバイダー内で選択された項目のインデックス。 (<i>FlexComboBase</i> から継承されました)
<i>selectionColor</i>	ドロップダウン リストの selectionColor。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>text</i>	テキスト フィールドの内容。 (<i>FlexComboBase</i> から継承されました)

名前	説明
textAlign	コンテナ内のテキストの配置。有効な値は left、right または center です。(FlexComboBase から継承されました)
textDecoration	テキストに下線を表示するかどうか。(FlexComboBase から継承されました)
textIndent	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。(FlexComboBase から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	close イベントのイベントオブジェクトの type プロパティの値を定義します。(FlexComboBase から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Input	テキスト イベントについての情報を含むイベント オブジェクトを作成します。 イベント オブジェクトは、パラメーターとしてイベント リスナーに渡されます。(FlexComboBase から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
Open	open イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexComboBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexComboBase から継承されました)
Select	単一の項目が選択されたときにディスパッチされるイベントの、 イベント オブジェクトの type プロパティの値を定義します。
SelectIndex	単一の項目が選択されたときにディスパッチされるイベントの、 イベント オブジェクトの type プロパティの値を定義します。
SelectText	テキスト フィールド内の選択範囲を追跡します。(FlexComboBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。 コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 ToString 	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
 Type 	ユーザーがキーを押したときにディスパッチされます。(FlexComboBase から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexContainer クラス

説明

コンポーネントの抽象基本クラスであり、子コンポーネントのレイアウト特性を制御します。

アプリケーションでは、Container のインスタンスを作成しません。代わりに、Container のいずれかのサブクラス (Canvas や HBox など) のインスタンスを作成します。

Container クラスには、スクロール、切り取り、およびダイナミックなインスタンス化のためのロジックが含まれています。また、子コンテナを追加および削除するためのメソッドや、コンテナの背景と境界を描画するためのロジックも含まれています。

継承階層

- [FlexObject](#)
 - FlexContainer
 - [FlexAccordion](#)
 - [FlexApplication](#)
 - [FlexBox](#)
 - [FlexCanvas](#)
 - [FlexForm](#)
 - [FlexFormItem](#)
 - [FlexPanel](#)
 - [FlexViewStack](#)
 - [FlexWindow](#)

構文

```
'Declaration
Public Class FlexContainer _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)

名前	説明
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。
backgroundColor	コンポーネントの背景色。
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。
creationIndex	このコンテナの子をインスタンス化および描画する順番。
creationPolicy	このコンテナの子の作成ポリシー。
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
fillAlphas	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。
fillColors	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。
firstVisibleRow	最初の可視の子のインデックス。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
fontFamily	使用するフォントの名前。
fontSize	テキストの高さ (ピクセル単位)。
fontStyle	テキストがイタリック フォントかどうか。
fontWeight	テキストがボールド体かどうか。
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
label	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。

名前	説明
lastVisibleRow	最後の可視の子のインデックス。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numChildren	このコンテナに含まれる子コンポーネントの数。
numColumns	利用可能なデータ中の列の合計数。
numRows	利用可能なデータ中の行の合計数。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexContainerMovieClip クラス

説明

Adobe Flash CS3 Professional で Flex 用に作成するコンテナ コンポーネントは、mx.flash.ContainerMovieClip クラスのサブクラスになります。

ContainerMovieClip のサブクラスは、Flex コンテナとして使用したり、子を保持したりできます。また、このサブクラスを使用すると、Flex コンポーネントと同じ方法で、イベントに対する応答、ビューの状態や遷移の定義およびエフェクトの操作を行うことができます。

Flash コンテナは、1 つの Flex の子のみを保持できます。ただし、この子は、別の子を追加できる Flex コンテナにすることができます。

継承階層

- [FlexUIMovieClip](#)
 - FlexContainerMovieClip

構文

```
'Declaration  
Public Class FlexContainerMovieClip _  
Inherits FlexUIMovieClip
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexUIMovieClip から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexUIMovieClip から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexUIMovieClip から継承されました)
className	"Button" のようなこのインスタンスのクラスの名前。(FlexUIMovieClip から継承されました)
currentState	このコンポーネントの現在の状態。(FlexUIMovieClip から継承されました)
enabled	ムービー クリップが有効かどうか。(FlexUIMovieClip から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexUIMovieClip から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexUIMovieClip から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexUIMovieClip から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexUIMovieClip から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexUIMovieClip から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。有効な値は 0 から 100 です。(FlexUIMovieClip から継承されました)

名前	説明
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。有効な値は 0 から 100 です。(FlexUIMovieClip から継承されました)
scaleX	基準点から適用されるオブジェクトの水平倍率（パーセンテージ）。デフォルトの基準点は (0,0) です。1.0 は 100% の倍率です。(FlexUIMovieClip から継承されました)
scaleY	オブジェクトの基準点から適用されるオブジェクトの垂直倍率（パーセンテージ）。デフォルトの基準点は (0,0) です。1.0 は 100% の倍率です。(FlexUIMovieClip から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexUIMovieClip から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexUIMovieClip から継承されました)
useHandCursor	buttonMode プロパティが true に設定されたスプライト上にポインターが移動したときに、指差しハンドポインター（ハンドカーソル）を表示するかどうか。(FlexUIMovieClip から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保

名前	説明
	<p>存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	<p>keyFocusChange イベント オブジェクトの type プロパティの値を定義します。 (FlexUIMovieClip から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
DragDrop	<p>dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
GetStyle	<p>指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)</p>

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コンポーネントがフォーカスを受け取ったときに FocusManager によって呼ばれます。コンポーネントはその内部のコンポーネントに順々にフォーカスを設定できます。(FlexUIMovieClip から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexDataGrid クラス

説明

DataGrid コントロールは、List コントロールに似ていますが、複数の列のデータを表示でき、複数のプロパティを持つオブジェクトの表示に適している点が異なります。

継承階層

- [FlexListBase](#)
 - FlexDataGrid

構文

```
'Declaration  
Public Class FlexDataGrid _  
Inherits FlexListBase
```

プロパティ

名前	説明
allowMultipleSelection	1 つ以上の項目を同時に選択することが許可されているかどうか。 (FlexListBase から継承されました)
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
alternatingItemColors	リストの項目の背景に使用する色。 (FlexListBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。 (FlexListBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnCount	TileList コントロールまたは HorizontalList コントロールの項目に表示される列の数。 (FlexListBase から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。 (FlexListBase から継承されました)
columnWidth	コントロールの列の幅。 (FlexListBase から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
dataTipField	データ ヒントとして表示するデータ プロバイダー項目内のフィールドの名前。 (FlexListBase から継承されました)
editable	データ プロバイダーの項目を編集できるかどうか。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)

名前	説明
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexListBase から継承されました)
<i>focusAlpha</i>	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。(FlexListBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>headerColors</i>	ヘッダーの背景にグラデーションを描画するために使用する 2 色の配列。
<i>headerStyleName</i>	列ヘッダーの他の外観要素を制御する CSS スタイル宣言の名前。デフォルト値は <i>dataGridStyles</i> です。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>horizontalGridLineColor</i>	水平グリッド線の色。
<i>horizontalGridLines</i>	行の間に水平グリッド線を表示するかどうか。
<i>iconField</i>	アイコンとして表示するものを決定する、データ プロバイダー オブジェクト内のフィールドの名前。(FlexListBase から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。(FlexListBase から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexListBase から継承されました)
<i>lockedColumnCount</i>	スクロールするコントロールの最初の列のインデックス。(FlexListBase から継承されました)
<i>lockedRowCount</i>	スクロールするコントロールの最初の行のインデックス。(FlexListBase から継承されました)
<i>minColumnWidth</i>	列の最小幅 (ピクセル単位)。NaN ではない場合、DataGrid コントロールではこの値がすべての列の最小幅として適用されます。それ以外の場合は、個々の列に最小幅を指定できます。デフォルト値は NaN です。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)

名前	説明
numColumns	利用可能なデータ中の列の合計数。 (FlexListBase から継承されました)
numRows	利用可能なデータ中の行の合計数。 (FlexListBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
resizableColumns	列のサイズをユーザーが変更できるかどうか。
rollOverColor	ユーザーがレンダラの上にマウスを置いたときのレンダラの背景色。 デフォルト値は 0xEEFEE6 です。 (FlexListBase から継承されました)
rowCount	表示する行の数。 (FlexListBase から継承されました)
rowHeight	行の高さ (ピクセル単位)。 (FlexListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectable	選択された項目を選択状態でリストに表示するかどうか。 (FlexListBase から継承されました)
selectedIndex	データプロバイダー内で選択された項目のインデックス。 (FlexListBase から継承されました)
selectedIndices	データプロバイダーで選択された項目のインデックスの配列。 (FlexListBase から継承されました)
selectedItem	データプロバイダー内で選択された項目への参照。 (FlexListBase から継承されました)
selectedItems	データプロバイダーで選択された項目への参照の配列。 (FlexListBase から継承されました)
selectionColor	ユーザーが選択したときのレンダラの背景色。 デフォルト値は 0x7FCEFF です。 (FlexListBase から継承されました)
selectionDisabledColor	コンポーネントが無効化されたときのレンダラの背景色。 デフォルト値は 0xDDDDDD です。 (FlexListBase から継承されました)
showDataTips	行内のテキストに対して dataTips を表示するかどうか。 (FlexListBase から継承されました)
sortableColumns	ユーザーが列ヘッダーセルをクリックすることでデータプロバイダー項目をソートできるかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)

名前	説明
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>textRollOverColor</i>	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。 (<i>FlexListBase</i> から継承されました)
<i>textSelectedColor</i>	ユーザーが選択したときのレンダラのテキスト色。デフォルト値は 0x2B333C です。 (<i>FlexListBase</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (<i>FlexObject</i> から継承されました)
<i>useRollOver</i>	項目の上にマウスを置いたときに、その項目をハイライト表示するかどうか。 (<i>FlexListBase</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>variableRowHeight</i>	個々の行が別々の高さを持つことができるかどうか。 (<i>FlexListBase</i> から継承されました)
<i>verticalAlign</i>	行のレンダラの垂直方向の配置。 (<i>FlexListBase</i> から継承されました)
<i>verticalGridLineColor</i>	垂直グリッド線の色。デフォルト値は 0x666666 です。
<i>verticalGridLines</i>	列の間に垂直グリッド線を表示するかどうか。
<i>visible</i>	表示オブジェクトが可視かどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>wordWrap</i>	行内のテキストを折り返すかどうか。 (<i>FlexListBase</i> から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
ColumnStretch	ユーザーが列の幅を変更したときにディスプレイされます。これは、列に表示されるデータ量に変更されたことを意味します。
Deselect	以前に選択した項目の選択を解除するときにディスプレイされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClick	コントロールの可視項目上でマウスがダブルクリックされたことを示す <code>itemDoubleClick</code> イベントにおける <code>ListEvent</code> オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
Edit	項目が編集可能な状態かどうかを示す <code>itemEditBegin</code> イベントにおけるイベント オブジェクトの type プロパティの値を定義します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexListBase から継承されました)
HeaderClick	列の内容に基づいてグリッドの内容の並べ替えを行うために、ユーザーが列ヘッダーでマウス ボタンを離れたときにディスパッチされます。
HeaderShift	ユーザーが列ヘッダーを新しい場所にドラッグし、マウス ボタンを離れたときにディスパッチされます。その結果、その列は新しいインデックスに移されます。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
MultiSelect	複数項目を選択するアクションの一部として項目が選択されたときにディスパッチされるイベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
ScrollToIndex	指定したインデックス位置にあるデータ プロバイダー項目を表示状態にします。 (FlexListBase から継承されました)
Select	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
SelectIndex	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexDateChooser クラス

説明

DateChooser コントロールは、月の名前、年、および月の各日を示すグリッドを表示し、そのグリッドの列ラベルとして曜日を表示します。

ユーザーはある日付、日付の範囲、または複数の日付を選択できます。このコントロールには、月と年を前後に移動するための矢印ボタンがあります。

ユーザーが複数の日付を選択できるようにしたり、特定の日付を選択できないようにしたり、表示される日付を一定の範囲に制限したりすることができます。

継承階層

- [FlexObject](#)
 - FlexDateChooser

構文

```
'Declaration
Public Class FlexDateChooser _
Inherits FlexObject
```

プロパティ

名前	説明
allowDisjointSelection	DateChooser コントロールで非連続した選択が許可されているかどうか。
allowMultipleSelection	DateChooser コントロールで複数選択が許可されているかどうか。
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。 デフォルト値は 0x0B333C です。
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
disabledColor	コンポーネントが無効な場合のテキストの色。 デフォルト値は 0xAAB3B3 です。
disabledDays	無効にする曜日。
disabledRanges	無効にする 1 日または複数の日。

名前	説明
<i>displayedMonth</i>	DateChooser コントロールに表示される月。
<i>displayedYear</i>	DateChooser コントロールに表示される年。デフォルト値は現在の年です。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。
<i>firstDayOfWeek</i>	DateChooser コントロールの最初の列に表示される曜日を表す数値。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。
<i>fontSize</i>	テキストの高さ (ピクセル単位)。
<i>fontStyle</i>	テキストがイタリック フォントかどうか。
<i>fontWeight</i>	テキストがボールド体かどうか。
<i>headerColors</i>	DateChooser コントロールの上部にある帯状の領域の色。デフォルト値は [0xE6EEEE, 0xFFFFF] です。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>maxYear</i>	コントロール内で選択可能な最後の年。
<i>minYear</i>	コントロール内で選択可能な最初の年。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナーに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、こ

名前	説明
	のプロパティは NaN にリセットされます。(FlexObject から継承されました)
rollOverColor	DateChooser コントロール内の日付にマウス ポインターを合わせたときの日付のハイライト領域の色。デフォルト値は 0xE3FFD6 です。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedDate	DateChooser コントロールで選択された日付。
selectedRanges	選択された日付の範囲。
selectionColor	DateChooser コントロールで現在選択されている日付のハイライト領域の色。デフォルト値は 0xCDFFC1 です。
showToday	DateChooser コントロールで今日の日付がハイライトされているかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	テキストがボールド体かどうか。
textDecoration	テキストに下線を表示するかどうか。
textIndent	テキストの 1 行目のコンテナー左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
todayColor	DateChooser コントロールで今日の日付のハイライトの色。デフォルト値は 0x2B333 です。
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>yearNavigationEnabled</i>	年ナビゲーションを有効かどうか。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Change</i>	日付が選択または変更されたときにディスパッチされます。
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。 (<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。 (<i>TestObject</i> から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollDate	DateChooserEvent.SCROLL 定数は、scrollevent イベントのイベント オブジェクトの type プロパティの値を定義します。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexDateField クラス

説明

右側にカレンダー アイコンを持った日付を表示するテキスト フィールドです。

ユーザーがこのコントロールの境界ボックス内をクリックすると、DateChooser コントロールがポップアップし、現在の日付の月に含まれる日が一覧表示されます。日が選択されない場合、このテキスト フィールドは空になり、DateChooser コントロールには現在の日付の月が表示されます。

継承階層

- [FlexComboBase](#)
 - FlexDateField

構文

```
'Declaration
Public Class FlexDateField _
Inherits FlexComboBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)

名前	説明
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。 (<i>FlexComboBase</i> から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。 (<i>FlexComboBase</i> から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。 (<i>FlexComboBase</i> から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。 (<i>FlexComboBase</i> から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>color</i>	コンポーネント ラベルを含むコンポーネントのテキストの色。 (<i>FlexComboBase</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>disabledColor</i>	コンポーネントが無効な場合の背景色。 (<i>FlexComboBase</i> から継承されました)
<i>disabledDays</i>	無効にする曜日。
<i>disabledRanges</i>	無効にされた 1 日または複数の日。
<i>displayedMonth</i>	DateChooser コントロールに表示される月。
<i>displayedYear</i>	DateChooser コントロールに表示される年。
<i>dropShadowEnabled</i>	ComboBase コンテナのドロップ シャドウが可視かどうか。デフォルト値は true です。 (<i>FlexComboBase</i> から継承されました)
<i>editable</i>	コントロールが編集可能であるかどうか。編集可能な場合は、dataProvider で指定されていないエントリをユーザーが直接入力できます。編集できない場合は、dataProvider 内の項目から選択する必要があります。 (<i>FlexComboBase</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コントロールの背景の塗りつぶしに使用するアルファ値。 (<i>FlexComboBase</i> から継承されました)
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。 (<i>FlexComboBase</i> から継承されました)
<i>firstDayOfWeek</i>	DateChooser コントロールの最初の列に表示する曜日 (0-6、ここで 0 は dayNames 配列の最初の要素)。

名前	説明
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexComboBase</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexComboBase</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexComboBase</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexComboBase</i> から継承されました)
<i>headerColors</i>	DateChooser コントロールの上部にある帯状の領域の色。デフォルト値は [0xE6EEEE, 0xFFFFFFFF] です。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>highlightColor</i>	ドロップダウン リストのハイライト色。 (<i>FlexComboBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>maxYear</i>	コントロール内で選択可能な最後の年。
<i>minYear</i>	コントロール内で選択可能な最初の年。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>restrict</i>	ユーザーがテキスト フィールドに入力できる文字、または入力できない文字のセット。 (<i>FlexComboBase</i> から継承されました)
<i>rollOverColor</i>	DateChooser コントロール内の日付にマウス ポインターを合わせたときの日付のハイライト領域の色。デフォルト値は 0xE3FFD6 です。
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)

名前	説明
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedDate	DateChooser コントロールで選択された日付。
selectedIndex	データ プロバイダー内で選択された項目のインデックス。 (FlexComboBase から継承されました)
showToday	DateChooser コントロールにハイライトされる日。
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
text	テキスト フィールドの内容。 (FlexComboBase から継承されました)
textAlign	コンテナー内のテキストの配置。 有効な値は left、right または center です。 (FlexComboBase から継承されました)
textDecoration	テキストに下線を表示するかどうか。 (FlexComboBase から継承されました)
textIndent	テキストの 1 行目のコンテナー左端からのオフセット (ピクセル単位)。 デフォルト値は 0 です。 (FlexComboBase から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
todayColor	DateChooser コントロールで今日の日付のハイライトの色。 デフォルト値は 0x2B333 です。
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)

名前	説明
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。 (FlexDisplayObject から継承されました)
yearNavigationEnabled	年のナビゲーション用上下ボタンが利用可能かどうか。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Change	日付が選択または変更され、 <code>DateChooser</code> コントロールが閉じられたときにディスパッチされます。
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Close	<code>close</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexComboBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Input	テキスト イベントについての情報を含むイベント オブジェクトを作成します。 イベント オブジェクトは、パラメーターとしてイベント リスナーに渡されます。(FlexComboBase から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)

名前	説明
Open	open イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexComboBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexComboBase から継承されました)
ScrollDate	DateChooserEvent.SCROLL 定数は、scrollevent イベントのイベント オブジェクトの type プロパティの値を定義します。
SelectText	テキスト フィールド内の選択範囲を追跡します。(FlexComboBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexComboBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexDisplayObject クラス

説明

表示リストに含めることのできるすべてのオブジェクトに関する基本クラスです。表示リストは、Flash Player または Adobe AIR に表示されるすべてのオブジェクトを管理します。

継承階層

- [TestObject](#)
 - FlexDisplayObject
 - [FlexListLabel](#)
 - [FlexObject](#)
 - [FlexUIMovieClip](#)

構文

```
'Declaration
Public Class FlexDisplayObject _
```

Inherits TestObject _
Implements IClickable, IKeyable

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。
height	表示オブジェクトの高さ (ピクセル単位)。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。
tabIndex	SWF ファイル内のオブジェクトのタブ順序。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。
width	表示オブジェクトの幅 (ピクセル単位)。
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ToString</i>	コントロールの文字列表現を返します。
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexDividedBox クラス

説明

Box コンテナと同じ方法で子を水平方向または垂直方向に測定して配置しますが、子の間にはドラッグ可能な仕切りが挿入されます。ユーザーは、任意の側の仕切りをドラッグして子のサイズを変更できます。

DividedBox クラスは *HDividedBox* および *VDividedBox* クラスの基本クラスです。

継承階層

- [FlexBox](#)
 - FlexDividedBox

構文

```
'Declaration
Public Class FlexDividedBox _
Inherits FlexBox
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。(FlexContainer から継承されました)
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>direction</i>	垂直レイアウト (デフォルト)、水平レイアウトのどちらを使用するか。(FlexBox から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)

名前	説明
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalAlign</i>	このコンテナ内の子の水平方向の配置。 (<i>FlexBox</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は

名前	説明
	NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizeToContent	DividedBox が自動的に子のサイズに変更されるかどうか。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAlign	このコンテナ内の子の垂直方向の配置。(FlexBox から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	<p>現在のフォーカスを変更します。 (FlexObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
DragCancel	<p><code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)</p>
DragDrop	<p><code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)</p>
Dragged	<p>ユーザーが分割線をドラッグするときに複数回ディスパッチされます。 <code>dividerDrag</code> イベントは、 <code>dividerPress</code> イベントの後、および <code>dividerRelease</code> イベントの前にディスパッチされます。</p>
DragStart	<p><code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
Pressed	ユーザーがこのコンテナの分割線を押したときにディスパッチされます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Released	ユーザーが分割線を離れたときにディスパッチされます。
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexForm クラス

説明

フォームのレイアウトを制御し、フォーム フィールドが必須かオプションであることを明示し、エラー メッセージを処理し、フォーム データを Flex データ モデルにバインドしてデータのチェックと検証を実行することができます。また、スタイルシートを使用してフォームの外観を設定することもできます。

継承階層

- [FlexContainer](#)
 - FlexForm

構文

```
'Declaration
Public Class FlexForm _
Inherits FlexContainer _
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)

名前	説明
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。 (FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。 (FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。 (FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。 (FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。 (FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。 (FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。 (FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。 (FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
fillAlphas	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (FlexContainer から継承されました)
fillColors	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (FlexContainer から継承されました)
firstVisibleRow	最初の可視の子のインデックス。 (FlexContainer から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
fontFamily	使用するフォントの名前。 (FlexContainer から継承されました)
fontSize	テキストの高さ (ピクセル単位)。 (FlexContainer から継承されました)

名前	説明
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)

名前	説明
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexFormItem クラス

説明

ラベル、および水平方向または垂直方向に配置されている子を定義します。フォームの子には、コントロールやその他のコンテナなどがあります。1つのFormコンテナで複数のFormItemコンテナを保持できます。

継承階層

- [FlexContainer](#)
 - FlexFormItem

構文

```
'Declaration  
Public Class FlexFormItem _  
Inherits FlexContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)

名前	説明
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexContainer から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexContainer から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexContainer から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>horizontalAlign</i>	このコンテナ内の子の水平方向の配置。
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)

名前	説明
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (<i>FlexObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プ

名前	説明
	ロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexHLOCChart クラス

説明

HLOCChart (高値、安値、始値、終値) コントロールは、データ系列の高値、安値、始値、終値などの財務データを一連のローソク足で示します。

各要素の垂直線の上端および下端が、データ ポイントの高値および安値になります。右向きが目盛りが終値になり、始値を指定した場合には左向きが目盛りが始値になります。

継承階層

- [FlexCartesianChart](#)
 - FlexHLOCChart

構文

```
'Declaration
Public Class FlexHLOCChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>clipContent</i>	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には <i>false</i> を設定します。true に設定すると、データがレンダリングされるときにクリップされません。 (<i>FlexChart</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>FlexChart</i> から継承されました)
<i>columnWidthRatio</i>	HLOC 線を描く際の水平軸のカテゴリ幅に相対の広さ (0 から 1 の範囲の割合)。
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>dataTipMode</i>	グラフの DataTip コントロールの表示方法。 (<i>FlexChart</i> から継承されました)
<i>description</i>	グラフのデータの簡単な説明。 (<i>FlexChart</i> から継承されました)
<i>dragEnabled</i>	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。 (<i>FlexChart</i> から継承されました)
<i>dropEnabled</i>	ドラッグした項目をグラフ上にドロップできるかどうか。 (<i>FlexChart</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexChart</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexChart</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。 (<i>FlexChart</i> から継承されました)

名前	説明
gutterBottom	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterLeft	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterRight	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterTop	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
horizontalAxisRatio	水平軸の高さの制限。(FlexCartesianChart から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChart から継承されました)
maxColumnWidth	描画する HLOC 線の幅 (ピクセル単位)。
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。(FlexChart から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域とにあるピクセル数。(FlexChart から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端とにあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)

名前	説明
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。(FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。(FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexHLOCSeries クラス

説明

データ系列の高値、安値、終値、および始値（オプション）などの財務データを一連の要素で示します。各要素の垂直線の上端および下端が、データポイントの高値および安値になります。右向きが目盛りが終値になり、始値を指定した場合には左向きが目盛りが始値になります。

継承階層

- [FlexHLOCSeriesBase](#)
 - FlexHLOCSeries

構文

```
'Declaration  
Public Class FlexHLOCSeries _  
Inherits FlexHLOCSeriesBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
columnWidthRatio	カテゴリ幅に相対的な要素の幅。(FlexHLOCSeriesBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChartSeries から継承されました)
maxColumnWidth	要素の幅 (ピクセル単位)。(FlexHLOCSeriesBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味しま

名前	説明
	す。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChartSeries から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。(FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の

名前	説明
	場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
xField	要素の x 軸の位置を決定するデータ プロバイダーのフィールド。(FlexHLOCSeriesBase から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインタを動かしたインデックスです。(FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexHLOCSeriesBase クラス

説明

データ系列の高値、安値、終値、および始値（オプション）などの財務データを一連の要素で示します。各要素の垂直線の上端および下端が、データポイントの高値および安値になります。右向きが目盛りが終値になり、始値を指定した場合には左向きが目盛りが始値になります。

継承階層

- [FlexChartSeries](#)
 - FlexHLOCSeriesBase
 - [FlexCandlestickSeries](#)
 - [FlexHLOCSeries](#)

構文

```
'Declaration  
Public Class FlexHLOCSeriesBase _  
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
columnWidthRatio	カテゴリ幅に相対的な要素の幅。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChartSeries から継承されました)
maxColumnWidth	要素の幅 (ピクセル単位)。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタ

名前	説明
	スの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(<i>FlexChartSeries</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(<i>FlexChartSeries</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>selectedIndex</i>	データプロバイダーのデータ内で選択された項目のインデックス。(<i>FlexChartSeries</i> から継承されました)
<i>selectedIndices</i>	データプロバイダーのデータ内で選択された項目のインデックスの配列。(<i>FlexChartSeries</i> から継承されました)
<i>selectedItem</i>	系列内で選択されたグラフ項目。(<i>FlexChartSeries</i> から継承されました)
<i>selectedItems</i>	系列内で選択されたグラフ項目の配列。(<i>FlexChartSeries</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は <code>Nothing</code> です。(<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は <code>true</code> です。 <code>useHandCursor</code> が <code>false</code> の場合、代わりに矢印カーソルが表示されます。(<i>FlexObject</i> から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>xField</i>	要素の x 軸の位置を決定するデータ プロバイダーのフィールド。
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインターを動かしたインデックスです。(FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexImage クラス

説明

実行時に、JPEG、PNG、GIF および SWF のファイルをインポートできます。

@Embed(source='filename') を使用して、これらのファイルと SVG ファイルをコンパイル時に埋め込むこともできます。

メモ： Flex には、Flex アプリケーションをロードする SWFLoader コントロールもあります。静止グラフィックファイルと SWF ファイルのロードには、通常 Image コントロールを使用し、Flex アプリケーションのロードには、SWFLoader コントロールを使用します。Image コントロールは、カスタム項目レングラと項目エディターでも使用できるように設計されています。

継承階層

- [FlexLoader](#)
 - FlexImage

構文

```
'Declaration  
Public Class FlexImage _  
Inherits FlexLoader
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)

名前	説明
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。（ FlexObject から継承されました）
scaleX	水平方向の倍率を指定する係数。（ FlexObject から継承されました）
scaleY	垂直方向の倍率を指定する係数。（ FlexObject から継承されました）
source	コンテンツとしてロードする URL、オブジェクト、クラス、または暮らすの文字列名。（ FlexLoader から継承されました）
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。（ FlexDisplayObject から継承されました）
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。（ FlexDisplayObject から継承されました）
tabIndex	SWF ファイル内のオブジェクトのタブ順序。（ FlexDisplayObject から継承されました）
Text	コントロールのテキスト。（ TestObject から継承されました）
themeColor	コンポーネントのテーマ色。（ FlexObject から継承されました）
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。（ FlexObject から継承されました）
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。（ FlexObject から継承されました）
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。（ TestObject から継承されました）
visible	表示オブジェクトが可視かどうか。（ FlexDisplayObject から継承されました）
width	表示オブジェクトの幅（ピクセル単位）。（ FlexDisplayObject から継承されました）
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。（ FlexDisplayObject から継承されました）
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。（ FlexDisplayObject から継承されました）

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexLabel クラス

説明

編集できないテキストの単一行を表示します。

継承階層

- [FlexObject](#)
 - FlexLabel

構文

```
'Declaration  
Public Class FlexLabel _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
disabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
fontFamily	使用するフォントの名前。
fontSize	テキストの高さ (ピクセル単位)。

名前	説明
<i>fontStyle</i>	テキストがイタリック フォントかどうか。
<i>fontWeight</i>	テキストがボールド体かどうか。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>htmlText</i>	Label コントロールによって表示されるテキスト (そのテキストのスタイルを表現する HTML マークアップを含む)。
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>text</i>	このコントロールで表示されるプレーン テキスト。 その外観は、この Label コントロールの CSS スタイルによって決定されます。
<i>textAlign</i>	コンテナ内のテキストの配置。
<i>textDecoration</i>	テキストに下線を表示するかどうか。
<i>textIndent</i>	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。 デフォルト値は 0 です。

名前	説明
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexLineChart クラス

説明

連続する 1 本の線でつながれた点としてデータ系列を表します。

アイコンやシンボルを使用して線上の各データ ポイントを表すことも、アイコンなしの単純な線を表示することもできます。

継承階層

- [FlexCartesianChart](#)
 - FlexLineChart

構文

```
'Declaration
Public Class FlexLineChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>clipContent</i>	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipMode</i>	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
<i>description</i>	グラフのデータの簡単な説明。(FlexChart から継承されました)
<i>dragEnabled</i>	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
<i>dropEnabled</i>	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChart から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexChart から継承されました)
<i>fontSize</i>	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。(FlexChart から継承されました)

名前	説明
gutterBottom	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterLeft	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterRight	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterTop	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
horizontalAxisRatio	水平軸の高さの制限。(FlexCartesianChart から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChart から継承されました)
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。(FlexChart から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)

名前	説明
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。 (FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。 (FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。 (FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。 (FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。 (FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。 (FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。 (FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。 (TestObject から継承されました)

名前	説明
verticalAxisRatio	垂直軸の幅の制限。(FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexLineSeries クラス

説明

LineChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexLineSeries

構文

```
'Declaration
Public Class FlexLineSeries _
Inherits FlexChartSeries
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。 (<i>FlexDisplayObject</i> から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>FlexChartSeries</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexChartSeries</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>form</i>	グラフの線のタイプ。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexChartSeries</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <i>mouseEnabled</i> を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <i>mouseEnabled</i> 動作を変更するには、 <i>flash.display.DisplayObjectContainer.mouseChildren</i> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)

名前	説明
numColumns	利用可能なデータ中の列の合計数。(FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChartSeries から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
radius	データ ポイントのグラフ要素の半径 (ピクセル単位)。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。(FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
sortOnXField	レンダリングの前に線データポイントを左から右にソートするかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
xField	各データ ポイントの x 軸の位置を決定するデータ プロバイダーのフィールド。
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)
yField	各データ ポイントの y 軸の位置を決定するデータ プロバイダーのフィールド。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインターを動かしたインデックスです。(FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexLinkBar クラス

説明

一連のリンク先を指定する LinkButton コントロールの水平方向または垂直方向の行を定義します。

LinkBar コントロールは通常、ViewStack コンテナのアクティブな子コンテナを制御するため、またはリンクの単独のセットを作成するために使用します。

継承階層

- [FlexNavigationBar](#)
 - FlexLinkBar

構文

```
'Declaration
Public Class FlexLinkBar _
Inherits FlexNavigationBar
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
direction	垂直レイアウト (デフォルト)、水平レイアウトのどちらを使用するか。(FlexBox から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)

名前	説明
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalAlign</i>	このコンテナ内の子の水平方向の配置。 (<i>FlexBox</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は

名前	説明
	NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	アクティブなナビゲーション項目のインデックス。最初の項目のインデックスは 0 です。デフォルト値は -1 です。(FlexNavigationBar から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAlign	このコンテナ内の子の垂直方向の配置。(FlexBox から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Change	itemClick イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexNavigationBar から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)

名前	説明
getParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。 (FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexNavigationBar から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexList クラス

説明

項目のリストを水平方向、垂直方向、またはタイル状に並べて表示します。

継承階層

- [FlexListBase](#)
 - FlexList
 - [FlexTree](#)

構文

```
'Declaration
Public Class FlexList _
Inherits FlexListBase
```

プロパティ

名前	説明
allowMultipleSelection	1 つ以上の項目を同時に選択することが許可されているかどうか。(FlexListBase から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
<i>alternatingItemColors</i>	リストの項目の背景に使用する色。(FlexListBase から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。(FlexListBase から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnCount</i>	TileList コントロールまたは HorizontalList コントロールの項目に表示される列の数。(FlexListBase から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexListBase から継承されました)
<i>columnWidth</i>	コントロールの列の幅。(FlexListBase から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipField</i>	データ ヒントとして表示するデータ プロバイダー項目内のフィールドの名前。(FlexListBase から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexListBase から継承されました)
<i>focusAlpha</i>	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。(FlexListBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>iconField</i>	アイコンとして表示するものを決定する、データ プロバイダー オブジェクト内のフィールドの名前。(FlexListBase から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。(FlexListBase から継承されました)

名前	説明
lastVisibleRow	最後の可視の子のインデックス。(FlexListBase から継承されました)
lockedColumnCount	スクロールするコントロールの最初の列のインデックス。(FlexListBase から継承されました)
lockedRowCount	スクロールするコントロールの最初の行のインデックス。(FlexListBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexListBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexListBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
rollOverColor	ユーザーがレンダラの上にマウスを置いたときのレンダラの背景色。デフォルト値は 0xEEFFEE6 です。(FlexListBase から継承されました)
rowCount	表示する行の数。(FlexListBase から継承されました)
rowHeight	行の高さ (ピクセル単位)。(FlexListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectable	選択された項目を選択状態でリストに表示するかどうか。(FlexListBase から継承されました)
selectedIndex	データ プロバイダー内で選択された項目のインデックス。(FlexListBase から継承されました)
selectedIndices	データ プロバイダーで選択された項目のインデックスの配列。(FlexListBase から継承されました)

名前	説明
selectedItem	データプロバイダー内で選択された項目への参照。(FlexListBase から継承されました)
selectedItems	データプロバイダーで選択された項目への参照の配列。(FlexListBase から継承されました)
selectionColor	ユーザーが選択したときのレンダラの背景色。デフォルト値は 0x7FCEFF です。(FlexListBase から継承されました)
selectionDisabledColor	コンポーネントが無効化されたときのレンダラの背景色。デフォルト値は 0xDDDDDD です。(FlexListBase から継承されました)
showDataTips	行内のテキストに対して dataTips を表示するかどうか。(FlexListBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textRollOverColor	マウスポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。(FlexListBase から継承されました)
textSelectedColor	ユーザーが選択したときのレンダラのテキスト色。デフォルト値は 0x2B333C です。(FlexListBase から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
useRollOver	項目の上にマウスを置いたときに、その項目をハイライト表示するかどうか。(FlexListBase から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
variableRowHeight	個々の行が別々の高さを持つことができるかどうか。(FlexListBase から継承されました)
verticalAlign	行のレンダラの垂直方向の配置。(FlexListBase から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)

名前	説明
wordWrap	行内のテキストを折り返すかどうか。(FlexListBase から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Deselect	以前に選択した項目の選択を解除するときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClick	コントロールの可視項目上でマウスがダブルクリックされたことを示す itemDoubleClick イベントにおける ListEvent オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
Edit	項目が編集可能な状態かどうかを示す itemEditBegin イベントにおけるイベント オブジェクトの type プロパティの値を定義します。
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
Exists	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Find	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexListBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
MultiSelect	複数項目を選択するアクションの一部として項目が選択されたときにディスパッチされるイベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
ScrollToIndex	指定したインデックス位置にあるデータ プロバイダー項目を表示状態にします。 (FlexListBase から継承されました)
Select	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
SelectIndex	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (FlexListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexListBase クラス

説明

ListBase クラスは、項目のリストを表すコントロールの基本クラスです。項目リストでは、1 つまたは複数の項目を選択したり、項目をスクロールしたりできます。

項目は、dataProvider プロパティを使用して指定され、項目レンダラによって表示されます。

モデル/ビュー アーキテクチャでは、ListBase から派生したクラスはビューを表し、dataProvider オブジェクトはモデルを表します。

継承階層

- [FlexScrollBase](#)
 - FlexListBase
 - [FlexAdvancedDataGrid](#)
 - [FlexDataGrid](#)
 - [FlexList](#)

構文

```
'Declaration
Public Class FlexListBase _
Inherits FlexScrollBase
```

プロパティ

名前	説明
allowMultipleSelection	1 つ以上の項目を同時に選択することが許可されているかどうか。
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alternatingItemColors	リストの項目の背景に使用する色。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合のテキストの色。 デフォルト値は 0xAAB3B3 です。
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)

名前	説明
<i>columnCount</i>	TileList コントロールまたは HorizontalList コントロールの項目に表示される列の数。
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。
<i>columnWidth</i>	コントロールの列の幅。
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipField</i>	データ ヒントとして表示するデータ プロバイダー項目内のフィールドの名前。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。
<i>focusAlpha</i>	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>iconField</i>	アイコンとして表示するものを決定する、データ プロバイダー オブジェクト内のフィールドの名前。
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>lockedColumnCount</i>	スクロールするコントロールの最初の列のインデックス。
<i>lockedRowCount</i>	スクロールするコントロールの最初の行のインデックス。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。

名前	説明
numRows	利用可能なデータ中の行の合計数。
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。（ FlexObject から継承されました）
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。（ FlexObject から継承されました）
rollOverColor	ユーザーがレンダラの上にマウスを置いたときのレンダラの背景色。デフォルト値は 0xEEEEEE です。
rowCount	表示する行の数。
rowHeight	行の高さ（ピクセル単位）。
scaleX	水平方向の倍率を指定する係数。（ FlexObject から継承されました）
scaleY	垂直方向の倍率を指定する係数。（ FlexObject から継承されました）
selectable	選択された項目を選択状態でリストに表示するかどうか。
selectedIndex	データプロバイダー内で選択された項目のインデックス。
selectedIndices	データプロバイダーで選択された項目のインデックスの配列。
selectedItem	データプロバイダー内で選択された項目への参照。
selectedItems	データプロバイダーで選択された項目への参照の配列。
selectionColor	ユーザーが選択したときのレンダラの背景色。デフォルト値は 0x7FCEFF です。
selectionDisabledColor	コンポーネントが無効化されたときのレンダラの背景色。デフォルト値は 0xDDDDDD です。
showDataTips	行内のテキストに対して dataTips を表示するかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。（ FlexDisplayObject から継承されました）
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。（ FlexDisplayObject から継承されました）
tabIndex	SWF ファイル内のオブジェクトのタブ順序。（ FlexDisplayObject から継承されました）
Text	コントロールのテキスト。（ TestObject から継承されました）
textRollOverColor	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。
textSelectedColor	ユーザーが選択したときのレンダラのテキスト色。デフォルト値は 0x2B333C です。
themeColor	コンポーネントのテーマ色。（ FlexObject から継承されました）
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。（ FlexObject から継承されました）

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
useRollOver	項目の上にマウスを置いたときに、その項目をハイライト表示するかどうか。
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
variableRowHeight	個々の行が別々の高さを持つことができるかどうか。
verticalAlign	行のレンダラの垂直方向の配置。
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
wordWrap	行内のテキストを折り返すかどうか。
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Deselect	以前に選択した項目の選択を解除するときにディスパッチされるイベントの、イベントオブジェクトの type プロパティの値を定義します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleSelect	コントロールの可視項目上でマウスがダブルクリックされたことを示す itemDoubleClick イベントにおける ListEvent オブジェクトの type プロパティの値を定義します。

名前	説明
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
MultiSelect	複数項目を選択するアクションの一部として項目が選択されたときにディスパッチされるイベントにおけるイベント オブジェクトの type プロパティの値を定義します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。
ScrollToIndex	指定したインデックス位置にあるデータ プロバイダー項目を表示状態にします。
Select	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。

名前	説明
SelectIndex	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexListLabel クラス

説明

コントロールのデフォルトの項目レンダラを定義します。 デフォルトでは、項目レンダラはリスト中の各項目に関連付けられたテキストとオプションでアイコンを描画します。

継承階層

- [FlexDisplayObject](#)
 - FlexListLabel

構文

```
'Declaration
Public Class FlexListLabel _
Inherits FlexDisplayObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。
automationName	このオブジェクトの識別子として使用することのできる名前。
automationValue	オブジェクトのレンダリング表示に対応する値。また、アプリケーションでの視覚表示と同じにするために、識別子とオブジェクトを関連付けるのに使用できます。
className	"Button" のようなこのインスタンスのクラスの名前。
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウ

名前	説明
	ス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>x</code> 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (<code>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps</code>)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
WaitForObject	<p>タイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexLoader クラス

説明

指定した SWF ファイルをロードし表示します。通常、ある Flex アプリケーションをホストする Flex アプリケーションにロードする場合に `SWFLoader` を使用します。

継承階層

- [FlexObject](#)
 - FlexLoader
 - [FlexImage](#)

構文

```
'Declaration
Public Class FlexLoader _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)

名前	説明
source	コンテンツとしてロードする URL、オブジェクト、クラス、または暮らすの文字列名。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(<i>FlexObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(<i>FlexObject</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ToString</i>	コントロールの文字列表現を返します。(<i>FlexDisplayObject</i> から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

FlexMenu クラス

説明

大部分のソフトウェア アプリケーションに見られるファイル メニューや編集メニューのような、個別に選択可能な選択肢を含むポップアップ メニューを作成します。 ポップアップ メニューには、必要に応じて複数のサブメニューのレベルを設定できます。

継承階層

- [FlexObject](#)

- FlexMenu

構文

```
'Declaration
Public Class FlexMenu _
Inherits FlexObject
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>alternatingItemColors</i>	メニューまたはサブメニューのメニュー項目に使用する背景色の代替パターン配列。
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべて

名前	説明
	の子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。
<code>numRows</code>	利用可能なデータ中の行の合計数。
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<code>rollOverColor</code>	ユーザーがメニュー項目の上にマウスを移動したときのメニュー項目の背景色。デフォルト値は 0xB2E1FF です。
<code>rowCount</code>	表示する行の数。
<code>scaleX</code>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<code>selectedIndex</code>	データプロバイダー内で選択された項目のインデックス。
<code>selectionColor</code>	メニュー項目が選択されたときのメニュー項目の背景色。デフォルト値は 0x7FCEFF です。
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<code>Text</code>	コントロールのテキスト。(TestObject から継承されました)
<code>textRollOverColor</code>	ユーザーがメニュー項目の上にマウスを移動したときのメニュー項目のテキスト色。デフォルト値は 0x2B333C です。
<code>textSelectedColor</code>	メニュー項目が選択されたときのメニュー項目のテキスト色。デフォルト値は 0x2B333C です。
<code>themeColor</code>	コンポーネントのテーマ色。(FlexObject から継承されました)
<code>toolTip</code>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<code>useHandCursor</code>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
Hide	メニューまたはサブメニューが閉じたときにディスパッチされます。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プ

名前	説明
	ロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	メニュー項目が選択された場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Show	メニューまたはサブメニューが開いたときにディスパッチされます。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexMenuBar クラス

説明

1 つ以上のメニューを含む水平な最上位メニューバーを定義します。最上位メニュー項目をクリックすると、Menu コントロールのインスタンスであるポップアップサブメニューが開きます。

継承階層

- [FlexObject](#)
 - FlexMenuBar

構文

```
'Declaration
Public Class FlexMenuBar _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>color</i>	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>disabledColor</i>	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。
<i>firstVisibleRow</i>	最初の可視の子のインデックス。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。
<i>fontSize</i>	テキストの高さ (ピクセル単位)。
<i>fontStyle</i>	テキストがイタリック フォントかどうか。
<i>fontWeight</i>	テキストがボールド体かどうか。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべて

名前	説明
	の子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このオブジェクトの子の数。
<i>numColumns</i>	利用可能なデータ中の列の合計数。
<i>numRows</i>	利用可能なデータ中の行の合計数。
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<i>rollOverColor</i>	ユーザーがメニュー項目の上にマウスを移動したときのメニュー項目の背景色。
<i>scaleX</i>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>selectedIndex</i>	現在開いている Menu、または現在開いている Menu がない場合は最後に開いていた Menu の MenuBar コントロールのインデックス。
<i>selectionColor</i>	メニュー項目が選択されたときのメニュー項目の背景色。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>textAlign</i>	コンテナ内のテキストの配置。
<i>textDecoration</i>	テキストに下線を表示するかどうか。
<i>textIndent</i>	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。
<i>themeColor</i>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。 <code>useHandCursor</code> が false の

名前	説明
	場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
Hide	メニューまたはサブメニューが閉じたときにディスパッチされます。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Show	メニューまたはサブメニューが開いたときにディスパッチされます。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定し

名前	説明
WaitForDisappearance	<p>ない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexNavigationBar クラス

説明

LinkBar コントロールや TabBar コントロールなどのナビゲーター コントロールに対するスーパークラスです。このクラスは直接インスタンス化できません。

継承階層

- [FlexBox](#)
 - FlexNavigationBar
 - [FlexButtonBar](#)
 - [FlexLinkBar](#)

構文

```
'Declaration
Public Class FlexNavigationBar _
Inherits FlexBox
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。(FlexContainer から継承されました)
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>direction</i>	垂直レイアウト (デフォルト)、水平レイアウトのどちらを使用するか。(FlexBox から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)

名前	説明
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalAlign</i>	このコンテナ内の子の水平方向の配置。 (<i>FlexBox</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は

名前	説明
	NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	アクティブなナビゲーション項目のインデックス。最初の項目のインデックスは 0 です。デフォルト値は -1 です。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAlign	このコンテナ内の子の垂直方向の配置。(FlexBox から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Change	<p>itemClick イベントのイベント オブジェクトの type プロパティの値を定義します。</p>
ChangeFocus	<p>現在のフォーカスを変更します。 (FlexObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
DragCancel	<p>dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)</p>
DragDrop	<p>dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)</p>
DragStart	<p>dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexContainer から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
	ムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexNumericStepper クラス

説明

順序化された集合から数値を選択することができます。

NumericStepper コントロールは、単一行のテキスト入力フィールドと、使用できる値を増減するための 2 つの矢印ボタンから構成されます。上矢印キーと下矢印キーを使用して増減する値を循環させることもできます。

継承階層

- [FlexObject](#)
 - FlexNumericStepper

構文

```
'Declaration
Public Class FlexNumericStepper _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<code>className</code>	<code>Button</code> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<code>color</code>	コンポーネント ラベルを含むコンポーネントのテキストの色。 デフォルト値は 0x0B333C です。
<code>currentState</code>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<code>disabledColor</code>	コンポーネントが無効な場合のテキストの色。 デフォルト値は 0xAAB3B3 です。
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>fillColors</code>	コントロールの背景を濃淡色化するのに使用する色。 平面的なコントロールには同じ色を両方の値として渡します。 デフォルト値は [0xFFFFFFFF, 0xC0C0C0] です。
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>fontFamily</code>	使用するフォントの名前。
<code>fontSize</code>	テキストの高さ (ピクセル単位)。
<code>fontStyle</code>	テキストがイタリック フォントかどうか。
<code>fontWeight</code>	テキストがボールド体かどうか。 有効な値は、normal および bold です。
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>maximum</code>	NumericStepper の最大値。
<code>minimum</code>	NumericStepper の最小値。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>nextValue</code>	<code>value</code> プロパティの現在の値の 1 つ後の値。 <code>maximum</code> プロパティの値より大きくなることはありません。
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)

名前	説明
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
previousValue	value プロパティの現在値の 1 つ前の値。maximum プロパティの値より小さくなることはありません。
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
stepSize	値間の 0 でない変化単位。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。有効な値は left、right または center です。
textDecoration	テキストに下線を表示するかどうか。有効な値は none と underline です。デフォルト値は none です。
textIndent	テキストの 1 行目のコンテナ左端からのオフセット（ピクセル単位）。デフォルト値は 0 です。
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。 (TestObject から継承されました)
value	NumericStepper コントロールのテキスト領域に表示される値。
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。 (FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Change</i>	ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。 (<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。 (<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (<i>TestObject</i> から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Input	テキスト イベントについての情報を含むイベント オブジェクトを作成します。 イベント オブジェクトは、パラメーターとしてイベント リスナーに渡されます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SelectText	テキスト フィールド内の選択範囲を追跡します。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexObject クラス

説明

対話型と非対話型の両方のすべてのビジュアル コンポーネントの基本クラスです。

継承階層

- [FlexDisplayObject](#)
 - FlexObject
 - [FlexAxisRenderer](#)
 - [FlexButton](#)
 - [FlexChart](#)
 - [FlexChartLegend](#)
 - [FlexChartSeries](#)
 - [FlexComboBase](#)
 - [FlexContainer](#)
 - [FlexDateChooser](#)
 - [FlexLabel](#)
 - [FlexLoader](#)
 - [FlexMenu](#)
 - [FlexMenuBar](#)
 - [FlexNumericStepper](#)
 - [FlexProgressBar](#)
 - [FlexRule](#)
 - [FlexScrollBar](#)
 - [FlexScrollBase](#)
 - [FlexSlider](#)

- [FlexVideoDisplay](#)
- [SparkDataGridLabel](#)
- [SparkGroupBase](#)
- [SparkObject](#)
- [SparkPopUpAnchor](#)
- [SparkTextBase](#)
- [SparkVideoDisplay](#)

構文

```
'Declaration
Public Class FlexObject _
Inherits FlexDisplayObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。
automationName	このオブジェクトの識別子として使用することのできる名前。
className	<i>Button</i> のようなこのインスタンスのクラスの名前。
currentState	コンポーネントの現在の表示状態。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。
errorColor	検証が失敗したときにコンポーネントをハイライトする色。
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、 flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は

名前	説明
	NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。
scaleX	水平方向の倍率を指定する係数。
scaleY	垂直方向の倍率を指定する係数。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexOLAPDataGrid クラス

説明

AdvancedDataGrid コントロールの機能を拡張したもので、OLAP クエリの結果表示のサポートが追加されています。

すべての Flex データ グリッド コントロールと同様、OLAPDataGrid はデータを行と列の 2 次元で表現するように設計されています。

継承階層

- [FlexAdvancedDataGrid](#)

- FlexOLAPDataGrid

構文

```
'Declaration
Public Class FlexOLAPDataGrid _
Inherits FlexAdvancedDataGrid
```

プロパティ

名前	説明
allowDragSelection	ドラッグによる選択が有効かどうか。通常の選択では、選択する項目上にマウスを移動させてもマウス ボタンを押すことはできませんが、ドラッグによる選択では、項目にドラッグすることによって、その項目を選択できます。(FlexAdvancedDataGrid から継承されました)
allowMultipleSelection	1 つ以上の項目を同時に選択することが許可されているかどうか。(FlexListBase から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alternatingItemColors	リストの項目の背景に使用する色。(FlexListBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。(FlexListBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnCount	TileList コントロールまたは HorizontalList コントロールの項目に表示される列の数。(FlexListBase から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexListBase から継承されました)
columnWidth	コントロールの列の幅。(FlexListBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dataTipField	データ ヒントとして表示するデータ プロバイダー項目内のフィールドの名前。(FlexListBase から継承されました)
defaultCellString	IOLAPResult インスタンスから返されたセルのデータが Nothing または NaN の場合に、そのセルに表示される文字列。
depthColors	AdvancedDataGrid コントロールのナビゲーション ツリーの各レベルの行で使用する色の配列 (降順)。デフォルト値は未定義です。(FlexAdvancedDataGrid から継承されました)

名前	説明
<i>disabledColor</i>	コンポーネントが無効な場合のテキストの色。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>displayItemsExpanded</i>	ナビゲーション ツリーが展開されてすべての項目が表示されているかどうか。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>editable</i>	データ プロバイダーの項目を編集できるかどうか。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>focusAlpha</i>	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。 (<i>FlexListBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>headerColors</i>	ヘッダーの背景にグラデーションを描画するために使用する 2 色の配列。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>headerHeight</i>	列のヘッダー セルの高さ (ピクセル単位)。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalGridLineColor</i>	水平グリッド線の色。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>iconField</i>	アイコンとして表示するものを決定する、データ プロバイダー オブジェクト内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>indentation</i>	ナビゲーション ツリーの各ノードのインデント (ピクセル単位)。デフォルト値は 17 です。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>lockedColumnCount</i>	スクロールするコントロールの最初の列のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>lockedRowCount</i>	スクロールするコントロールの最初の行のインデックス。 (<i>FlexListBase</i> から継承されました)

名前	説明
minColumnWidth	列の最小幅 (ピクセル単位)。 (FlexAdvancedDataGrid から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (FlexListBase から継承されました)
numRows	利用可能なデータ中の行の合計数。 (FlexListBase から継承されました)
openDuration	ナビゲーション ツリーの開いた状態や閉じた状態への遷移にかかる時間の長さ (ミリ秒単位)。 デフォルト値は 250 です。 (FlexAdvancedDataGrid から継承されました)
paddingBottom	行の下端と行内のレンダラの下端との間のピクセル数。 (FlexAdvancedDataGrid から継承されました)
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。 デフォルト値は 2 です。 (FlexAdvancedDataGrid から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。 デフォルト値は 0 です。 (FlexAdvancedDataGrid から継承されました)
paddingTop	行の上端と行内のレンダラの上端との間のピクセル数。 (FlexAdvancedDataGrid から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
resizableColumns	列のサイズをユーザーが変更できるかどうか。 (FlexAdvancedDataGrid から継承されました)
rollOverColor	ユーザーがレンダラの上にマウスを置いたときのレンダラの背景色。 デフォルト値は 0xEEFFEE6 です。 (FlexListBase から継承されました)
rowCount	表示する行の数。 (FlexListBase から継承されました)
rowHeight	行の高さ (ピクセル単位)。 (FlexListBase から継承されました)

名前	説明
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectable	選択された項目を選択状態でリストに表示するかどうか。(FlexListBase から継承されました)
selectedCells	行インデックスおよび列インデックスとしてセル位置の配列。(FlexAdvancedDataGrid から継承されました)
selectedIndex	データプロバイダー内で選択された項目のインデックス。(FlexListBase から継承されました)
selectedIndices	データプロバイダーで選択された項目のインデックスの配列。(FlexListBase から継承されました)
selectedItem	データプロバイダー内で選択された項目への参照。(FlexListBase から継承されました)
selectedItems	データプロバイダーで選択された項目への参照の配列。(FlexListBase から継承されました)
selectionColor	ユーザーが選択したときのレンダラの背景色。デフォルト値は 0x7FCEFF です。(FlexListBase から継承されました)
selectionDisabledColor	コンポーネントが無効化されたときのレンダラの背景色。デフォルト値は 0xDDDDDD です。(FlexListBase から継承されました)
selectionMode	コントロールの選択モード。(FlexAdvancedDataGrid から継承されました)
showDataTips	行内のテキストに対して dataTips を表示するかどうか。(FlexListBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textRollOverColor	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。(FlexListBase から継承されました)
textSelectedColor	ユーザーが選択したときのレンダラのテキスト色。デフォルト値は 0x2B333C です。(FlexListBase から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の

名前	説明
	場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>useRollOver</i>	項目の上にマウスを置いたときに、その項目をハイライト表示するかどうか。(FlexListBase から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>variableRowHeight</i>	個々の行が別々の高さを持つことができるかどうか。(FlexListBase から継承されました)
<i>verticalAlign</i>	行のレンダラの垂直方向の配置。(FlexListBase から継承されました)
<i>verticalGridLineColor</i>	垂直グリッド線の色。デフォルト値は 0x666666 です。(FlexAdvancedDataGrid から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>wordWrap</i>	行内のテキストを折り返すかどうか。(FlexListBase から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(FlexObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>Close</i>	AdvancedDataGrid の分岐を閉じるまたは折りたたみます。(FlexAdvancedDataGrid から継承されました)

名前	説明
ColumnGroupedADGHeaderShift	ユーザーが列ヘッダーを新しい場所にドラッグし、マウス ボタンを離れたときにディスパッチされます。その結果、その列は新しいインデックスに移されます。(FlexAdvancedDataGrid から継承されました)
ColumnStretch	ユーザーが列の幅を変更したときにディスパッチされます。これは、列に表示されるデータ量が変更されたことを意味します。(FlexAdvancedDataGrid から継承されました)
Deselect	以前に選択した項目の選択を解除するときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClick	コントロールの可視項目上でマウスがダブルクリックされたことを示す itemDoubleClick イベントにおける ListEvent オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexListBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetCellData	グリッド内のセルのデータを返します。(FlexAdvancedDataGrid から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetGroupedItemChildrenCount	グループの最初の項目内にある子の数を返します。 (FlexAdvancedDataGrid から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetRowData	データ プロバイダーの項目の行のデータを返します。 (FlexAdvancedDataGrid から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexListBase から継承されました)
HeaderClick	列の内容に基づいてグリッドの内容の並べ替えを行うために、ユーザーが列ヘッダーでマウス ボタンを離れたときにディスパッチされます。(FlexAdvancedDataGrid から継承されました)
HeaderShift	ユーザーが列ヘッダーを新しい場所にドラッグし、マウス ボタンを離れたときにディスパッチされます。その結果、その列は新しいインデックスに移されます。(FlexAdvancedDataGrid から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
<i>IsGroupedItem</i>	項目がグループのメンバーかどうかを指定する値を返します。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>MouseScroll</i>	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (<i>FlexScrollBase</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (<i>FlexObject</i> から継承されました)
<i>MultiSelect</i>	複数項目を選択するアクションの一部として項目が選択されたときにディスパッチされるイベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (<i>FlexListBase</i> から継承されました)
<i>Open</i>	AdvancedDataGrid の分岐を開くまたは展開します。 (<i>FlexAdvancedDataGrid</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Scroll</i>	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。 (<i>FlexListBase</i> から継承されました)
<i>ScrollToIndex</i>	指定したインデックス位置にあるデータ プロバイダー項目を表示状態にします。 (<i>FlexListBase</i> から継承されました)
<i>Select</i>	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (<i>FlexListBase</i> から継承されました)
<i>SelectIndex</i>	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (<i>FlexListBase</i> から継承されました)
<i>SetFocus</i>	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (<i>FlexObject</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexPanel クラス

説明

タイトルバー、キャプション、境界線、および子のコンテンツ領域から構成されています。

通常、Panel コンテナは最上位のアプリケーション モジュールをラップするために使用します。例えば、Panel コンテナにショッピング カートを含めることもできます。

継承階層

- [FlexContainer](#)
 - FlexPanel
 - [FlexAlert](#)
 - [FlexTitleWindow](#)

構文

```
'Declaration
Public Class FlexPanel _
Inherits FlexContainer _
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)

名前	説明
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>FlexContainer</i> から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。 (<i>FlexContainer</i> から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。 (<i>FlexContainer</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>dropShadowEnabled</i>	Panel コンテナのドロップ シャドウが可視かどうか。 デフォルト値は true です。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。 (<i>FlexContainer</i> から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。 (<i>FlexContainer</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexContainer</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexContainer</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexContainer</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexContainer</i> から継承されました)
<i>footerColors</i>	フッター (ControlBar コンテナの領域) の背景の描画に使用される 2 色の配列。
<i>headerColors</i>	ヘッダーを描画するために使用する 2 色の配列。
<i>headerHeight</i>	ヘッダーの高さ。 デフォルト値はタイトル テキストのスタイルに基づいて計算されます。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)

名前	説明
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。(FlexContainer から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>status</i>	タイトル バーのステータス領域に表示されるテキスト。デフォルト値は "" です。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>title</i>	タイトル バーに表示されるタイトルまたはキャプション。デフォルト値は "" です。

名前	説明
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>GetValues</i>	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(<i>FlexContainer</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。 (FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexPieChart クラス

説明

標準的な円グラフでデータ系列を表します。

円グラフ内の各区分の大きさは、データ プロバイダーのデータにより、他の区分との関連で決定されます。PieSeries クラスを使用すると、標準的な円グラフ、ドーナツ グラフ、および積み上げ円グラフを作成できます。

継承階層

- [FlexChart](#)
 - FlexPieChart

構文

```
'Declaration  
Public Class FlexPieChart _  
Inherits FlexChart
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipContent	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dataTipMode	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
description	グラフのデータの簡単な説明。(FlexChart から継承されました)
dragEnabled	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
dropEnabled	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexChart から継承されました)

名前	説明
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexChart</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。 (<i>FlexChart</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>innerRadius</i>	円グラフの中心の空洞のサイズ。
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexChart</i> から継承されました)
<i>maximumDataTipCount</i>	グラフが表示するデータ ヒントの最大数。 (<i>FlexChart</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>mouseSensitivity</i>	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。 (<i>FlexChart</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexChart</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexChart</i> から継承されました)
<i>paddingBottom</i>	グラフの下境界線とコンテンツ領域との間にあるピクセル数。 (<i>FlexChart</i> から継承されました)
<i>paddingLeft</i>	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。 (<i>FlexChart</i> から継承されました)
<i>paddingRight</i>	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。 (<i>FlexChart</i> から継承されました)
<i>paddingTop</i>	グラフの上境界線とコンテンツ領域との間にあるピクセル数。 (<i>FlexChart</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)

名前	説明
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。(FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナー内のテキストの配置。
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。(FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>DragCancel</i>	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。 (<i>FlexChart</i> から継承されました)
<i>DragDrop</i>	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。 (<i>FlexChart</i> から継承されました)
<i>DragStart</i>	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。 (<i>FlexChart</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。 (<i>TestObject</i> から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChart から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プ

名前	説明
	ロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
	<p>アウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexPieSeries クラス

説明

PieChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexPieSeries

構文

```
'Declaration
Public Class FlexPieSeries _
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)

名前	説明
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
calloutGap	引き出し線をレンダリングする際に円グラフの縁とラベルの間にどの程度のスペースが挿入されるか。 デフォルト値は 10 です。
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。 (FlexChartSeries から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
explodeRadius	0 から 1 までの数値。 円系列の全区分をグラフの中心からどのくらい分離するかを全半径のパーセンテージで指定します。
field	PieChart コントロールの各区分のデータを決定するデータ プロバイダーのフィールド。 デフォルト値は <code>Nothing</code> です。
firstVisibleRow	最初の可視の子のインデックス。 (FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
fontSize	テキストの高さ (ピクセル単位)。
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
innerRadius	系列の中心からレンダリング済み区分の内側エッジまでの距離を、系列に割り当てられた全半径のパーセンテージで指定する、0 ~ 1 の数。
insideLabelSizeLimit	フォントサイズのしきい値 (ポイント)。 これ以下のサイズの内部ラベルは判読不能と見なされます。
labelPosition	値のラベルのレンダリング方法。
lastVisibleRow	最後の可視の子のインデックス。 (FlexChartSeries から継承されました)
maxLabelRadius	ラベルに割り当て可能な PieSeries の半径の最大値。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は <code>true</code> であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味しま

名前	説明
	す。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
nameField	PieChart コントロールの各区分の名前を決定するデータ プロバイダーのフィールド。デフォルト値は Nothing です。
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChartSeries から継承されました)
outerRadius	領域全体のうち PieSeries が系列の内容をレンダリングする場合に使用可能な領域の割合。
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。(FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
startAngle	PieChart コントロールの最初の区分の開始角度。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。 (FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインタを動かしたインデックスです。 (FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

FlexPlotChart クラス

説明

各データ ポイントにつき 2 つの値でデータを表します。 1 つの値は水平方向の軸に沿ったデータ ポイントの位置を、もう 1 つの値は垂直方向の軸に沿った位置を決定します。

継承階層

- [FlexCartesianChart](#)

- FlexPlotChart

構文

```
'Declaration
Public Class FlexPlotChart _
Inherits FlexCartesianChart
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>clipContent</i>	Flex が軸によって境界付けられた領域でグラフをクリップするかどうか。グラフをクリップする場合には false を設定します。true に設定すると、データがレンダリングされるときにクリップされません。(FlexChart から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexChart から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dataTipMode</i>	グラフの DataTip コントロールの表示方法。(FlexChart から継承されました)
<i>description</i>	グラフのデータの簡単な説明。(FlexChart から継承されました)
<i>dragEnabled</i>	このグラフの外に項目をドラッグして他のコントロールにドロップすることができるかどうか。(FlexChart から継承されました)
<i>dropEnabled</i>	ドラッグした項目をグラフ上にドロップできるかどうか。(FlexChart から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexChart から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)

名前	説明
fontFamily	使用するフォントの名前。(FlexChart から継承されました)
fontSize	テキストの高さ(ピクセル単位)。ColorPicker コントロール以外のすべてのコントロールのデフォルト値は 10 です。ColorPicker コントロールのデフォルト値は 11 です。(FlexChart から継承されました)
gutterBottom	グラフのデータ領域の下端とグラフ コントロールの下端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterLeft	グラフのデータ領域の左端とグラフ コントロールの左端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterRight	グラフのデータ領域の右端とグラフ コントロールの外側との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
gutterTop	グラフのデータ領域の上端とグラフ コントロールの上端との間の領域のサイズ (ピクセル単位)。(FlexCartesianChart から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
horizontalAxisRatio	水平軸の高さの制限。(FlexCartesianChart から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChart から継承されました)
maximumDataTipCount	グラフが表示するデータ ヒントの最大数。(FlexChart から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseSensitivity	マウス ポインターをグラフ上で移動させたときに、データ ポイントがマウス ポインターの下にあると Flex で判断するための距離 (ピクセル単位)。(FlexChart から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChart から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChart から継承されました)
paddingBottom	グラフの下境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)

名前	説明
paddingLeft	コントロールの左境界線とコンテンツ領域の左端との間にあるピクセル数。デフォルト値は 2 です。(FlexChart から継承されました)
paddingRight	コントロールの右境界線とコンテンツ領域の右端との間にあるピクセル数。デフォルト値は 0 です。(FlexChart から継承されました)
paddingTop	グラフの上境界線とコンテンツ領域との間にあるピクセル数。(FlexChart から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChartItem	グラフ内の選択された ChartItem。複数の項目が選択されている場合、このプロパティは最も新しく選択された項目を指定します。(FlexChart から継承されました)
selectedChartItems	グラフ内の選択されたすべての ChartItem オブジェクトの配列。(FlexChart から継承されました)
selectionMode	ChartItem オブジェクトが選択できるかどうか。(FlexChart から継承されました)
showDataTips	グラフの DataTip コントロールが表示されるかどうか。(FlexChart から継承されました)
showDataTipTargets	showDataTips が true に設定されているときに、データポイント上にターゲットを表示するかどうか。(FlexChart から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexCartesianChart から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalAxisRatio	垂直軸の幅の制限。(FlexCartesianChart から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexChart から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>GetValues</i>	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(<i>FlexChart</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	ユーザーが列ヘッダー上でマウスを押下し、離れたことを示す headerRelease イベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexChart から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexPlotSeries クラス

説明

PlotChart コントロールのデータ系列を定義します。

継承階層

- [FlexChartSeries](#)
 - FlexPlotSeries

構文

```
'Declaration  
Public Class FlexPlotSeries _  
Inherits FlexChartSeries
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexChartSeries から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexChartSeries から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(FlexChartSeries から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、

名前	説明
	flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexChartSeries から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexChartSeries から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
radius	半径のスタイル
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	データ プロバイダーのデータ内で選択された項目のインデックス。(FlexChartSeries から継承されました)
selectedIndices	データ プロバイダーのデータ内で選択された項目のインデックスの配列。(FlexChartSeries から継承されました)
selectedItem	系列内で選択されたグラフ項目。(FlexChartSeries から継承されました)
selectedItems	系列内で選択されたグラフ項目の配列。(FlexChartSeries から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>xField</i>	各データ ポイントの x 軸の位置を決定するデータ プロバイダーのフィールド。
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>yField</i>	各データ ポイントの y 軸の位置を決定するデータ プロバイダーのフィールド。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexChartSeries から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SeriesClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesDoubleClick	グラフ内のデータを表すグラフ項目上でマウス ボタンをダブルクリックしたインデックスです。(FlexChartSeries から継承されました)
SeriesItemRollOver	グラフ内のデータを表すグラフ項目上でマウス ポインタを動かしたインデックスです。(FlexChartSeries から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexPopUpButton クラス

説明

PopUpButton コントロールは、Button コントロールにポップアップ コントロールのインターフェイスを追加します。メイン ボタンとポップアップ ボタンと呼ばれる小さなボタンを持ち、ユーザーがポップアップ ボタンをクリックすると、任意の UIComponent オブジェクトをポップアップします。

PopUpButton コントロールには、テキスト ラベル、アイコンのいずれか、または両方を表示できます。PopUpButton コントロールのメイン ボタンをユーザーがクリックすると、click イベントがディスパッチされます。

PopUpButton コントロールの一般的な使用法の 1 つとして、ポップアップ ボタンで、メイン ボタンの機能やラベルを変更する List コントロールや Menu コントロールを開くことが挙げられます。

継承階層

- [FlexButton](#)
 - FlexPopUpButton

構文

```
'Declaration  
Public Class FlexPopUpButton _  
Inherits FlexButton
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。(FlexButton から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
disabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。(FlexButton から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コントロールの背景の塗りつぶしに使用するアルファ値。(FlexButton から継承されました)
fillColors	コントロールの背景を濃淡色化するのに使用する色。(FlexButton から継承されました)
focusAlpha	フォーカス スキンのアルファ透過値。デフォルト値は 0.4 です。(FlexButton から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
fontFamily	使用するフォントの名前。(FlexButton から継承されました)

名前	説明
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexButton</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexButton</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexButton</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	Button コントロールに表示されるテキスト。 (<i>FlexButton</i> から継承されました)
<i>labelPlacement</i>	指定されたアイコンを基準にしたラベルの向き。 (<i>FlexButton</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selected</i>	トグルボタンがオン (true) またはオフ (false) のどちらであるか。 (<i>FlexButton</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
textAlign	コンテナ内のテキストの配置。(FlexButton から継承されました)
textDecoration	テキストに下線を表示するかどうか。(FlexButton から継承されました)
textIndent	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。(FlexButton から継承されました)
textRollOverColor	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。(FlexButton から継承されました)
textSelectedColor	ユーザーがラベルを選択したときのラベルのテキストの色。デフォルト値は 0x000000 です。(FlexButton から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toggle	Button を toggle 状態にするかどうか。(FlexButton から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	指定した <code>UIComponent</code> が閉じるときにディスパッチされます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
Open	指定した UIComponent が開くときにディスパッチされます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexButton から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexProgressBar クラス

説明

ProgressBar コントロールは、時間の経過に伴うタスクの進捗状況を視覚的に表示します。ProgressBar コントロールには、確定モードと不確定モードの 2 つの種類があります。

確定モードの ProgressBar コントロールは、時間の経過に伴うタスクの進捗状況を直線的に表示します。タスクの範囲がわかっている場合は、確定モードの ProgressBar を使用します。タスクの終了までユーザーが長時間待たなければならぬときに表示します。

不確定モードの ProgressBar コントロールは、範囲が不明な時間ベースのプロセスを表示します。処理の規模が明らかな場合は、確定モードの ProgressBar コントロールを使用することをお勧めします。

継承階層

- [FlexObject](#)
 - FlexProgressBar

構文

```
'Declaration
Public Class FlexProgressBar _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
barColor	ProgressBar の色。
borderColor	3D 境界線の黒セクションまたは 2D 境界線の色セクション。デフォルト値は 0xB7BABC です。
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。
conversion	読み込んだ値の現在のバイト数と合計バイト数を変換するために使用する数値。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
direction	タスクの完了に向かって ProgressBar が伸びていく方向。MXML での有効な値は "right" と "left" です。

名前	説明
disabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
fillColors	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
fontFamily	使用するフォントの名前。
fontSize	テキストの高さ (ピクセル単位)。
fontStyle	テキストがイタリック フォントかどうか。
fontWeight	テキストがボールド体かどうか。
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
indeterminate	ProgressBar コントロールの外観が確定モードか不確定モードか。
label	進捗バーの近くに表示するテキスト。
labelPlacement	ラベルの配置位置。
labelWidth	ラベルの幅 (ピクセル単位)。
maximum	ProgressBar の最大進捗量を示す値。このプロパティは手動モードでのみ使用できます。
minimum	ProgressBar の最小進捗量を示す値。このプロパティは、開発者が手動モードでのみ設定します。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentComplete	完了した処理のパーセンテージ。0 から 100 の範囲の値です。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)

名前	説明
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>textAlign</i>	コンテナー内のテキストの配置。
<i>textDecoration</i>	テキストに下線を表示するかどうか。
<i>textIndent</i>	テキストの 1 行目のコンテナー左端からのオフセット（ピクセル単位）。デフォルト値は 0 です。
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値（たとえば、テキストコントロールのテキスト）。(TestObject から継承されました)
<i>value</i>	最小値と最大値の間で進行した進捗量。
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅（ピクセル単位）。(FlexDisplayObject から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexRadioButton クラス

説明

RadioButton コントロールを使用すると、一連の相互排他的な選択肢の中から 1 つだけをユーザーに選択させることができます。

継承階層

- [FlexButton](#)
 - FlexRadioButton

構文

```
'Declaration
Public Class FlexRadioButton _
Inherits FlexButton
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。デフォルト値は 0x0B333C です。(FlexButton から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
disabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。(FlexButton から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)

名前	説明
<i>fillAlphas</i>	コントロールの背景の塗りつぶしに使用するアルファ値。 (<i>FlexButton</i> から継承されました)
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。 (<i>FlexButton</i> から継承されました)
<i>focusAlpha</i>	フォーカススキンのアルファ透過値。デフォルト値は 0.4 です。 (<i>FlexButton</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。 (<i>FlexButton</i> から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。 (<i>FlexButton</i> から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。 (<i>FlexButton</i> から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。 (<i>FlexButton</i> から継承されました)
<i>groupName</i>	この <i>RadioButton</i> コントロールが属するグループの名前。または、この <i>RadioButton</i> が <i>RadioButtonGroup</i> コントロールによって定義されたグループの一部の場合は、 <i>RadioButtonGroup</i> コントロールの <i>id</i> プロパティの値を指定します。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	Button コントロールに表示されるテキスト。 (<i>FlexButton</i> から継承されました)
<i>labelPlacement</i>	指定されたアイコンを基準にしたラベルの向き。 (<i>FlexButton</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の <i>InteractiveObject</i> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <i>mouseEnabled</i> を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <i>mouseEnabled</i> 動作を変更するには、 <i>flash.display.DisplayObjectContainer.mouseChildren</i> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <i>width</i> または <i>explicitWidth</i> プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)

名前	説明
<i>scaleX</i>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>selected</i>	トグルボタンがオン (true) またはオフ (false) のどちらであるか。(<i>FlexButton</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>textAlign</i>	コンテナ内のテキストの配置。(<i>FlexButton</i> から継承されました)
<i>textDecoration</i>	テキストに下線を表示するかどうか。(<i>FlexButton</i> から継承されました)
<i>textIndent</i>	テキストの 1 行目のコンテナ左端からのオフセット (ピクセル単位)。デフォルト値は 0 です。(<i>FlexButton</i> から継承されました)
<i>textRollOverColor</i>	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。(<i>FlexButton</i> から継承されました)
<i>textSelectedColor</i>	ユーザーがラベルを選択したときのラベルのテキストの色。デフォルト値は 0x000000 です。(<i>FlexButton</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<i>toggle</i>	Button を toggle 状態にするかどうか。(<i>FlexButton</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(<i>FlexObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)

名前	説明
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (FlexButton から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexRepeater クラス

説明

リピータ コントロールを識別し、そのサブコンポーネントの複数のインスタンスを作成します。

継承階層

- [TestObject](#)
 - FlexRepeater

構文

```
'Declaration
Public Class FlexRepeater _
Inherits TestObject _
Implements IClickable, IKeyable
```

プロパティ

名前	説明
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。
automationName	このオブジェクトの識別子として使用することのできる名前。
className	コントロールの開発者によって割り当てられた、要素のクラス名を含んだ文字列。
columnNames	データ内のすべての列の名前を含むリスト。
count	この Repeater が実行される回数。
firstVisibleRow	最初の可視の子のインデックス。
id	要素に対するオートメーション識別子 (ID) を含んだ文字列。
lastVisibleRow	最後の可視の子のインデックス。
numColumns	利用可能なデータ中の列の合計数。
numRows	利用可能なデータ中の行の合計数。
recycleChildren	この Repeater が前に作成した子を再使用するのか、新しい子を作成するのかどうか。

名前	説明
startingIndex	この Repeater が子の作成を開始する dataProvider へのインデックス。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexRule クラス

説明

単一の水平方向または垂直方向の線を作成します。通常、このコントロールはコンテナの内部を分割する線を作成するために使用します。

継承階層

- [FlexObject](#)
 - FlexRule

構文

```
'Declaration  
Public Class FlexRule _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
color	行の色。
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)

名前	説明
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
strokeWidth	罫線の太さです (ピクセル単位)。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。 (<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。 (<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステ

名前	説明
WaitForProperty	<p>ートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexScrollBar クラス

説明

表示領域に対してデータが多すぎる場合に表示するデータの部分を制御するために使用します。

継承階層

- [FlexObject](#)
 - FlexScrollBar

構文

```
'Declaration
Public Class FlexScrollBar _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>fillColors</code>	コントロールの背景を濃淡色化するのに使用する色。
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>lineScrollSize</code>	矢印ボタンが押されたときにスクロールする量 (ピクセル単位)。 デフォルト値は 1 です。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<code>pageScrollSize</code>	スクロール バーのトラックが押されたときのスクロール サムの移動量 (ピクセル単位)。 デフォルト値は 0 です。
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<code>scrollPosition</code>	現在のスクロール位置を表す数値。
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<code>Text</code>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)

名前	説明
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
trackColors	トラックの色 (2 色の配列として)。
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	ScrollBar コントロールがユーザーによる動作やプログラムによってスクロールされた場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexScrollBase クラス

説明

ScrollControlBase クラスは、スクロールバーを持つコントロールの基本クラスです。

スクロールバーはユーザーが操作し、表示可能領域を変更するメソッドおよびプロパティには開発者がアクセスします。ScrollControlBase が持つ子オブジェクトは 1 つのみで、そのオブジェクトを配置およびマスクするか、またはサイズ変更して、表示可能なコンテンツを表示します。スクロールするすべての項目は、このコンテンツ オブジェクトの子であることが必要です。

継承階層

- [FlexObject](#)
 - FlexScrollBase
 - [FlexListBase](#)
 - [FlexTextArea](#)

- [SparkAirHTML](#)

構文

```
'Declaration
Public Class FlexScrollBase _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)

名前	説明
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexSlider クラス

説明

スライダー トラックの両端の間でスライダー サムを移動することで値を選択できます。

スライダーの現在の値は、スライダーの最小値と最大値に対応するスライダーの両終点の間におけるサムの相対位置によって決定されます。

継承階層

- [FlexObject](#)
 - FlexSlider

構文

```
'Declaration
Public Class FlexSlider _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dataTipPrecision	データ ヒント テキストで使用する小数点以下の桁数。
direction	スライダー コントロールの向き。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)

名前	説明
<i>fillColors</i>	コントロールの背景を濃淡色化するのに使用する色。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>labels</i>	スライダーのラベルに使用する文字列の配列。
<i>labelStyleName</i>	スライダーのラベルに使用するスタイルの名前。 デフォルト値は未定義です。
<i>liveDragging</i>	スライダーのライブ ドラッグを有効にするかどうか。
<i>maximum</i>	スライダーに対して許可された最大値。 デフォルト値は 10 です。
<i>minimum</i>	スライダー コントロールに対して許可された最小値。 デフォルト値は 0 です。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>showDataTip</i>	ユーザーが操作している間、データ ヒントを表示するかどうか。
<i>snapInterval</i>	ユーザーがスライダー サムを移動したときの増分値。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>thumbCount</i>	スライダー上に配置できるサムの数。
<i>tickColor</i>	目盛りの色。
<i>tickLength</i>	目盛りの長さ (ピクセル単位)。
<i>tickThickness</i>	目盛りの太さ (ピクセル単位)。
<i>toolTip</i>	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>trackColors</i>	トラックの色 (2 色の配列として)。
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 <i>useHandCursor</i> が false の場合、代わりに矢印カーソルが表示されます。 (<i>FlexObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>value</i>	サムの位置。これは、minimum プロパティと maximum プロパティの間の値です。
<i>values</i>	<i>thumbCount</i> が 1 より大きい場合に、各サムの値を格納する配列。
<i>visible</i>	表示オブジェクトが可視かどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Change	マウスまたはキーボード操作によってスライダーの値が変更されたときにディスパッチされます。
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexStandalonePlayer クラス

説明

Flex スタンドアロン プレイヤー。

継承階層

- [Window](#)
 - FlexStandalonePlayer

構文

```
'Declaration  
Public Class FlexStandalonePlayer _  
Inherits Window
```

プロパティ

名前	説明
Application	このウィンドウが属するアプリケーションの名前。(Window から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Close	ウィンドウを閉じます。 (IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。 (IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。 (IMoveable から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexTabNavigator クラス

説明

子コンテナ間での移動のために TabBar コンテナを取り込むことで、ViewStack コンテナを拡張します。

継承階層

- [FlexViewStack](#)
 - FlexTabNavigator

構文

```
'Declaration
Public Class FlexTabNavigator _
Inherits FlexViewStack
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<i>backgroundAlpha</i>	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
<i>backgroundColor</i>	コンポーネントの背景色。(FlexContainer から継承されました)
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexContainer から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexContainer から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)

名前	説明
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexContainer から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>horizontalAlign</i>	この TabNavigator コンテナの上部にあるタブの水平位置。
<i>horizontalGap</i>	ピクセル単位で表したタブ間の間隔。デフォルト値は -1 で、この場合、隣接するタブの境界線が重なります。
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。(FlexContainer から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>selectedChild</i>	現在表示されている子コンテナへの参照。(FlexViewStack から継承されました)
<i>selectedIndex</i>	現在表示されている子コンテナの 0 を基準としたインデックス。(FlexViewStack から継承されました)

名前	説明
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabHeight	各タブの高さ (ピクセル単位)。
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
tabWidth	各タブの幅 (ピクセル単位)。
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Change	itemClick イベントのイベント オブジェクトの type プロパティの値を定義します。

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)

名前	説明
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。 (FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (FlexViewStack から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FlexTextArea クラス

説明

複数行のテキストフィールドで、境界線とオプションのスクロールバーが含まれます。

TextArea コントロールは、Flash Player と AIR の HTML レンダリング機能をサポートします。

継承階層

- [FlexScrollBase](#)
 - FlexTextArea

構文

```
'Declaration
Public Class FlexTextArea _
Inherits FlexScrollBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
color	コンポーネント ラベルを含むコンポーネントのテキストの色。 デフォルト値は 0x0B333C です。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
disabledColor	コンポーネントの背景色。

名前	説明
<i>displayAsPassword</i>	このコントロールがパスワード入力用であるかどうか。
<i>editable</i>	このコントロールでテキストを編集できるかどうか。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>fontFamily</i>	使用するフォントの名前。
<i>fontSize</i>	テキストの高さ (ピクセル単位)。
<i>fontStyle</i>	テキストがイタリック フォントかどうか。
<i>fontWeight</i>	テキストがボールド体かどうか。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalScrollPolicy</i>	水平スクロールバーが表示されるかどうか (常に表示: ScrollPolicy.ON、常に非表示: ScrollPolicy.OFF、必要に応じて表示: ScrollPolicy.AUTO)。
<i>htmlText</i>	TextInput コントロールによって表示されるテキスト (そのテキストのスタイルを表現する HTML マークアップを含む)。
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>length</i>	TextArea に表示されるテキストの文字数。
<i>maxChars</i>	テキスト フィールドにユーザーが入力できる最大文字数。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、こ

名前	説明
	のプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>restrict</i>	ユーザーがコントロールに入力できる文字のセット。
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>selectionBeginIndex</i>	現在の選択範囲の最初の文字を示す 0 を基準としたインデックス値。
<i>selectionEndIndex</i>	現在の選択範囲の最後の文字の直後の位置を示す 0 を基準としたインデックス。つまり、最後の文字の位置を示す 1 を基準としたインデックスと同じです。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>text</i>	コントロールに表示されるプレーン テキスト。
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>wordWrap</i>	行内のテキストを折り返すかどうか。
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。
SelectText	テキスト フィールド内の選択範囲を追跡します。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に

名前	説明
WaitForObject	対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexTitleWindow クラス

説明

タイトルバー、キャプション、境界線、および子のコンテンツ領域から構成されています。

通常 TitleWindow コンテナーは、必要な機能をすべて備えたアプリケーション モジュールをラップするために使用します。例えば、TitleWindow コンテナーにフォームを含めることができます。TitleWindow コンテナーは、ユーザーがフォームへの入力を完了したら、プログラムによって閉じることも、ユーザーが閉じるボタンを押して閉じるようにすることもできます。

継承階層

- [FlexPanel](#)
 - FlexTitleWindow

構文

```
'Declaration
Public Class FlexTitleWindow _
Inherits FlexPanel
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)

名前	説明
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dropShadowEnabled	Panel コンテナのドロップ シャドウが可視かどうか。デフォルト値は true です。(FlexPanel から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
fillColors	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexContainer から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
fontFamily	使用するフォントの名前。(FlexContainer から継承されました)
fontSize	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
fontStyle	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
fontWeight	テキストがボールド体かどうか。(FlexContainer から継承されました)

名前	説明
<i>footerColors</i>	フッター (ControlBar コンテナの領域) の背景の描画に使用される 2 色の配列。 (<i>FlexPanel</i> から継承されました)
<i>headerColors</i>	ヘッダーを描画するために使用する 2 色の配列。 (<i>FlexPanel</i> から継承されました)
<i>headerHeight</i>	ヘッダーの高さ。 デフォルト値はタイトル テキストのスタイルに基づいて計算されます。 (<i>FlexPanel</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>showCloseButton</i>	TitleWindow コンテナ内に閉じるボタンを表示するかどうか。 デフォルト値は false です。

名前	説明
status	タイトルバーのステータス領域に表示されるテキスト。デフォルト値は "" です。(FlexPanel から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
title	タイトルバーに表示されるタイトルまたはキャプション。デフォルト値は "" です。(FlexPanel から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)

名前	説明
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。 (FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexToggleButtonBar クラス

説明

選択状態または選択解除状態を維持しているボタンが水平方向に並んだグループまたは垂直方向に並んだグループを定義します。

ToggleButtonBar コントロールで 1 つのボタンだけが選択状態になることができます。つまり、ToggleButtonBar コントロールでいずれかのボタンを選択すると、別のボタンを選択するまでそのボタンは選択状態を維持します。

継承階層

- [FlexButtonBar](#)
 - FlexToggleButtonBar

構文

```
'Declaration
Public Class FlexToggleButtonBar _
Inherits FlexButtonBar
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)

名前	説明
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
direction	垂直レイアウト (デフォルト)、水平レイアウトのどちらを使用するか。(FlexBox から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
fillColors	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(FlexContainer から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
fontFamily	使用するフォントの名前。(FlexContainer から継承されました)
fontSize	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
fontStyle	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
fontWeight	テキストがボールド体かどうか。(FlexContainer から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
horizontalAlign	このコンテナ内の子の水平方向の配置。(FlexBox から継承されました)

名前	説明
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。(FlexContainer から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>selectedIndex</i>	アクティブなナビゲーション項目のインデックス。最初の項目のインデックスは 0 です。デフォルト値は -1 です。(FlexNavigationBar から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)

名前	説明
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
verticalAlign	このコンテナ内の子の垂直方向の配置。(FlexBox から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
Change	itemClick イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexNavigationBar から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)

名前	説明
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(FlexNavigationBar から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexTree クラス

説明

Tree コントロールを使用すると、展開可能なツリー形式で階層データを表示できます。

ツリー内の各項目はリーフかブランチのいずれかになります。リーフ項目とは、ツリー内の各末端にあるノードです。ブランチ項目とは、リーフ項目またはブランチ項目を含んだノードです。また、空になる場合もあります。

デフォルトでは、リーフは、ファイル アイコンの横のテキスト ラベルで表されます。ブランチは、フォルダー アイコンの横のテキスト ラベルで表されます。ブランチにはユーザーが開いて子を表示できる展開用矢印が付いています。

継承階層

- [FlexList](#)
 - FlexTree

構文

```
'Declaration
Public Class FlexTree _
Inherits FlexList
```

プロパティ

名前	説明
allowMultipleSelection	1 つ以上の項目を同時に選択することが許可されているかどうか。(FlexListBase から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alternatingItemColors	リストの項目の背景に使用する色。(FlexListBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合のテキストの色。デフォルト値は 0xAAB3B3 です。(FlexListBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)

名前	説明
<i>columnCount</i>	TileList コントロールまたは HorizontalList コントロールの項目に表示される列の数。 (<i>FlexListBase</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>FlexListBase</i> から継承されました)
<i>columnWidth</i>	コントロールの列の幅。 (<i>FlexListBase</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>dataTipField</i>	データ ヒントとして表示するデータ プロバイダー項目内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>depthColors</i>	Tree コントロールで使用する色の配列 (降順)。 デフォルト値は未定義です。
<i>disabledColor</i>	リストの項目が無効な場合の色。 デフォルト値は 0xDDDDDD です。
<i>editable</i>	データ プロバイダーの項目を編集できるかどうか。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>focusAlpha</i>	フォーカス スキンのアルファ透過値。 デフォルト値は 0.4 です。 (<i>FlexListBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>iconField</i>	アイコンとして表示するものを決定する、データ プロバイダー オブジェクト内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>indentation</i>	各ツリー レベルのインデント (ピクセル単位)。 デフォルト値は 17 です。
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。 (<i>FlexListBase</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>lockedColumnCount</i>	スクロールするコントロールの最初の列のインデックス。 (<i>FlexListBase</i> から継承されました)
<i>lockedRowCount</i>	スクロールするコントロールの最初の行のインデックス。 (<i>FlexListBase</i> から継承されました)

名前	説明
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexListBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexListBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
rollOverColor	ユーザーがレンダラの上にマウスを置いたときのレンダラの背景色。デフォルト値は 0xEEEEEE です。(FlexListBase から継承されました)
rowCount	表示する行の数。(FlexListBase から継承されました)
rowHeight	行の高さ (ピクセル単位)。(FlexListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectable	選択された項目を選択状態でリストに表示するかどうか。(FlexListBase から継承されました)
selectedIndex	データ プロバイダー内で選択された項目のインデックス。(FlexListBase から継承されました)
selectedIndices	データ プロバイダーで選択された項目のインデックスの配列。(FlexListBase から継承されました)
selectedItem	データ プロバイダー内で選択された項目への参照。(FlexListBase から継承されました)
selectedItems	データ プロバイダーで選択された項目への参照の配列。(FlexListBase から継承されました)
selectionColor	ユーザーが選択したときのレンダラの背景色。デフォルト値は 0x7FCEFF です。(FlexListBase から継承されました)

名前	説明
selectionDisabledColor	コンポーネントが無効化されたときのレンダラの背景色。デフォルト値は 0xDDDDDD です。(FlexListBase から継承されました)
showDataTips	行内のテキストに対して dataTips を表示するかどうか。(FlexListBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
textRollOverColor	マウス ポインターをボタンに合わせたときのラベルのテキストの色。デフォルト値は 0x2B333C です。(FlexListBase から継承されました)
textSelectedColor	ユーザーが選択したときのレンダラのテキスト色。デフォルト値は 0x2B333C です。(FlexListBase から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
useRollOver	項目の上にマウスを置いたときに、その項目をハイライト表示するかどうか。(FlexListBase から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
variableRowHeight	個々の行が別々の高さを持つことができるかどうか。(FlexListBase から継承されました)
verticalAlign	行のレンダラの垂直方向の配置。(FlexListBase から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
wordWrap	行内のテキストを折り返すかどうか。(FlexListBase から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)

名前	説明
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Close	ツリーの分岐を閉じるまたは折りたたみます。
Deselect	以前に選択した項目の選択を解除するときにディスパッチされるイベントの、イベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexListBase から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClick	コントロールの可視項目上でマウスがダブルクリックされたことを示す <code>itemDoubleClick</code> イベントにおける <code>ListEvent</code> オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexListBase から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexListBase から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexListBase から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexListBase から継承されました)
Edit	項目が編集可能な状態かどうかを示す <code>itemEditBegin</code> イベントにおけるイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexList から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメータが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(FlexListBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
MultiSelect	複数項目を選択するアクションの一部として項目が選択されたときにディスパッチされるイベントにおけるイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
Open	ツリーの分岐を開くまたは展開します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	scroll イベントのイベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
ScrollToIndex	指定したインデックス位置にあるデータ プロバイダー項目を表示状態にします。 (FlexListBase から継承されました)
Select	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
SelectIndex	単一の項目が選択されたときにディスパッチされるイベントの、イベント オブジェクトの type プロパティの値を定義します。 (FlexListBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (FlexListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexUIMovieClip クラス

説明

Adobe Flash CS3 Professional で Flex 用に作成するコンポーネントは、`mx.flash.UIMovieClip` クラスのサブクラスになります。

UIMovieClip クラスは、Flash コンポーネントを通常の Flex コンポーネントのように使用するために必要なインターフェイスを実装しています。したがって、UIMovieClip のサブクラスは、Flex コンテナの子として使用したり、スキンとして使用したりできます。また、このサブクラスを使用すると、Flex コンポーネントと同じ方法で、イベントに対する応答、ビューの状態や遷移の定義およびエフェクトの操作を行うことができます。

継承階層

- [FlexDisplayObject](#)
 - FlexUIMovieClip
 - [FlexContainerMovieClip](#)

構文

```
'Declaration
Public Class FlexUIMovieClip _
Inherits FlexDisplayObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。
automationName	このオブジェクトの識別子として使用することのできる名前。
className	"Button" のようなこのインスタンスのクラスの名前。
currentState	このコンポーネントの現在の状態。
enabled	ムービー クリップが有効かどうか。
errorColor	検証が失敗したときにコンポーネントをハイライトする色。
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。

名前	説明
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 有効な値は 0 から 100 です。
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 有効な値は 0 から 100 です。
<i>scaleX</i>	基準点から適用されるオブジェクトの水平倍率 (パーセンテージ)。 デフォルトの基準点は (0,0) です。 1.0 は 100% の倍率です。
<i>scaleY</i>	オブジェクトの基準点から適用されるオブジェクトの垂直倍率 (パーセンテージ)。 デフォルトの基準点は (0,0) です。 1.0 は 100% の倍率です。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。
<i>toolTip</i>	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。
<i>useHandCursor</i>	buttonMode プロパティが true に設定されたスプライト上にポインターが移動したときに、指差しハンド ポインター (ハンド カーソル) を表示するかどうか。
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>ChangeFocus</i>	keyFocusChange イベント オブジェクトの type プロパティの値を定義します。
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(TestObject から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
SetFocus	コンポーネントがフォーカスを受け取ったときに FocusManager によって呼ばれます。コンポーネントはその内部のコンポーネントに順々にフォーカスを設定できます。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexVideoDisplay クラス

説明

Flex アプリケーションで FLV ファイルを再生できます。HTTP を介したプログレッシブ ダウンロード、Flash Media Server からのストリーミング、および Camera オブジェクトからのストリーミングをサポートします。

継承階層

- [FlexObject](#)
 - FlexVideoDisplay

構文

```
'Declaration
Public Class FlexVideoDisplay _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(<i>FlexObject</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。(<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。(<i>FlexObject</i> から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<code>source</code>	ストリーミングする FLV ファイルの相対パスおよびファイル名。
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(<i>FlexDisplayObject</i> から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<code>Text</code>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<code>themeColor</code>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<code>toolTip</code>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(<i>FlexObject</i> から継承されました)
<code>totalTime</code>	メディアの全体の長さ (秒)。

名前	説明
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>volume</i>	ボリューム レベル。0 から 1 の範囲の値で指定します。デフォルト値は 0.75 です。
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(FlexObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexViewStack クラス

説明

ViewStack ナビゲーターのコンテナは、積み重ねられた一連の子コンテナから成ります。子コンテナは一度に 1 つのみ表示されます。別の子コンテナが選択されると、同じ場所に表示されるため、このコンテナによって既存のコンテナが置き換えられたように見えます。ただし、それまでの子コンテナもまだ存在しており、単に見えなくなっただけです。

継承階層

- [FlexContainer](#)
 - FlexViewStack
 - [FlexTabNavigator](#)

構文

```
'Declaration
Public Class FlexViewStack _
Inherits FlexContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
creationPolicy	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
fillAlphas	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
fillColors	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)

名前	説明
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexContainer から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexContainer から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexContainer から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	このコンテナーを表すために、一部のナビゲーター コンテナーによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナーに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numChildren</i>	このコンテナーに含まれる子コンポーネントの数。(FlexContainer から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)

名前	説明
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedChild	現在表示されている子コンテナへの参照。
selectedIndex	現在表示されている子コンテナの 0 を基準としたインデックス。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。(FlexContainer から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。 (FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Scroll	ユーザーがコンテナを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexWindow クラス

説明

Window は、AIR デスクトップ アプリケーションの追加ウィンドウに対する最上位のコンテナです。

継承階層

- [FlexContainer](#)
 - FlexWindow

構文

```
'Declaration
Public Class FlexWindow _
Inherits FlexContainer _
Implements IMoveable
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alwaysInFront	基になる NativeWindow が常に他のウィンドウの手前に表示されるかどうかを示す値。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)

名前	説明
<i>backgroundDisabledColor</i>	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
<i>backgroundImage</i>	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
<i>backgroundSize</i>	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>closed</i>	基になるウィンドウが閉じられているかどうかを示す値。
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
<i>creationIndex</i>	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexContainer から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexContainer から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexContainer から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)

名前	説明
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>isAIRWindow</i>	このウィンドウが AIR ウィンドウかどうかを示す値。
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。 (<i>FlexContainer</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>FlexContainer</i> から継承されました)
<i>maxHeight</i>	親がレイアウトを計算するために使用されるコンポーネントの高さの最大推奨値。
<i>maximizable</i>	ウィンドウを最大化できるかどうかを示す値。
<i>maxWidth</i>	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。
<i>minHeight</i>	親がレイアウトを計算するために使用されるコンポーネントの高さの最小推奨値。
<i>minimizable</i>	ウィンドウを最小化できるかどうかを示す値。
<i>minWidth</i>	親がレイアウトを計算するために使用されるコンポーネントの幅の最小推奨値。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numChildren</i>	このコンテナに含まれる子コンポーネントの数。 (<i>FlexContainer</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>FlexContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>FlexContainer</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>resizable</i>	ウィンドウをサイズ変更できるかどうかを示す値。
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)

名前	説明
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
showStatusBar	ステータスバーが可視かどうかを示す値。
status	ステータスバーに表示される文字 (表示されている場合)。
systemChrome	ウィンドウが持っているシステム クロム (存在する場合) のタイプ。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
title	ウィンドウのタイトルバーとタスクバーに表示されるタイトル。
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
transparent	ウィンドウを透明にするかどうか。
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Close	ウィンドウを閉じます。 (IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	<code>dragComplete</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
DragDrop	<code>dragDrop</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
DragStart	<code>dragStart</code> イベントのイベント オブジェクトの <code>type</code> プロパティの値を定義します。 (FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。 (IMoveable から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)

名前	説明
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(IMoveable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
StateChange	ウィンドウの displaystatus が変更されたときにディスパッチされません。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FlexWindowedApplication クラス

説明

WindowedApplication は、AIR アプリケーション用の Flex アプリケーションを作成するために使用するアプリケーション コンテナを定義します。

継承階層

- [FlexApplication](#)
 - FlexWindowedApplication

構文

```
'Declaration
Public Class FlexWindowedApplication _
Inherits FlexApplication _
Implements IMoveable
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alwaysInFront	基になる NativeWindow が常に他のウインドウの手前に表示されるかどうかを示す値。
applicationID	AIR がアプリケーションを識別するために使用する識別子。
autoExit	最後のウインドウを閉じたときに AIR アプリケーションが終了するか、バックグラウンドで実行を続けるかどうかを示す値。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
backgroundAlpha	backgroundImage スタイルによって定義された画像や SWF ファイルの backgroundColor プロパティによって定義される色のアルファレベル。有効な値は 0.0 から 1.0 の範囲です。(FlexContainer から継承されました)
backgroundColor	コンポーネントの背景色。(FlexContainer から継承されました)
backgroundDisabledColor	コンポーネントが無効な場合の背景色。グローバルのデフォルト値は未定義です。(FlexContainer から継承されました)
backgroundImage	コンポーネントの背景画像。絶対/相対 URL やクラスを指定できます。(FlexContainer から継承されました)
backgroundSize	指定した backgroundImage の画像サイズを変更する割合。(FlexContainer から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
closed	基になるウインドウが閉じられているかどうかを示す値。
columnNames	データ内のすべての列の名前を含むリスト。(FlexContainer から継承されました)
creationIndex	このコンテナの子をインスタンス化および描画する順番。(FlexContainer から継承されました)

名前	説明
<i>creationPolicy</i>	このコンテナの子の作成ポリシー。(FlexContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>fillAlphas</i>	コンポーネントの背景の塗りつぶしに使用するアルファ透過値。(FlexContainer から継承されました)
<i>fillColors</i>	コンポーネントの背景塗りつぶしを濃淡色化するのに使用する色。(FlexContainer から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(FlexContainer から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>fontFamily</i>	使用するフォントの名前。(FlexContainer から継承されました)
<i>fontSize</i>	テキストの高さ (ピクセル単位)。(FlexContainer から継承されました)
<i>fontStyle</i>	テキストがイタリック フォントかどうか。(FlexContainer から継承されました)
<i>fontWeight</i>	テキストがボールド体かどうか。(FlexContainer から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	このコンテナを表すために、一部のナビゲーター コンテナによって表示されるテキスト。(FlexContainer から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(FlexContainer から継承されました)
<i>maxHeight</i>	親がレイアウトを計算するために使用されるコンポーネントの高さの最大推奨値。
<i>maximizable</i>	ウィンドウを最大化できるかどうかを示す値。
<i>maxWidth</i>	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。
<i>minHeight</i>	親がレイアウトを計算するために使用されるコンポーネントの高さの最小推奨値。
<i>minimizable</i>	ウィンドウを最小化できるかどうかを示す値。

名前	説明
minWidth	親がレイアウトを計算するために使用されるコンポーネントの幅の最小推奨値。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numChildren	このコンテナに含まれる子コンポーネントの数。(FlexContainer から継承されました)
numColumns	利用可能なデータ中の列の合計数。(FlexContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。(FlexContainer から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizable	ウィンドウをサイズ変更できるかどうかを示す値。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
showStatusBar	ステータス バーが可視かどうかを示す値。
status	ステータス バーに表示される文字 (表示されている場合)。
systemChrome	ウィンドウが持っているシステム クロム (存在する場合) のタイプ。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)

名前	説明
title	ウィンドウのタイトルバーとタスクバーに表示されるタイトル。
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
transparent	ウィンドウを透明にするかどうか。
url	このアプリケーションの SWF ファイルがロードされた URL。(FlexApplication から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	dragComplete イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragDrop	dragDrop イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
DragStart	dragStart イベントのイベント オブジェクトの type プロパティの値を定義します。(FlexContainer から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。戻り値は行の配列で、各行は項目の配列になります。(FlexContainer から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。(FlexContainer から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(IMoveable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
Scroll	ユーザーがコンテナーを手動でスクロールさせたときにディスパッチされます。(FlexContainer から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkAirHTML クラス

説明

アプリケーションの HTML コンテンツを表示する HTML コントロール。

継承階層

- [FlexScrollBase](#)
 - SparkAirHTML

構文

```
'Declaration
Public Class SparkAirHTML _
Inherits FlexScrollBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)

名前	説明
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナーに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	InteractiveObject インスタンス上でマウス ホイールが回されたときにディスパッチされます。 (FlexScrollBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	スクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SparkApplication クラス

説明

Flex は、デフォルトのコンテナ (Application コンテナ) を定義するため、別のコンテナを明示的に定義することなく、アプリケーションにコンテンツを追加することができます。

継承階層

- [SparkSkinnableContainer](#)

- SparkApplication
 - [SparkWindowedApplication](#)

構文

```
'Declaration
Public Class SparkApplication _
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
colorCorrection	ステージの colorCorrection プロパティの値。
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
controlBarVisible	コントロール バーが可視かどうか。
creationPolicy	このコンポーネントの内容の作成ポリシー。(SparkSkinnableContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
deferredContentCreated	遅延コンテンツが作成されたかどうか。(SparkSkinnableContainer から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)

名前	説明
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。 (SparkSkinnableContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)

名前	説明
url	このアプリケーションの SWF ファイルがロードされた URL。
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
viewSourceURL	アプリケーションのソースを参照できる URL。
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkBorderContainer クラス

説明

BorderContainer クラスは、コンテナの境界線と背景の塗りつぶしの外観を制御する一連の CSS スタイルを定義します。

継承階層

- [SparkSkinnableContainer](#)
 - SparkBorderContainer

構文

```
'Declaration  
Public Class SparkBorderContainer _  
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
creationPolicy	このコンポーネントの内容の作成ポリシー。(SparkSkinnableContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
deferredContentCreated	遅延コンテンツが作成されたかどうか。(SparkSkinnableContainer から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)

名前	説明
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkSkinnableContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkButton クラス

説明

SparkButton コンポーネントはきわめて使用頻度の高い四角形のボタンです。SparkButton コンポーネントは、押すことができるような外観をしています。

継承階層

- [SparkButtonBase](#)
 - SparkButton
 - [SparkMuteButton](#)

構文

```
'Declaration
Public Class SparkButton _
Inherits SparkButtonBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に <code>buttonDown</code> イベントを繰り返しディスパッチするかどうか。(SparkButtonBase から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
emphasized	フォーカス マネージャーによって要求されたおりのデフォルト ボタンかどうか。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
label	<code>ButtonBase</code> コントロールに表示されるテキスト。 (SparkButtonBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は <code>true</code> であり、これは表示リスト上の <code>InteractiveObject</code> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は

名前	説明
	NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
stickyHighlighting	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。(SparkButtonBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkButtonBar クラス

説明

ButtonBar コントロールは、共通の外観と操作性を持つ論理的に関連するボタンの水平または垂直方向のグループを定義します。

継承階層

- [SparkListBase](#)
 - SparkButtonBar

構文

```
'Declaration  
Public Class SparkButtonBar _  
Inherits SparkListBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableDataContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
caretIndex	現在フォーカスがある項目。(SparkListBase から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)

名前	説明
firstVisibleRow	最初の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
labelField	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。(SparkListBase から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
requireSelection	コントロール内でデータ項目が常に選択されている必要があるかどうか。(SparkListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	選択された項目の 0 を基準としたインデックス。項目が選択されていない場合は -1 です。(SparkListBase から継承されました)
selectedItem	現在選択されている項目。(SparkListBase から継承されました)

名前	説明
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
useVirtualLayout	このコントロールに関連付けられているレイアウトの useVirtualLayout プロパティの値。 (SparkListBase から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemsCount	ComboBox の項目の数を取得します。(SparkListBase から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)

名前	説明
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。(SparkListBase から継承されました)
SelectIndex	ユーザーがリスト上の項目をクリックしたり、キーボードを使って項目を移動したりしたときにディスパッチされます。

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkButtonBarButton クラス

説明

BarButtonButton クラスは、BarButton コントロールによって使用されるカスタム項目レンダラを定義します。

継承階層

- [SparkToggleButton](#)
 - SparkButtonBarButton

構文

```
'Declaration
Public Class SparkButtonBarButton _
Inherits SparkToggleButton
```

プロパティ

名前	説明
allowDeselection	現在選択したボタンをユーザーがクリックすると選択解除されるかどうかを示す値。
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に buttonDown イベントを繰り返しディスパッチするかどうか。(SparkButtonBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)

名前	説明
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
<code>id</code>	コンポーネントの ID。 (FlexObject から継承されました)
<code>label</code>	ButtonBase コントロールに表示されるテキスト。 (SparkButtonBase から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
<code>selected</code>	ボタンがダウン状態にあるかどうか。アップ状態の場合は false です。 (SparkToggleButtonBase から継承されました)
<code>showsCaret</code>	項目レンダラがそれ自身をフォーカスありとして表示できるかどうかを示す値。
<code>stickyHighlighting</code>	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。 (SparkButtonBase から継承されました)

名前	説明
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。 (SparkToggleButton から継承されました)
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。 (SparkToggleButton から継承されました)
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。 (SparkToggleButton から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkButtonBase クラス

説明

SparkButtonBase クラス はすべての Spark ボタン コンポーネントの基本クラスです。

継承階層

- [SparkObject](#)
 - SparkButtonBase
 - [SparkButton](#)
 - [SparkToggleButtonBase](#)

構文

```
'Declaration  
Public Class SparkButtonBase _  
Inherits SparkObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に buttonDown イベントを繰り返しディスパッチするかどうか。
className	<i>Button</i> のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)

名前	説明
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	ButtonBase コントロールに表示されるテキスト。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>stickyHighlighting</i>	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkCheckBox クラス

説明

CheckBox コンポーネントは、オプションのラベルと、チェック マークの表示と非表示の状態を保持する小さなボックスから構成されます。

継承階層

- [SparkToggleButtonBase](#)
 - SparkCheckBox

構文

```
'Declaration
Public Class SparkCheckBox _
Inherits SparkToggleButtonBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<i>autoRepeat</i>	ユーザーがマウス ボタンを押し続けた場合に <code>buttonDown</code> イベントを繰り返しディスパッチするかどうか。 (<i>SparkButtonBase</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	<i>ButtonBase</i> コントロールに表示されるテキスト。 (<i>SparkButtonBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は <code>true</code> であり、これは表示リスト上の <i>InteractiveObject</i> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は <code>NaN</code> です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは <code>NaN</code> にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)

名前	説明
selected	ボタンがダウン状態にあるかどうか。アップ状態の場合は false です。(SparkToggleButtonBase から継承されました)
stickyHighlighting	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。(SparkButtonBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイム

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkComboBox クラス

説明

ComboBox コントロールは、DropDownListBase コントロールの子クラスです。DropDownListBase コントロールと同様に、ComboBox コントロールのドロップダウン リストからユーザーが項目を選択すると、コントロールのプロンプト領域にデータ項目が表示されます。

継承階層

- [SparkDropDownListBase](#)
 - SparkComboBox

構文

```
'Declaration
Public Class SparkComboBox _
Inherits SparkDropDownListBase
```

プロパティ

名前	説明
allowMultipleSelection	複数選択が可能かどうか。(SparkList から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableDataContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
caretIndex	現在フォーカスがある項目。(SparkListBase から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dragEnabled	このコントロールの外に項目をドラッグして他のコントロールにドロップすることができるかどうかを示すフラグ。(SparkList から継承されました)

名前	説明
dragMoveEnabled	ドラッグ&ドロップ操作としてコントロールから項目をコピーするのではなく、移動することができるかどうかを示すフラグ。 (SparkList から継承されました)
dropEnabled	ドラッグした項目をコントロール上にドロップできるかどうかを示すフラグ。 (SparkList から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
isDropDownOpen	ドロップダウンが開いているかどうか。 (SparkDropDownListBase から継承されました)
labelField	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。 (SparkListBase から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
maxChars	ユーザーが入力するときに、プロンプト領域に入る最大の文字数。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)

名前	説明
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>requireSelection</i>	コントロール内でデータ項目が常に選択されている必要があるかどうか。(SparkListBase から継承されました)
<i>restrict</i>	ユーザーがプロンプト領域に入力できる文字のセット。
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>selectedIndex</i>	選択された項目の 0 を基準としたインデックス。項目が選択されていない場合は -1 です。(SparkListBase から継承されました)
<i>selectedIndices</i>	現在選択されている項目のインデックスを表すリスト。(SparkList から継承されました)
<i>selectedItem</i>	現在選択されている項目。(SparkListBase から継承されました)
<i>selectedItems</i>	現在選択されている項目を表すオブジェクトのリスト。(SparkList から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>useVirtualLayout</i>	このコントロールに関連付けられているレイアウトの useVirtualLayout プロパティの値。(SparkListBase から継承されました)
<i>Value</i>	コントロールの値（たとえば、テキストコントロールのテキスト）。(TestObject から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅（ピクセル単位）。(FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>DragCancel</i>	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragDrop</i>	ドラッグした項目がドロップされた場合にディスパッチされます。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragStart</i>	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。 (<i>TestObject</i> から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemsCount	ComboBox の項目の数を取得します。(SparkListBase から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
Open	ユーザーがドロップダウン リストを表示するためにドロップダウン ボタンをクリックするときにディスパッチされます。(SparkDropDownListBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。(SparkListBase から継承されました)
SelectIndex	ユーザーがリスト上の項目をクリックしたり、キーボードを使って項目を移動したりしたときにディスパッチされます。(SparkList から継承されました)
SelectText	テキストが選択された場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkListBase から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SparkComplexDisplay クラス

説明

複雑なレンダリング オブジェクトです。

継承階層

- [SparkListLabel](#)
 - SparkComplexDisplay

構文

```
'Declaration  
Public Class SparkComplexDisplay _  
Inherits SparkListLabel
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkGroupBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
automationValue	この項目を一意に識別するのに使用できる値。(SparkListLabel から継承されました)
blendMode	使用するブレンド モードを指定する BlendMode クラスの値。(SparkGroup から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipAndEnableScrolling	ビューポートの境界に子をクリップするかどうか。(SparkGroupBase から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkGroupBase から継承されました)
contentHeight	ビューポートの内容の高さ。(SparkGroupBase から継承されました)
contentWidth	ビューポートの内容の幅。(SparkGroupBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dragging	SparkListLabel がドラッグ モード中かどうかの値。(SparkListLabel から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)

名前	説明
firstVisibleRow	最初の可視の子のインデックス。(SparkGroupBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
horizontalScrollPosition	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。(SparkGroupBase から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
label	SparkListLabel のラベル。(SparkListLabel から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(SparkGroupBase から継承されました)
luminosityClip	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。(SparkGroupBase から継承されました)
luminosityInvert	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。(SparkGroupBase から継承されました)
maskType	マスクの種類。(SparkGroupBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseEnabledWhereTransparent	Group の境界全体が、クリックやロールオーバーなどのマウス イベントに応答するかどうかを示す値。(SparkGroupBase から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkGroupBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkGroupBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkGroupBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、こ

名前	説明
	のプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizeMode	このコンテナの ResizeMode。(SparkGroupBase から継承されました)
scaleGridBottom	スケールグリッドの下座標。(SparkGroup から継承されました)
scaleGridLeft	スケールグリッドの左座標。(SparkGroup から継承されました)
scaleGridRight	スケールグリッドの右座標。(SparkGroup から継承されました)
scaleGridTop	スケールグリッドの上座標。(SparkGroup から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selected	SparkListLabel が選択されているかどうか。(SparkListLabel から継承されました)
showsCaret	キャレットが表示されるかどうか。(SparkListLabel から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。デフォルト値は (0,0) であり、コンポーネントの左上隅。(SparkGroupBase から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>DragCancel</i>	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (<i>SparkGroupBase</i> から継承されました)
<i>DragDrop</i>	ドラッグした項目がドロップされた場合にディスパッチされます。 (<i>SparkGroupBase</i> から継承されました)
<i>DragStart</i>	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。 (<i>SparkGroupBase</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケータで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケータで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケータを返します。 (<i>TestObject</i> から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkGroupBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナー上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。(SparkGroupBase から継承されました)

名前	説明
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。(SparkGroupBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkGroupBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定し

名前	説明
WaitForDisappearance	<p>ない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkDataGrid クラス

説明

Spark DataGrid コントロールを識別します。

継承階層

- [SparkSkinnableContainerBase](#)
 - SparkDataGrid

構文

```
'Declaration
Public Class SparkDataGrid _
Inherits SparkSkinnableContainerBase
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。 (<i>FlexDisplayObject</i> から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>columnsLength</i>	列の数。
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <i>mouseEnabled</i> を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <i>mouseEnabled</i> 動作を変更するには、 <i>flash.display.DisplayObjectContainer.mouseChildren</i> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)

名前	説明
numColumns	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
rowHeight	行の高さ。
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedCell	選択されているセル。
selectedCells	選択されているセル。
selectedIndex	選択されている行のインデックス。
selectedIndices	選択されている行のインデックス。
selectedItem	選択されている行の項目。
selectedItems	選択されている行の項目。
selectionLength	選択されている行の数。
selectionMode	選択モード (singlerow、multiplerows、singlecells、または multiplecells)。
showDataTips	データ ヒントが表示されるかどうか。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>ColumnStretch</i>	データ グリッドの指定した列のサイズを変更します。
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>DragCancel</i>	ドラッグ操作がキャンセルされた場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragDrop</i>	ドラッグした項目がドロップされた場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragStart</i>	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>EditNext</i>	次のデータ グリッド セルを編集モードに設定します。
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HeaderClick	指定した列ヘッダーをクリックします。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (<i>FlexObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	指定したデータ グリッド セルを選択します
<i>SetFocus</i>	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (<i>FlexObject</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ToString</i>	コントロールの文字列表現を返します。 (<i>FlexDisplayObject</i> から継承されました)
<i>Type</i>	ユーザーがキーを押したときにディスパッチされます。
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkDataGridLabel クラス

説明

Spark DataGridLabel コントロールを識別します。

継承階層

- [FlexObject](#)
 - SparkDataGridLabel

構文

```
'Declaration
Public Class SparkDataGridLabel _
Inherits FlexObject
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>automationValue</i>	この項目を一意に識別するのに使用できる値。
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)

名前	説明
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkDataGroup クラス

説明

DataGroup クラスはデータ項目の基本コンテナー クラスです。

継承階層

- [SparkGroupBase](#)
 - SparkDataGroup

構文

```
'Declaration  
Public Class SparkDataGroup _  
Inherits SparkGroupBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkGroupBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipAndEnableScrolling	ビューポートの境界に子をクリップするかどうか。(SparkGroupBase から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkGroupBase から継承されました)
contentHeight	ビューポートの内容の高さ。(SparkGroupBase から継承されました)
contentWidth	ビューポートの内容の幅。(SparkGroupBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkGroupBase から継承されました)

名前	説明
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>horizontalScrollPosition</i>	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。 (<i>SparkGroupBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(<i>SparkGroupBase</i> から継承されました)
<i>luminosityClip</i>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<i>luminosityInvert</i>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。(<i>SparkGroupBase</i> から継承されました)
<i>maskType</i>	マスクの種類。(<i>SparkGroupBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>mouseEnabledWhereTransparent</i>	Group の境界全体が、クリックやロールオーバーなどのマウス イベントに応答するかどうかを示す値。(<i>SparkGroupBase</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(<i>SparkGroupBase</i> から継承されました)
<i>numElements</i>	このコンテナに含まれるビジュアル要素の数。 (<i>SparkGroupBase</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(<i>SparkGroupBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)

名前	説明
resizeMode	このコンテナの ResizeMode。 (SparkGroupBase から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。 デフォルト値は (0,0) であり、コンポーネントの左上隅。 (SparkGroupBase から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。 キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保

名前	説明
	<p>存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
<code>ChangeFocus</code>	現在のフォーカスを変更します。 (<code>FlexObject</code> から継承されました)
<code>Click</code>	オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)
<code>DragCancel</code>	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (<code>SparkGroupBase</code> から継承されました)
<code>DragDrop</code>	ドラッグした項目がドロップされた場合にディスパッチされます。 (<code>SparkGroupBase</code> から継承されました)
<code>DragStart</code>	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。 (<code>SparkGroupBase</code> から継承されました)
<code>Exists</code>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)
<code>Find</code>	XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
<code>FindAll</code>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。 (<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (<code>TestObject</code> から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkGroupBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナー上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。(SparkGroupBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。(SparkGroupBase から継承されました)
Select	1 つ以上の項目がリストから選択解除されたときにディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkGroupBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkDataRenderer クラス

説明

SparkDataRenderer クラス は Spark のデータ コンポーネントの基本クラスです。

継承階層

- [SparkGroup](#)
 - SparkDataRenderer
 - [SparkListLabel](#)

構文

```
'Declaration
Public Class SparkDataRenderer _
Inherits SparkGroup
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkGroupBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)

名前	説明
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>blendMode</i>	使用するブレンド モードを指定する BlendMode クラスの値。 (<i>SparkGroup</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。(<i>FlexObject</i> から継承されました)
<i>clipAndEnableScrolling</i>	ビューポートの境界に子をクリップするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(<i>SparkGroupBase</i> から継承されました)
<i>contentHeight</i>	ビューポートの内容の高さ。(<i>SparkGroupBase</i> から継承されました)
<i>contentWidth</i>	ビューポートの内容の幅。(<i>SparkGroupBase</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(<i>SparkGroupBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>horizontalScrollPosition</i>	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。 (<i>SparkGroupBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(<i>SparkGroupBase</i> から継承されました)
<i>luminosityClip</i>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<i>luminosityInvert</i>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。(<i>SparkGroupBase</i> から継承されました)

名前	説明
maskType	マスクの種類。(SparkGroupBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
mouseEnabledWhereTransparent	Group の境界全体が、クリックやロール オーバーなどのマウス イベントに応答するかどうかを示す値。(SparkGroupBase から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkGroupBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkGroupBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkGroupBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizeMode	このコンテナの ResizeMode。(SparkGroupBase から継承されました)
scaleGridBottom	スケールグリッドの下座標。(SparkGroup から継承されました)
scaleGridLeft	スケールグリッドの左座標。(SparkGroup から継承されました)
scaleGridRight	スケールグリッドの右座標。(SparkGroup から継承されました)
scaleGridTop	スケールグリッドの上座標。(SparkGroup から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)

名前	説明
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。 デフォルト値は (0,0) であり、コンポーネントの左上隅。 (SparkGroupBase から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。 キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。 それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (SparkGroupBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。 (SparkGroupBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkGroupBase から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkGroupBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナ上のスクロール バーを動かすためにマウスが使用されたときにディスパッチされます。(SparkGroupBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。(SparkGroupBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkGroupBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkDropDownList クラス

説明

SparkDropDownList コントロールには、ユーザーが 1 つの値を選択できるドロップダウン リストが含まれています。

継承階層

- [SparkDropDownListBase](#)
 - SparkDropDownList

構文

```
'Declaration
Public Class SparkDropDownList _
Inherits SparkDropDownListBase
```

プロパティ

名前	説明
allowMultipleSelection	複数選択が可能かどうか。(SparkList から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableDataContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
caretIndex	現在フォーカスがある項目。(SparkListBase から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)

名前	説明
dragEnabled	このコントロールの外に項目をドラッグして他のコントロールにドロップすることができるかどうかを示すフラグ。(SparkList から継承されました)
dragMoveEnabled	ドラッグ&ドロップ操作としてコントロールから項目をコピーするのではなく、移動することができるかどうかを示すフラグ。(SparkList から継承されました)
dropEnabled	ドラッグした項目をコントロール上にドロップできるかどうかを示すフラグ。(SparkList から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
isDropDownOpen	ドロップダウンが開いているかどうか。(SparkDropDownListBase から継承されました)
labelField	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。(SparkListBase から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)

名前	説明
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。（ FlexObject から継承されました）
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。（ FlexObject から継承されました）
prompt	DropDownList コントロールのプロンプト。
requireSelection	コントロール内でデータ項目が常に選択されている必要があるかどうか。（ SparkListBase から継承されました）
scaleX	水平方向の倍率を指定する係数。（ FlexObject から継承されました）
scaleY	垂直方向の倍率を指定する係数。（ FlexObject から継承されました）
selectedIndex	選択された項目の 0 を基準としたインデックス。項目が選択されていない場合は -1 です。（ SparkListBase から継承されました）
selectedIndices	現在選択されている項目のインデックスを表すリスト。（ SparkList から継承されました）
selectedItem	現在選択されている項目。（ SparkListBase から継承されました）
selectedItems	現在選択されている項目を表すオブジェクトのリスト。（ SparkList から継承されました）
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。（ FlexDisplayObject から継承されました）
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。（ FlexDisplayObject から継承されました）
tabIndex	SWF ファイル内のオブジェクトのタブ順序。（ FlexDisplayObject から継承されました）
Text	コントロールのテキスト。（ TestObject から継承されました）
themeColor	コンポーネントのテーマ色。（ FlexObject から継承されました）
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。（ FlexObject から継承されました）
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。（ FlexObject から継承されました）
useVirtualLayout	このコントロールに関連付けられているレイアウトの useVirtualLayout プロパティの値。（ SparkListBase から継承されました）
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。（ TestObject から継承されました）
visible	表示オブジェクトが可視かどうか。（ FlexDisplayObject から継承されました）

名前	説明
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemsCount	ComboBox の項目の数を取得します。(SparkListBase から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
Open	ユーザーがドロップダウン リストを表示するためにドロップダウン ボタンをクリックするときにディスパッチされます。(SparkDropDownListBase から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。(SparkListBase から継承されました)
SelectIndex	ユーザーがリスト上の項目をクリックしたり、キーボードを使って項目を移動したりしたときにディスパッチされます。(SparkList から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkDropDownListBase クラス

説明

SparkDropDownListBase コントロールには、ユーザーが 1 つの値を選択できるドロップダウン リストが含まれています。

継承階層

- [SparkList](#)
 - SparkDropDownListBase
 - [SparkComboBox](#)

- [SparkDropDownList](#)

構文

```
'Declaration
Public Class SparkDropDownListBase _
Inherits SparkList
```

プロパティ

名前	説明
allowMultipleSelection	複数選択が可能かどうか。(SparkList から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableDataContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
caretIndex	現在フォーカスがある項目。(SparkListBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dragEnabled	このコントロールの外に項目をドラッグして他のコントロールにドロップすることができるかどうかを示すフラグ。(SparkList から継承されました)
dragMoveEnabled	ドラッグ&ドロップ操作としてコントロールから項目をコピーするのではなく、移動することができるかどうかを示すフラグ。(SparkList から継承されました)
dropEnabled	ドラッグした項目をコントロール上にドロップできるかどうかを示すフラグ。(SparkList から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)

名前	説明
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<i>isDropDownOpen</i>	ドロップダウンが開いているかどうか。
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。 (<i>SparkListBase</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>requireSelection</i>	コントロール内でデータ項目が常に選択されている必要があるかどうか。 (<i>SparkListBase</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selectedIndex</i>	選択された項目の 0 を基準としたインデックス。項目が選択されていない場合は -1 です。 (<i>SparkListBase</i> から継承されました)

名前	説明
selectedIndices	現在選択されている項目のインデックスを表すリスト。 (SparkList から継承されました)
selectedItem	現在選択されている項目。 (SparkListBase から継承されました)
selectedItems	現在選択されている項目を表すオブジェクトのリスト。 (SparkList から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
useVirtualLayout	このコントロールに関連付けられているレイアウトの useVirtualLayout プロパティの値。 (SparkListBase から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。 キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。 それぞれのピクセルの 4 バイト目 (アルファ チャンネル)

名前	説明
	はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetItemsCount	ComboBox の項目の数を取得します。 (SparkListBase から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
Open	ユーザーがドロップダウン リストを表示するためにドロップダウン ボタンをクリックするときにディスパッチされます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。(SparkListBase から継承されました)
SelectIndex	ユーザーがリスト上の項目をクリックしたり、キーボードを使って項目を移動したりしたときにディスパッチされます。(SparkList から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkForm クラス

説明

Spark Form コントロールを識別します。

継承階層

- [SparkSkinnableContainer](#)
 - SparkForm

構文

```
'Declaration
Public Class SparkForm _
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)

名前	説明
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>creationPolicy</i>	このコンポーネントの内容の作成ポリシー。 (<i>SparkSkinnableContainer</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>deferredContentCreated</i>	遅延コンテンツが作成されたかどうか。 (<i>SparkSkinnableContainer</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>numElements</i>	このコンテナに含まれるビジュアル要素の数。 (<i>SparkSkinnableContainer</i> から継承されました)

名前	説明
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。 デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (<i>FlexObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkFormItem クラス

説明

Spark FormItem コントロールを識別します。

継承階層

- [SparkSkinnableContainer](#)
 - SparkFormItem

構文

```
'Declaration
Public Class SparkFormItem _
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)

名前	説明
<i>creationPolicy</i>	このコンポーネントの内容の作成ポリシー。 (<i>SparkSkinnableContainer</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(<i>FlexObject</i> から継承されました)
<i>deferredContentCreated</i>	遅延コンテンツが作成されたかどうか。 (<i>SparkSkinnableContainer</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<i>label</i>	FormItem のテキスト ラベルです。
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>numElements</i>	このコンテナに含まれるビジュアル要素の数。 (<i>SparkSkinnableContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)

名前	説明
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>required</i>	FormItem がフォームを送信する値を持たなければならないかどうか。
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>sequenceLabel</i>	フォーム内のフォーム項目の数です。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
<i>Value</i>	コントロールの値（たとえば、テキストコントロールのテキスト）。(TestObject から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
<i>width</i>	表示オブジェクトの幅（ピクセル単位）。(FlexDisplayObject から継承されました)
<i>x</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
<i>y</i>	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkGroup クラス

説明

視覚的な子である、ビジュアル コンポーネントとグラフィック コンポーネントの両方を管理する場合に、このコンテナーを使用します。

継承階層

- [SparkGroupBase](#)
 - SparkGroup
 - [SparkDataRenderer](#)
 - [SparkTileGroup](#)

構文

```
'Declaration
Public Class SparkGroup _
Inherits SparkGroupBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkGroupBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<i>blendMode</i>	使用するブレンド モードを指定する BlendMode クラスの値。
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<i>clipAndEnableScrolling</i>	ビューポートの境界に子をクリップするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。 (<i>SparkGroupBase</i> から継承されました)
<i>contentHeight</i>	ビューポートの内容の高さ。 (<i>SparkGroupBase</i> から継承されました)
<i>contentWidth</i>	ビューポートの内容の幅。 (<i>SparkGroupBase</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。 (<i>SparkGroupBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalScrollPosition</i>	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。 (<i>SparkGroupBase</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkGroupBase</i> から継承されました)
<i>luminosityClip</i>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<i>luminosityInvert</i>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。 (<i>SparkGroupBase</i> から継承されました)
<i>maskType</i>	マスクの種類。 (<i>SparkGroupBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <i>mouseEnabled</i> を false に設定すると、インスタンスでは、マウ

名前	説明
	ス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)
mouseEnabledWhereTransparent	Group の境界全体が、クリックやロールオーバーなどのマウス イベントに応答するかどうかを示す値。(SparkGroupBase から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkGroupBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkGroupBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkGroupBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizeMode	このコンテナの <code>ResizeMode</code> 。(SparkGroupBase から継承されました)
scaleGridBottom	スケールグリッドの下座標。
scaleGridLeft	スケールグリッドの左座標。
scaleGridRight	スケールグリッドの右座標。
scaleGridTop	スケールグリッドの上座標。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)

名前	説明
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。デフォルト値は (0,0) であり、コンポーネントの左上隅。(SparkGroupBase から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkGroupBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkGroupBase から継承されました)

名前	説明
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkGroupBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkGroupBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナ上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。(SparkGroupBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。(SparkGroupBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkGroupBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkGroupBase クラス

説明

GroupBase クラス はビジュアル要素を表示するコンポーネントの基本クラスです。

継承階層

- [FlexObject](#)
 - SparkGroupBase
 - [SparkDataGroup](#)
 - [SparkGroup](#)

構文

```
'Declaration  
Public Class SparkGroupBase _  
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipAndEnableScrolling	ビューポートの境界に子をクリップするかどうか。
columnNames	データ内のすべての列の名前を含むリスト。
contentHeight	ビューポートの内容の高さ。
contentWidth	ビューポートの内容の幅。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>horizontalScrollPosition</code>	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。
<code>luminosityClip</code>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。
<code>luminosityInvert</code>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。
<code>maskType</code>	マスクの種類。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>mouseEnabledWhereTransparent</code>	Group の境界全体が、クリックやロール オーバーなどのマウス イベントに応答するかどうかを示す値。
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。
<code>numElements</code>	このコンテナに含まれるビジュアル要素の数。
<code>numRows</code>	利用可能なデータ中の行の合計数。
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<code>resizeMode</code>	このコンテナの ResizeMode。
<code>scaleX</code>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)

名前	説明
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。デフォルト値は (0,0) であり、コンポーネントの左上隅。
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナ上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 ToString 	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
 Type 	ユーザーがキーを押したときにディスパッチされます。
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイ

名前	説明
	ムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SparkImage クラス

説明

Spark Image コントロールを識別します。

継承階層

- [SparkObject](#)
 - SparkImage

構文

```
'Declaration
Public Class SparkImage _
Inherits SparkObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
bytesLoaded	既にロードされたイメージのバイト数です。
bytesTotal	ロード済みまたはロード保留中のイメージ データの総バイト数です。
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
clearOnLoad	新しいコンテンツをロードする前に、前のイメージ コンテンツをクリアするかどうかを示します。
contentLoaderGrouping	関連付けられた IContentLoader インスタンスの load() メソッドに渡される、オプション コンテンツのグループ化識別子です。
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)

名前	説明
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>fillMode</i>	次元内でのビットマップ塗りつぶしの方法を決定します。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>horizontalAlign</i>	縦横比が 1 対 1 でなく、scaleMode が BitmapScaleMode.LETTERBOX に設定されているときの、コンテンツの水平方向の配置です。
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>preliminaryHeight</i>	イメージの境界が "measured" となるようレイアウトで要求されているものの、イメージデータのロードがまだ完了していない場合に、高さを使用する予測を示します。
<i>preliminaryWidth</i>	イメージの境界が "measured" となるようレイアウトで要求されているものの、イメージデータのロードがまだ完了していない場合に、幅を使用する予測を示します。
<i>scaleMode</i>	fillMode が BitmapFillMode.SCALE に設定されている場合にイメージを伸縮する方法を指定します。
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)

名前	説明
smooth	ビットマップ イメージにスムージング アルゴリズムを適用するかどうかを指定します。
source	コンテンツの場所です。
sourceHeight	伸縮されていない元のイメージ データの高さを示します。
sourceWidth	伸縮されていない元のイメージ データの幅を示します。
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
trustedSource	現在ロードされているコンテンツを、セキュリティ ポリシーでクロスドメインのイメージ アクセスが許可されているソースからロードされたものと見なすかどうかを示すフラグです。
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
verticalAlign	縦横比が 1 対 1 でなく、scaleMode が BitmapScaleMode.LETTERBOX に設定されているときの、コンテンツの垂直方向の配置です。
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	<p>現在のフォーカスを変更します。 (FlexObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
GetStyle	<p>指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)</p>

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkLabel クラス

説明

Label は、単一行または複数行のテキストを均一のフォーマットでレンダリングできる低レベルの UIComponent です。

継承階層

- [SparkTextBase](#)
 - SparkLabel

構文

```
'Declaration  
Public Class SparkLabel _  
Inherits SparkTextBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
isTruncated	テキストが切り詰められたかどうか。 (SparkTextBase から継承されました)

名前	説明
maxDisplayedLines	テキストが切り詰められるかどうか、および切り詰められる位置を指定する整数。 (SparkTextBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
text	このテキスト コンポーネントによって表示されるテキスト。 (SparkTextBase から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)

名前	説明
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。(SparkTextBase から継承されました)
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。(SparkTextBase から継承されました)
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。(SparkTextBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プ

名前	説明
	ロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkTextBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkList クラス

説明

List コントロールでは、項目を垂直方向に並べて表示します。

継承階層

- [SparkListBase](#)
 - SparkList
 - [SparkDropDownListBase](#)

構文

```
'Declaration
Public Class SparkList _
Inherits SparkListBase
```

プロパティ

名前	説明
allowMultipleSelection	複数選択が可能かどうか。
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableDataContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
caretIndex	現在フォーカスがある項目。(SparkListBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
dragEnabled	このコントロールの外に項目をドラッグして他のコントロールにドロップすることができるかどうかを示すフラグ。
dragMoveEnabled	ドラッグ&ドロップ操作としてコントロールから項目をコピーするのではなく、移動することができるかどうかを示すフラグ。
dropEnabled	ドラッグした項目をコントロール上にドロップできるかどうかを示すフラグ。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
labelField	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。(SparkListBase から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject イン

名前	説明
	スタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナーに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
requireSelection	コントロール内でデータ項目が常に選択されている必要があるかどうか。(SparkListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	選択された項目の 0 を基準としたインデックス。項目が選択されていない場合は -1 です。(SparkListBase から継承されました)
selectedIndices	現在選択されている項目のインデックスを表すリスト。
selectedItem	現在選択されている項目。(SparkListBase から継承されました)
selectedItems	現在選択されている項目を表すオブジェクトのリスト。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は <code>Nothing</code> です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
useVirtualLayout	このコントロールに関連付けられているレイアウトの useVirtualLayout プロパティの値。(SparkListBase から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemCount	ComboBox の項目の数を取得します。(SparkListBase から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。(SparkListBase から継承されました)
SelectIndex	ユーザーがリスト上の項目をクリックしたり、キーボードを使って項目を移動したりしたときにディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkListBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkListBase クラス

説明

ListBase クラス は選択をサポートするすべてのコンポーネントの基本クラスです。

継承階層

- [SparkSkinnableDataContainer](#)
 - SparkListBase
 - [SparkButtonBar](#)
 - [SparkList](#)
 - [SparkTabBar](#)

構文

```
'Declaration  
Public Class SparkListBase _  
Inherits SparkSkinnableDataContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。 (FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。 (SparkSkinnableDataContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
caretIndex	現在フォーカスがある項目。
className	Button のようなこのインスタンスのクラスの名前。 (FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。 (SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。 (FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)

名前	説明
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
labelField	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。
lastVisibleRow	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
requireSelection	コントロール内でデータ項目が常に選択されている必要があるかどうか。
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selectedIndex	選択された項目の 0 を基準としたインデックス。 項目が選択されていない場合は -1 です。

名前	説明
selectedItem	現在選択されている項目。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
useVirtualLayout	このコントロールに関連付けられているレイアウトの useVirtualLayout プロパティの値。
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)

名前	説明
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemsCount	ComboBox の項目の数を取得します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)

名前	説明
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オ

名前	説明
	プシオン OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkListLabel クラス

説明

単純なレンダリング オブジェクトです。

継承階層

- [SparkDataRenderer](#)
 - SparkListLabel
 - [SparkComplexDisplay](#)

構文

```
'Declaration
Public Class SparkListLabel _
Inherits SparkDataRenderer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkGroupBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
automationValue	この項目を一意に識別するのに使用できる値。
blendMode	使用するブレンド モードを指定する BlendMode クラスの値。(SparkGroup から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipAndEnableScrolling	ビューポートの境界に子をクリップするかどうか。(SparkGroupBase から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkGroupBase から継承されました)

名前	説明
<i>contentHeight</i>	ビューポートの内容の高さ。(SparkGroupBase から継承されました)
<i>contentWidth</i>	ビューポートの内容の幅。(SparkGroupBase から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>dragging</i>	SparkListLabel がドラッグ モード中かどうかの値。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(SparkGroupBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>horizontalScrollPosition</i>	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。(SparkGroupBase から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>label</i>	SparkListLabel のラベル。
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(SparkGroupBase から継承されました)
<i>luminosityClip</i>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。(SparkGroupBase から継承されました)
<i>luminosityInvert</i>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。(SparkGroupBase から継承されました)
<i>maskType</i>	マスクの種類。(SparkGroupBase から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)

名前	説明
<code>mouseEnabledWhereTransparent</code>	Group の境界全体が、クリックやロールオーバーなどのマウス イベントに応答するかどうかを示す値。(SparkGroupBase から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。(SparkGroupBase から継承されました)
<code>numElements</code>	このコンテナに含まれるビジュアル要素の数。(SparkGroupBase から継承されました)
<code>numRows</code>	利用可能なデータ中の行の合計数。(SparkGroupBase から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<code>resizeMode</code>	このコンテナの ResizeMode。(SparkGroupBase から継承されました)
<code>scaleGridBottom</code>	スケールグリッドの下座標。(SparkGroup から継承されました)
<code>scaleGridLeft</code>	スケールグリッドの左座標。(SparkGroup から継承されました)
<code>scaleGridRight</code>	スケールグリッドの右座標。(SparkGroup から継承されました)
<code>scaleGridTop</code>	スケールグリッドの上座標。(SparkGroup から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<code>selected</code>	SparkListLabel が選択されているかどうか。
<code>showsCaret</code>	caret が表示されるかどうか。
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<code>Text</code>	コントロールのテキスト。(TestObject から継承されました)
<code>themeColor</code>	コンポーネントのテーマ色。(FlexObject から継承されました)
<code>toolTip</code>	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。デフォルト値は (0,0) であり、コンポーネントの左上隅。(SparkGroupBase から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkGroupBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkGroupBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkGroupBase から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkGroupBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	このコンテナ上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。 (SparkGroupBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。 (SparkGroupBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkGroupBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkMuteButton クラス

説明

MuteButton クラスは、VideoPlayer コントロールによって使用されるミュート ボタンを定義します。

継承階層

- [SparkButton](#)
 - SparkMuteButton

構文

```
'Declaration  
Public Class SparkMuteButton _  
Inherits SparkButton
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に buttonDown イベントを繰り返しディスパッチするかどうか。(SparkButtonBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
emphasized	フォーカス マネージャーによって要求されたおりのデフォルト ボタンかどうか。(SparkButton から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)

名前	説明
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	ButtonBase コントロールに表示されるテキスト。 (<i>SparkButtonBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>muted</i>	コントロールがミュートされているかどうか。
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>stickyHighlighting</i>	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。 (<i>SparkButtonBase</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)
<i>useHandCursor</i>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の

名前	説明
	場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
volume	コントロールのボリューム。
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
MuteChange	コントロールのミュートを切り替えます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkNavigatorContent クラス

説明

NavigatorContent クラスは、ViewStack、TabNavigator、Accordion などの MX ナビゲーター コンテナで使用できる Spark コンテナを定義します。

継承階層

- [SparkSkinnableContainer](#)
 - SparkNavigatorContent

構文

```
'Declaration
Public Class SparkNavigatorContent _
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>autoLayout</i>	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
<i>creationPolicy</i>	このコンポーネントの内容の作成ポリシー。(SparkSkinnableContainer から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>deferredContentCreated</i>	遅延コンテンツが作成されたかどうか。(SparkSkinnableContainer から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタ

名前	説明
	スの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkSkinnableContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。 <code>useHandCursor</code> が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)

名前	説明
<i>x</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの x 座標。 (<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの y 座標。 (<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。 (<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>DragCancel</i>	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragDrop</i>	ドラッグした項目がドロップされた場合にディスパッチされます。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragStart</i>	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。 (<i>TestObject</i> から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プ

名前	説明
	ロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkNumericStepper クラス

説明

NumericStepper コントロールを使用すると、一定の間隔で増減する数値を選択できます。NumericStepper は Spinner コンポーネントと同じ機能を提供しますが、コンポーネントの値を、コントロールの矢印ボタンを使用して変更するのではなく直接編集できるように、TextInput コントロールを追加します。

継承階層

- [SparkSpinner](#)
 - SparkNumericStepper

構文

```
'Declaration
Public Class SparkNumericStepper _
Inherits SparkSpinner
```

プロパティ

名前	説明
allowValueWrap	現在の値が最大または最小値のいずれかの場合のステップに対するコントロールの動作。(SparkSpinner から継承されました)
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。 (FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
enableIME	コンポーネントにフォーカスがあるときに IME を有効にするかどうかの値。
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
imeMode	IME (Input Method Editor) モード。
maxChars	フィールドに入力できる最大文字数。
maximum	有効な最大値。(SparkRange から継承されました)
minimum	有効な最小値。(SparkRange から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)

名前	説明
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。（ FlexObject から継承されました）
scaleX	水平方向の倍率を指定する係数。（ FlexObject から継承されました）
scaleY	垂直方向の倍率を指定する係数。（ FlexObject から継承されました）
snapInterval	snapInterval。（ SparkRange から継承されました）
stepSize	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。（ SparkRange から継承されました）
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。（ FlexDisplayObject から継承されました）
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。（ FlexDisplayObject から継承されました）
tabIndex	SWF ファイル内のオブジェクトのタブ順序。（ FlexDisplayObject から継承されました）
Text	コントロールのテキスト。（ TestObject から継承されました）
themeColor	コンポーネントのテーマ色。（ FlexObject から継承されました）
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。（ FlexObject から継承されました）
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。（ FlexObject から継承されました）
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。（ TestObject から継承されました）
value	この範囲の現在の値。（ SparkRange から継承されました）
visible	表示オブジェクトが可視かどうか。（ FlexDisplayObject から継承されました）
width	表示オブジェクトの幅（ピクセル単位）。（ FlexDisplayObject から継承されました）
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。（ FlexDisplayObject から継承されました）
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。（ FlexDisplayObject から継承されました）

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Change	ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。 (SparkSpinner から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
SelectText	テキストが選択された場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkSpinner から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkObject クラス

説明

SparkObject クラス はすべての Spark コンポーネントの基本クラスを定義します。

継承階層

- [FlexObject](#)
 - SparkObject
 - [SparkButtonBase](#)
 - [SparkImage](#)
 - [SparkRange](#)
 - [SparkRichEditableText](#)
 - [SparkSkinnableContainerBase](#)
 - [SparkSkinnableTextBase](#)
 - [SparkVideoPlayer](#)

構文

```
'Declaration
Public Class SparkObject _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<code>className</code>	<code>Button</code> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<code>currentState</code>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は <code>true</code> であり、これは表示リスト上の <code>InteractiveObject</code> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は <code>NaN</code> です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは <code>NaN</code> にリセットされます。 (<i>FlexObject</i> から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は <code>true</code> です。 (<i>FlexDisplayObject</i> から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkPanel クラス

説明

Panel クラスは、タイトルバー、キャプション、境界線、および子のコンテンツ領域を含むコンテナを定義します。

継承階層

- [SparkSkinnableContainer](#)
 - SparkPanel
 - [SparkTitleWindow](#)

構文

```
'Declaration
Public Class SparkPanel _
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(<i>FlexDisplayObject</i> から継承されました)
<i>autoLayout</i>	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(<i>SparkSkinnableContainer</i> から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。(<i>FlexObject</i> から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>controlBarVisible</i>	コントロールバーが可視かどうか。
<i>creationPolicy</i>	このコンポーネントの内容の作成ポリシー。(<i>SparkSkinnableContainer</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(<i>FlexObject</i> から継承されました)
<i>deferredContentCreated</i>	遅延コンテンツが作成されたかどうか。(<i>SparkSkinnableContainer</i> から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(<i>FlexObject</i> から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味しま

名前	説明
	す。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkSkinnableContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
title	タイトルバーに表示されるタイトルまたはキャプション。
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)

名前	説明
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkPopUpAnchor クラス

説明

Spark PopUpAnchor コントロールを識別します。

継承階層

- [FlexObject](#)
 - SparkPopUpAnchor

構文

```
'Declaration
Public Class SparkPopUpAnchor _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)

名前	説明
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。 (<i>FlexObject</i> から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。 (<i>FlexObject</i> から継承されました)
<i>className</i>	<i>Button</i> のようなこのインスタンスのクラスの名前。(<i>FlexObject</i> から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(<i>FlexObject</i> から継承されました)
<i>displayPopUp</i>	ポップアップが表示されるかどうかを指定します。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(<i>FlexObject</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。(<i>FlexObject</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<i>popUpHeightMatchesAnchorHeight</i>	ポップアップ コントロールの高さが PopUpAnchor の高さと同じ値に設定されるかどうかを指定します。

名前	説明
popUpPosition	ポップアップ コントロールが開かれるときの、PopUpAnchor コンポーネントに対する相対的な位置です。
popUpWidthMatchesAnchorWidth	ポップアップ コントロールの幅が PopUpAnchor の幅と同じ値に設定されるかどうかを指定します。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル)

名前	説明
	<p>はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	<p>現在のフォーカスを変更します。 (FlexObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
GetStyle	<p>指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkRadioButton クラス

説明

RadioButton コンポーネントを使用すると、一連の相互排他的な選択肢の中から 1 つだけをユーザーに選択させることができます。RadioButtonGroup は、同じ groupName プロパティを持つ複数の RadioButton コンポーネントで構成されます。

継承階層

- [SparkToggleButtonBase](#)
 - SparkRadioButton

構文

```
'Declaration  
Public Class SparkRadioButton _  
Inherits SparkToggleButtonBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に buttonDown イベントを繰り返しディスパッチするかどうか。(SparkButtonBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
groupName	この RadioButton コンポーネントが属するグループの名前。または、この RadioButton が RadioButtonGroup コンポーネントによって定

名前	説明
	義されたグループの一部の場合は、RadioButtonGroup コンポーネントの id プロパティの値を指定します。
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	ButtonBase コントロールに表示されるテキスト。 (<i>SparkButtonBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selected</i>	ボタンがダウン状態にあるかどうか。 アップ状態の場合は false です。 (<i>SparkToggleButtonBase</i> から継承されました)
<i>stickyHighlighting</i>	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。 (<i>SparkButtonBase</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)

名前	説明
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パ

名前	説明
<i>Exists</i>	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkRange クラス

説明

Range クラスは、値と minimum および maximum プロパティで定義された許可された範囲を保持します。

継承階層

- [SparkObject](#)
 - SparkRange
 - [SparkSpinner](#)
 - [SparkTrackBase](#)

構文

```
'Declaration  
Public Class SparkRange _  
Inherits SparkObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
maximum	有効な最大値。
minimum	有効な最小値。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)

名前	説明
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
snapInterval	snapInterval。
stepSize	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。 (TestObject から継承されました)
value	この範囲の現在の値。
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkRichEditableText クラス

説明

RichEditableText は、リッチ フォーマット テキストの表示、スクロール、選択および編集を行うための低レベルの UIComponent です。

継承階層

- [SparkObject](#)
 - SparkRichEditableText

構文

```
'Declaration
Public Class SparkRichEditableText _
Inherits SparkObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
clipAndEnableScrolling	ビューポートの境界に子をクリップするかどうか。
columnNames	データ内のすべての列の名前を含むリスト。
contentHeight	テキストの高さ。
contentWidth	テキストの幅。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
displayAsPassword	テキスト フィールドがパスワード テキスト フィールドであるかどうか。
editable	このコントロールのテキストの編集が許可されているかどうかを示すフラグ。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)

名前	説明
enableIME	コンポーネントにフォーカスがあるときに IME を有効にするかどうかを示すフラグ。
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
heightInLines	行数で表したコントロールのデフォルトの高さ
horizontalScrollPosition	テキストを横にスクロールするピクセル数。
id	コンポーネントの ID。 (FlexObject から継承されました)
imeMode	IME (Input Method Editor) モード。
lastVisibleRow	最後の可視の子のインデックス。
maxChars	ユーザーが入力するときに、テキストフィールドに入る最大の文字数。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
multiline	複数行のテキストを入力できるかどうか。
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。
numRows	利用可能なデータ中の行の合計数。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
restrict	ユーザーがテキストフィールドに入力できる文字のセット。

名前	説明
<code>scaleX</code>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<code>selectable</code>	コンテンツをマウスで選択できるかどうか (また、コントロールにキーボード フォーカスがあるときはキーボードで選択できるかどうか) を示すフラグ。
<code>selectionActivePosition</code>	矢印キーで選択範囲が拡張されるときに移動する選択範囲の終端を示す文字位置 (テキスト文字列の先頭を基準とした相対位置)。
<code>selectionAnchorPosition</code>	矢印キーで選択範囲が拡張されるときに固定されている選択範囲の終端を示す文字位置 (テキスト文字列の先頭を基準とした相対位置)。
<code>selectionHighlighting</code>	テキスト選択がハイライトされるかどうかを示す値。
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(<i>FlexDisplayObject</i> から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<code>Text</code>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<code>text</code>	このコンポーネントによって表示されるテキスト文字列。
<code>themeColor</code>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<code>toolTip</code>	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(<i>FlexObject</i> から継承されました)
<code>useHandCursor</code>	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。 <code>useHandCursor</code> が false の場合、代わりに矢印カーソルが表示されます。(<i>FlexObject</i> から継承されました)
<code>Value</code>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<code>verticalScrollPosition</code>	テキストを縦にスクロールするピクセル数。
<code>visible</code>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<code>width</code>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<code>widthInChars</code>	em 単位で測定されたコントロールのデフォルトの幅。
<code>x</code>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<code>y</code>	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナ上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SelectText	テキストが選択された場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkRichText クラス

説明

RichText は、単一行または複数行のリッチ フォーマット テキストや埋め込み画像を表示する低レベルの UIComponent です。

継承階層

- [SparkTextBase](#)
 - SparkRichText

構文

```
'Declaration
Public Class SparkRichText _
Inherits SparkTextBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<code>className</code>	<code>Button</code> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<code>currentState</code>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>isTruncated</code>	テキストが切り詰められたかどうか。 (<i>SparkTextBase</i> から継承されました)
<code>luminosityClip</code>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。
<code>luminosityInvert</code>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値を計算するかどうか。
<code>maskType</code>	コンポーネントにマスクを適用する方法。 有効な値は、 <code>MaskType.CLIP</code> 、 <code>MaskType.ALPHA</code> 、および <code>MaskType.LUMINOSITY</code> です。 クリップ マスク。 クリップ モードでマスクを適用すると、クリップ マスクが 1 ビットに減らされます。
<code>maxDisplayedLines</code>	テキストが切り詰められるかどうか、および切り詰められる位置を指定する整数。 (<i>SparkTextBase</i> から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は <code>true</code> であり、これは表示リスト上の <code>InteractiveObject</code> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は <code>NaN</code> です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、こ

名前	説明
	のプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
text	このテキスト コンポーネントによって表示されるテキスト。(SparkTextBase から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル)

名前	説明
	<p>はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	<p>現在のフォーカスを変更します。 (FlexObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
DragCancel	<p>このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。 (SparkTextBase から継承されました)</p>
DragDrop	<p>このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。 (SparkTextBase から継承されました)</p>
DragStart	<p>このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。 (SparkTextBase から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkTextBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkScrollBar クラス

説明

ScrollBarBase クラスは、表示領域に対してデータが多すぎる場合に表示するデータの部分を制御するために使用します。

継承階層

- [SparkTrackBase](#)
 - SparkScrollBar

構文

```
'Declaration
Public Class SparkScrollBar _
Inherits SparkTrackBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)

名前	説明
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
maximum	有効な最大値。 (SparkRange から継承されました)
minimum	有効な最小値。 (SparkRange から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
pageSize	changeValueByPage() メソッドを呼び出したときに value プロパティの値における変更。
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
snapInterval	snapInterval。 (SparkRange から継承されました)
stepSize	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。 (SparkRange から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TextObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
value	この範囲の現在の値。(SparkRange から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Change	ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。(SparkTrackBase から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkTrackBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkSkinnableContainer クラス

説明

SkinnableContainer クラスは、ビジュアル コンテンツを格納するスキン可能コンテナの基本クラスです。

継承階層

- [SparkSkinnableContainerBase](#)
 - SparkSkinnableContainer
 - [SparkApplication](#)
 - [SparkBorderContainer](#)
 - [SparkForm](#)
 - [SparkFormItem](#)
 - [SparkNavigatorContent](#)
 - [SparkPanel](#)

- [SparkSkinnablePopUpContainer](#)
- [SparkWindow](#)

構文

```
'Declaration
Public Class SparkSkinnableContainer _
Inherits SparkSkinnableContainerBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
creationPolicy	このコンポーネントの内容の作成ポリシー。
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
deferredContentCreated	遅延コンテンツが作成されたかどうか。
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)

名前	説明
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、 <code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkSkinnableContainerBase クラス

説明

Spark のコンテナ コンポーネントの基本クラスです。

継承階層

- [SparkObject](#)
 - SparkSkinnableContainerBase
 - [SparkDataGrid](#)
 - [SparkSkinnableContainer](#)
 - [SparkSkinnableDataContainer](#)

構文

```
'Declaration
Public Class SparkSkinnableContainerBase _
Inherits SparkObject
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。
<i>numRows</i>	利用可能なデータ中の行の合計数。

名前	説明
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅（ピクセル単位）。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。 (TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。 (TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
 ToString 	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkSkinnableDataContainer クラス

説明

SkinnableDataContainer クラスはデータ項目の基本コンテナ クラスです。

継承階層

- [SparkSkinnableContainerBase](#)
 - SparkSkinnableDataContainer
 - [SparkListBase](#)

構文

```
'Declaration
Public Class SparkSkinnableDataContainer _
Inherits SparkSkinnableContainerBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>numRows</code>	利用可能なデータ中の行の合計数。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<code>tabIndex</code>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)

名前	説明
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SparkSkinnablePopUpContainer クラス

説明

SparkSkinnablePopUpContainer コントロールを識別します。

継承階層

- [SparkSkinnableContainer](#)
 - SparkSkinnablePopUpContainer

構文

```
'Declaration  
Public Class SparkSkinnablePopUpContainer _  
Inherits SparkSkinnableContainer
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
creationPolicy	このコンポーネントの内容の作成ポリシー。(SparkSkinnableContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
deferredContentCreated	遅延コンテンツが作成されたかどうか。(SparkSkinnableContainer から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
isOpen	コンテナが開いているかどうか。

名前	説明
lastVisibleRow	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。 (SparkSkinnableContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>DragCancel</i>	ドラッグ操作がキャンセルされた場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragDrop</i>	ドラッグした項目がドロップされた場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragStart</i>	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkSkinnableTextBase クラス

説明

Spark TextInput や TextArea などのスキン可能なコンポーネントの基本クラスです。このようなコンポーネントは、リッチ テキスト表示、スクロール、選択、および編集などの機能を提供するために、スキン内に RichEditableText のインスタンスを含みます。

継承階層

- [SparkObject](#)
 - SparkSkinnableTextBase
 - [SparkTextArea](#)
 - [SparkTextInput](#)

構文

```
'Declaration
Public Class SparkSkinnableTextBase _
Inherits SparkObject
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>displayAsPassword</i>	テキスト フィールドがパスワード テキスト フィールドであるかどうか。
<i>editable</i>	このコントロールのテキストの編集が許可されているかどうかを示すフラグ。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>enableIME</i>	コンポーネントにフォーカスがあるときに IME を有効にするかどうかを示すフラグ。
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>imeMode</i>	IME (Input Method Editor) モード。
<i>lastVisibleRow</i>	最後の可視の子のインデックス。
<i>maxChars</i>	ユーザーが入力するときに、テキスト フィールドに入る最大の文字数。
<i>maxWidth</i>	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject イン

名前	説明
	スタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、 flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナーに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。
numRows	利用可能なデータ中の行の合計数。
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
restrict	ユーザーがテキスト フィールドに入力できる文字のセット。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectable	コンテンツをマウスで選択できるかどうか（また、コントロールにキーボード フォーカスがあるときはキーボードで選択できるかどうか）を示すフラグ。
selectionActivePosition	矢印キーで選択範囲が拡張されるときに移動する選択範囲の終端を示す文字位置（テキスト文字列の先頭を基準とした相対位置）。
selectionAnchorPosition	矢印キーで選択範囲が拡張されるときに固定されている選択範囲の終端を示す文字位置（テキスト文字列の先頭を基準とした相対位置）。
selectionHighlighting	テキスト選択がハイライトされるかどうかを示す値。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
text	このコンポーネントによって表示されるテキスト文字列。
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MouseScroll	このコンテナ上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SelectText	テキストが選択された場合にディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkSlider クラス

説明

スライダー トラックの両端の間でスライダー サムを移動することで値を選択できます。

継承階層

- [SparkTrackBase](#)
 - SparkSlider
 - [SparkVolumeBar](#)

構文

```
'Declaration
Public Class SparkSlider _
Inherits SparkTrackBase
```


プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>maximum</i>	有効な最大値。(SparkRange から継承されました)
<i>minimum</i>	有効な最小値。(SparkRange から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、こ

名前	説明
	のプロパティは NaN にリセットされます。(FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
snapInterval	snapInterval。(SparkRange から継承されました)
stepSize	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。(SparkRange から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
value	この範囲の現在の値。(SparkRange から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Change	<p>ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。 (SparkTrackBase から継承されました)</p>
ChangeFocus	<p>現在のフォーカスを変更します。 (FlexObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkTrackBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkSpinner クラス

説明

Spinner コンポーネントは順序化された集合から値を選択します。 `stepSize` プロパティの現在の値に基づいて、現在の値を増減させる 2 つのボタンを使用します。

継承階層

- [SparkRange](#)
 - SparkSpinner
 - [SparkNumericStepper](#)

構文

```
'Declaration
Public Class SparkSpinner _
Inherits SparkRange
```

プロパティ

名前	説明
allowValueWrap	現在の値が最大または最小値のいずれかの場合のステップに対するコントロールの動作。
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<code>Button</code> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
id	コンポーネントの ID。 (FlexObject から継承されました)
maximum	有効な最大値。 (SparkRange から継承されました)
minimum	有効な最小値。 (SparkRange から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
snapInterval	snapInterval。 (SparkRange から継承されました)
stepSize	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。 (SparkRange から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)

名前	説明
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
value	この範囲の現在の値。(SparkRange から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Change	ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウスボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの

名前	説明
WaitForDisappearance	'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkTabBar クラス

説明

TabBar クラスは同じようなタブのセットを表示します。1 回に 1 つのタブを選択でき、デフォルトでは最初のタブが選択されます。

継承階層

- [SparkListBase](#)
 - SparkTabBar

構文

```
'Declaration
Public Class SparkTabBar _
Inherits SparkListBase
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>autoLayout</i>	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableDataContainer から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>caretIndex</i>	現在フォーカスがある項目。(SparkListBase から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>labelField</i>	ラベルとして表示するデータ プロバイダー項目内のフィールドの名前。(SparkListBase から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。(SparkSkinnableContainerBase から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべて

名前	説明
	の子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
requireSelection	コントロール内でデータ項目が常に選択されている必要があるかどうか。(SparkListBase から継承されました)
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
selectedIndex	選択された項目の 0 を基準としたインデックス。項目が選択されていない場合は -1 です。(SparkListBase から継承されました)
selectedItem	現在選択されている項目。(SparkListBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は <code>Nothing</code> です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は <code>true</code> です。 <code>useHandCursor</code> が <code>false</code> の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
useVirtualLayout	このコントロールに関連付けられているレイアウトの <code>useVirtualLayout</code> プロパティの値。(SparkListBase から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>visible</i>	表示オブジェクトが可視かどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>width</i>	表示オブジェクトの幅 (ピクセル単位)。(<i>FlexDisplayObject</i> から継承されました)
<i>x</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの x 座標。(<i>FlexDisplayObject</i> から継承されました)
<i>y</i>	親 <i>DisplayObjectContainer</i> のローカル座標を基準にした <i>DisplayObject</i> インスタンスの y 座標。(<i>FlexDisplayObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>ChangeFocus</i>	現在のフォーカスを変更します。(<i>FlexObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>DragCancel</i>	ドラッグ操作がキャンセルされた場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragDrop</i>	ドラッグした項目がドロップされた場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>DragStart</i>	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemsCount	ComboBox の項目の数を取得します。(SparkListBase から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	リスト中の項目が選択された場合にディスパッチされます。 (SparkListBase から継承されました)
SelectIndex	ユーザーがリスト上の項目をクリックしたり、キーボードを使って項目を移動したりしたときにディスパッチされます。
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkListBase から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SparkTextArea クラス

説明

TextArea は、ユーザーが複数行のリッチ フォーマット テキストを入力および編集するとき使用するテキスト入力コントロールです。 テキストをスクロールするための水平方向と垂直方向のスクロールバーを表示し、マウス ホイールによる垂直スクロールをサポートします。

継承階層

- [SparkSkinnableTextBase](#)

- SparkTextArea

構文

```
'Declaration
Public Class SparkTextArea _
Inherits SparkSkinnableTextBase
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(SparkSkinnableTextBase から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>displayAsPassword</i>	テキスト フィールドがパスワード テキスト フィールドであるかどうか。(SparkSkinnableTextBase から継承されました)
<i>editable</i>	このコントロールのテキストの編集が許可されているかどうかを示すフラグ。(SparkSkinnableTextBase から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>enableIME</i>	コンポーネントにフォーカスがあるときに IME を有効にするかどうかを示すフラグ。(SparkSkinnableTextBase から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(SparkSkinnableTextBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>heightInLines</i>	行数で表したコントロールのデフォルトの高さ

名前	説明
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>imeMode</i>	IME (Input Method Editor) モード。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>maxChars</i>	ユーザーが入力するときに、テキスト フィールドに入る最大の文字数。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>maxWidth</i>	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナーに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>restrict</i>	ユーザーがテキスト フィールドに入力できる文字のセット。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selectable</i>	コンテンツをマウスで選択できるかどうか (また、コントロールにキーボード フォーカスがあるときはキーボードで選択できるかどうか) を示すフラグ。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>selectionActivePosition</i>	矢印キーで選択範囲が拡張されるときに移動する選択範囲の終端を示す文字位置 (テキスト文字列の先頭を基準とした相対位置)。 (<i>SparkSkinnableTextBase</i> から継承されました)

名前	説明
selectionAnchorPosition	矢印キーで選択範囲が拡張されるときに固定されている選択範囲の終端を示す文字位置 (テキスト文字列の先頭を基準とした相対位置)。(SparkSkinnableTextBase から継承されました)
selectionHighlighting	テキスト選択がハイライトされるかどうかを示す値。(SparkSkinnableTextBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
text	このコンポーネントによって表示されるテキスト文字列。(SparkSkinnableTextBase から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
widthInChars	em 単位で測定されたコントロールのデフォルトの幅。
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル)

名前	説明
	はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 (SparkSkinnableTextBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。 (SparkSkinnableTextBase から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	このコンテナー上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。 (SparkSkinnableTextBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SelectText	テキストが選択された場合にディスパッチされます。(SparkSkinnableTextBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkSkinnableTextBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForObject	トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkTextBase クラス

説明

デフォルト形式の CSS スタイルを使用してテキストを表示する Label や RichText などの Spark テキストコントロール用の基本クラスです。

継承階層

- [FlexObject](#)
 - SparkTextBase
 - [SparkLabel](#)
 - [SparkRichText](#)

構文

```
'Declaration
Public Class SparkTextBase _
Inherits FlexObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名称。(FlexObject から継承されました)

名前	説明
<code>currentState</code>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<code>id</code>	コンポーネントの ID。(FlexObject から継承されました)
<code>isTruncated</code>	テキストが切り詰められたかどうか。
<code>maxDisplayedLines</code>	テキストが切り詰められるかどうか、および切り詰められる位置を指定する整数。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)

名前	説明
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
text	このテキスト コンポーネントによって表示されるテキスト。
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。

名前	説明
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SparkTextInput クラス

説明

TextInput は、単一行のテキストを均一のフォーマットで入力および編集するときに使用するテキスト入力コントロールです。

継承階層

- [SparkSkinnableTextBase](#)

- SparkTextInput

構文

```
'Declaration
Public Class SparkTextInput _
Inherits SparkSkinnableTextBase
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>columnNames</i>	データ内のすべての列の名前を含むリスト。(SparkSkinnableTextBase から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>displayAsPassword</i>	テキスト フィールドがパスワード テキスト フィールドであるかどうか。(SparkSkinnableTextBase から継承されました)
<i>editable</i>	このコントロールのテキストの編集が許可されているかどうかを示すフラグ。(SparkSkinnableTextBase から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>enableIME</i>	コンポーネントにフォーカスがあるときに IME を有効にするかどうかを示すフラグ。(SparkSkinnableTextBase から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>firstVisibleRow</i>	最初の可視の子のインデックス。(SparkSkinnableTextBase から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)

名前	説明
<i>imeMode</i>	IME (Input Method Editor) モード。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>lastVisibleRow</i>	最後の可視の子のインデックス。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>maxChars</i>	ユーザーが入力するときに、テキスト フィールドに入る最大の文字数。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>maxWidth</i>	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>restrict</i>	ユーザーがテキスト フィールドに入力できる文字のセット。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selectable</i>	コンテンツをマウスで選択できるかどうか (また、コントロールにキーボード フォーカスがあるときはキーボードで選択できるかどうか) を示すフラグ。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>selectionActivePosition</i>	矢印キーで選択範囲が拡張されるときに移動する選択範囲の終端を示す文字位置 (テキスト文字列の先頭を基準とした相対位置)。 (<i>SparkSkinnableTextBase</i> から継承されました)
<i>selectionAnchorPosition</i>	矢印キーで選択範囲が拡張されるときに固定されている選択範囲の終端を示す文字位置 (テキスト文字列の先頭を基準とした相対位置)。 (<i>SparkSkinnableTextBase</i> から継承されました)

名前	説明
selectionHighlighting	テキスト選択がハイライトされるかどうかを示す値。 (SparkSkinnableTextBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
text	このコンポーネントによって表示されるテキスト文字列。 (SparkSkinnableTextBase から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
widthInChars	em 単位で測定されたコントロールのデフォルトの幅。
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)

名前	説明
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。 (SparkSkinnableTextBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Input	ユーザーがコントロール内でテキストの入力、削除、貼り付けを行ったときにディスパッチされます。 (SparkSkinnableTextBase から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	このコンテナー上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。 (SparkSkinnableTextBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SelectText	テキストが選択された場合にディスパッチされます。(SparkSkinnableTextBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkSkinnableTextBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForObject	トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkTileGroup クラス

説明

TileGroup コンテナは、TileLayout クラスを使用する Group コンテナのインスタンスです。

継承階層

- [SparkGroup](#)
 - SparkTileGroup

構文

```
'Declaration
Public Class SparkTileGroup _
Inherits SparkGroup
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkGroupBase から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
blendMode	使用するブレンド モードを指定する BlendMode クラスの値。(SparkGroup から継承されました)

名前	説明
<code>className</code>	<code>Button</code> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<code>clipAndEnableScrolling</code>	ビューポートの境界に子をクリップするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<code>columnNames</code>	データ内のすべての列の名前を含むリスト。 (<i>SparkGroupBase</i> から継承されました)
<code>contentHeight</code>	ビューポートの内容の高さ。 (<i>SparkGroupBase</i> から継承されました)
<code>contentWidth</code>	ビューポートの内容の幅。 (<i>SparkGroupBase</i> から継承されました)
<code>currentState</code>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。 (<i>SparkGroupBase</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>horizontalScrollPosition</code>	コンポーネントの座標系における、ビューポートの原点の x 座標。デフォルト値は (0,0) であり、コンポーネントの左上隅に対応します。 (<i>SparkGroupBase</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。 (<i>SparkGroupBase</i> から継承されました)
<code>luminosityClip</code>	マスクされたコンテンツを輝度マスクがクリッピングするかどうか。 (<i>SparkGroupBase</i> から継承されました)
<code>luminosityInvert</code>	輝度マスクによってマスクされているグラフィック要素の RGB カラー値の計算を制御する値。 (<i>SparkGroupBase</i> から継承されました)
<code>maskType</code>	マスクの種類。 (<i>SparkGroupBase</i> から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべて

名前	説明
<i>mouseEnabledWhereTransparent</i>	の子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)
<i>numAutomationChildren</i>	Group の境界全体が、クリックやロールオーバーなどのマウス イベントに応答するかどうかを示す値。(SparkGroupBase から継承されました)
<i>numColumns</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>numElements</i>	利用可能なデータ中の列の合計数。(SparkGroupBase から継承されました)
<i>numRows</i>	このコンテナに含まれるビジュアル要素の数。(SparkGroupBase から継承されました)
<i>percentHeight</i>	利用可能なデータ中の行の合計数。(SparkGroupBase から継承されました)
<i>percentWidth</i>	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
<i>resizeMode</i>	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>scaleGridBottom</i>	このコンテナの ResizeMode。(SparkGroupBase から継承されました)
<i>scaleGridLeft</i>	スケールグリッドの下座標。(SparkGroup から継承されました)
<i>scaleGridRight</i>	スケールグリッドの左座標。(SparkGroup から継承されました)
<i>scaleGridTop</i>	スケールグリッドの右座標。(SparkGroup から継承されました)
<i>scaleX</i>	スケールグリッドの上座標。(SparkGroup から継承されました)
<i>scaleY</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>tabChildren</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>tabEnabled</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabIndex</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
<i>Text</i>	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
<i>themeColor</i>	コントロールのテキスト。(TestObject から継承されました)
<i>toolTip</i>	コンポーネントのテーマ色。(FlexObject から継承されました)
	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
verticalScrollPosition	コンポーネントの座標系における、ビューポートの原点の y 座標を取得します。デフォルト値は (0,0) であり、コンポーネントの左上隅。(SparkGroupBase から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkGroupBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkGroupBase から継承されました)
DragStart	ドラッグ & ドロップ操作が始まった場合にディスパッチされます。(SparkGroupBase から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkGroupBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MouseScroll	このコンテナ上のスクロールバーを動かすためにマウスが使用されたときにディスパッチされます。 (SparkGroupBase から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	コンテナのスクロール イベントが発生したときにディスパッチされます。このイベントは、スクロールがプログラムまたはユーザーによって行われた場合にディスパッチされます。 (SparkGroupBase から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkGroupBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkTitleWindow クラス

説明

TitleWindow クラスは Panel を拡張し、閉じるボタンと移動領域をサポートします。

継承階層

- [SparkPanel](#)
 - SparkTitleWindow

構文

```
'Declaration  
Public Class SparkTitleWindow _  
Inherits SparkPanel
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
columnNames	データ内のすべての列の名前を含むリスト。(SparkSkinnableContainerBase から継承されました)
controlBarVisible	コントロールバーが可視かどうか。(SparkPanel から継承されました)
creationPolicy	このコンポーネントの内容の作成ポリシー。(SparkSkinnableContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
deferredContentCreated	遅延コンテンツが作成されたかどうか。(SparkSkinnableContainer から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)

名前	説明
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (FlexObject から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (FlexDisplayObject から継承されました)
<code>id</code>	コンポーネントの ID。 (FlexObject から継承されました)
<code>lastVisibleRow</code>	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (FlexDisplayObject から継承されました)
<code>numAutomationChildren</code>	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
<code>numColumns</code>	利用可能なデータ中の列の合計数。 (SparkSkinnableContainerBase から継承されました)
<code>numElements</code>	このコンテナに含まれるビジュアル要素の数。 (SparkSkinnableContainer から継承されました)
<code>numRows</code>	利用可能なデータ中の行の合計数。 (SparkSkinnableContainerBase から継承されました)
<code>percentHeight</code>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
<code>percentWidth</code>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
<code>scaleX</code>	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
<code>scaleY</code>	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
<code>tabChildren</code>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (FlexDisplayObject から継承されました)
<code>tabEnabled</code>	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)

名前	説明
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
title	タイトルバーに表示されるタイトルまたはキャプション。 (SparkPanel から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(<i>FlexObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(<i>FlexObject</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkPanel から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkToggleButton クラス

説明

ToggleButton コンポーネントはトグル ボタンを定義します。このボタンをクリックすると、アップ状態とダウン状態が切り替わります。

継承階層

- [SparkToggleButtonBase](#)
 - SparkToggleButton
 - [SparkButtonBarButton](#)

構文

```
'Declaration  
Public Class SparkToggleButton _  
Inherits SparkToggleButtonBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に buttonDown イベントを繰り返しディスパッチするかどうか。(SparkButtonBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)

名前	説明
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>label</i>	ButtonBase コントロールに表示されるテキスト。 (<i>SparkButtonBase</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。 表示リスト上のこのインスタンスの子は影響を受けません。 表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。 指定できる値は 0 から 100 です。 デフォルト値は NaN です。 width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>selected</i>	ボタンがダウン状態にあるかどうか。 アップ状態の場合は false です。 (<i>SparkToggleButtonBase</i> から継承されました)
<i>stickyHighlighting</i>	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。 (<i>SparkButtonBase</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。 デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。 (<i>FlexObject</i> から継承されました)
<i>toolTip</i>	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (<i>FlexObject</i> から継承されました)

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	このイベントは、ドラッグ操作がキャンセルされた場合にトリガーされます。
DragDrop	このイベントは、ドラッグした項目がドロップされた場合にトリガーされます。
DragStart	このイベントは、ドラッグ&ドロップ操作が始まった場合にトリガーされます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkToggleButtonBase クラス

説明

ToggleButtonBase コンポーネントは、selected プロパティをサポートする Spark ボタン コンポーネントの基本クラスです。

継承階層

- [SparkButtonBase](#)
 - SparkToggleButtonBase
 - [SparkCheckBox](#)
 - [SparkRadioButton](#)
 - [SparkToggleButton](#)

構文

```
'Declaration  
Public Class SparkToggleButtonBase _  
Inherits SparkButtonBase
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
autoRepeat	ユーザーがマウス ボタンを押し続けた場合に buttonDown イベントを繰り返しディスパッチするかどうか。(SparkButtonBase から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
label	ButtonBase コントロールに表示されるテキスト。(SparkButtonBase から継承されました)
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)

名前	説明
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。 (FlexObject から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (FlexObject から継承されました)
scaleX	水平方向の倍率を指定する係数。 (FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。 (FlexObject から継承されました)
selected	ボタンがダウン状態にあるかどうか。アップ状態の場合は false です。
stickyHighlighting	ボタンをクリックするとダウン状態のスキンが表示され、ボタンからマウスを離すとオーバー状態のスキンが表示されるかどうかを示す値。 (SparkButtonBase から継承されました)
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。 (FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)

名前	説明
y	親 <code>DisplayObjectContainer</code> のローカル座標を基準にした <code>DisplayObject</code> インスタンスの <code>y</code> 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。 (SparkButtonBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkTrackBase クラス

説明

TrackBase クラスは、Slider や ScrollBar などの、トラックと 1 つ以上のサム ボタンを持つコンポーネントの基本クラスです。

継承階層

- [SparkRange](#)
 - SparkTrackBase
 - [SparkScrollBar](#)
 - [SparkSlider](#)

構文

```
'Declaration
Public Class SparkTrackBase _
Inherits SparkRange
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	<i>Button</i> のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)

名前	説明
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<i>id</i>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<i>maximum</i>	有効な最大値。 (<i>SparkRange</i> から継承されました)
<i>minimum</i>	有効な最小値。 (<i>SparkRange</i> から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。 (<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。 (<i>FlexObject</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。 (<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。 (<i>FlexObject</i> から継承されました)
<i>scaleX</i>	水平方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。 (<i>FlexObject</i> から継承されました)
<i>snapInterval</i>	snapInterval。 (<i>SparkRange</i> から継承されました)
<i>stepSize</i>	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。 (<i>SparkRange</i> から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。 (<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。 (<i>FlexDisplayObject</i> から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
value	この範囲の現在の値。(SparkRange から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Change	ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkVideoDisplay クラス

説明

Spark VideoDisplay コントロールを識別します。

継承階層

- [FlexObject](#)
 - SparkVideoDisplay

構文

```
'Declaration
Public Class SparkVideoDisplay _
Inherits FlexObject
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>autoDisplayFirstFrame</i>	source が設定されている場合、ビデオをロードするかどうかを指定します。
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>autoPlay</i>	source プロパティが設定されている場合に、ビデオの再生をすぐに開始するかどうかを指定します。
<i>autoRewind</i>	再生が停止したときに、FLV ファイルを最初のフレームまで巻き戻すかどうかを指定します。
<i>bytesLoaded</i>	既にアプリケーションにダウンロードされているデータのバイト数です。
<i>bytesTotal</i>	アプリケーションにダウンロードされるデータの総バイト数です。
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>currentTime</i>	ビデオの再生開始以降の再生ヘッドの現在時間 (秒単位) です。
<i>duration</i>	ビデオの再生時間 (秒単位) です。
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>loop</i>	再生が完了した後でメディアを再び再生する必要があるかどうかを示します。
<i>mediaPlayerState</i>	ビデオの現在の状態です。
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject イン

名前	説明
	<p>スタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、<code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(FlexDisplayObject から継承されました)</p>
<code>muted</code>	<p>ビデオをミュートするかどうかを指定します。</p>
<code>numAutomationChildren</code>	<p>このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)</p>
<code>pauseWhenHidden</code>	<p>ビデオが非表示になっても再生を続行するかどうかを制御します。</p>
<code>percentHeight</code>	<p>コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)</p>
<code>percentWidth</code>	<p>コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)</p>
<code>playing</code>	<p>ビデオが再生中である場合、または再生試行中であるかどうか。</p>
<code>scaleMode</code>	<p>ビデオ コンテンツの様々なサイズ変更の方法を示します。</p>
<code>scaleX</code>	<p>水平方向の倍率を指定する係数。(FlexObject から継承されました)</p>
<code>scaleY</code>	<p>垂直方向の倍率を指定する係数。(FlexObject から継承されました)</p>
<code>tabChildren</code>	<p>オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(FlexDisplayObject から継承されました)</p>
<code>tabEnabled</code>	<p>このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)</p>
<code>tabIndex</code>	<p>SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)</p>
<code>Text</code>	<p>コントロールのテキスト。(TestObject から継承されました)</p>
<code>themeColor</code>	<p>コンポーネントのテーマ色。(FlexObject から継承されました)</p>
<code>toolTip</code>	<p>ツール チップに表示されるテキスト。デフォルト値は <code>Nothing</code> です。(FlexObject から継承されました)</p>
<code>useHandCursor</code>	<p>マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。デフォルト値は <code>true</code> です。 <code>useHandCursor</code> が <code>false</code> の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)</p>
<code>Value</code>	<p>コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)</p>
<code>visible</code>	<p>表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)</p>

名前	説明
volume	ボリューム レベル。0 から 1 の範囲の値で指定します。
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。 このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForProperty	<p>値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SparkVideoPlayer クラス

説明

VideoPlayer コントロールは、プログレッシブ ダウンロード、マルチビットレート ストリーミング、およびストリーミング ビデオをサポートするスキニング可能ビデオ プレーヤーです。

継承階層

- [SparkObject](#)
 - SparkVideoPlayer

構文

```
'Declaration
Public Class SparkVideoPlayer _
Inherits SparkObject
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)

名前	説明
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>muted</i>	コントロールがミュートされているかどうか。
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
<i>playing</i>	ビデオが再生中である場合、または再生試行中であるかどうか。
<i>scaleX</i>	水平方向の倍率を指定する係数。(FlexObject から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)

名前	説明
tabIndex	SWF ファイル内のオブジェクトのタブ順序。 (FlexDisplayObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
themeColor	コンポーネントのテーマ色。 (FlexObject から継承されました)
toolTip	ツール チップに表示されるテキスト。 デフォルト値は Nothing です。 (FlexObject から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンド カーソルを表示するかどうか。 デフォルト値は true です。 useHandCursor が false の場合、代わりに矢印カーソルが表示されます。 (FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。 (FlexDisplayObject から継承されました)
volume	ボリューム レベル。 0 から 1 の範囲の値で指定します。
width	表示オブジェクトの幅 (ピクセル単位)。 (FlexDisplayObject から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。 (FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。 (FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パ

名前	説明
<i>Exists</i>	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetStyle</i>	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(<i>FlexDisplayObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。 (FlexObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。 (FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ToString	コントロールの文字列表現を返します。 (FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkVolumeBar クラス

説明

VolumeBar クラスは、VideoDisplay コントロールのボリュームを制御するドロップダウン スライダーを定義します。

継承階層

- [SparkSlider](#)
 - SparkVolumeBar

構文

```
'Declaration
Public Class SparkVolumeBar _
Inherits SparkSlider
```

プロパティ

名前	説明
<i>alpha</i>	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
<i>automationClassName</i>	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
<i>automationIndex</i>	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
<i>automationName</i>	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)
<i>className</i>	Button のようなこのインスタンスのクラスの名前。(FlexObject から継承されました)
<i>currentState</i>	コンポーネントの現在の表示状態。(FlexObject から継承されました)
<i>enabled</i>	コンポーネントがユーザー操作を受け付けることができるかどうか。(FlexObject から継承されました)
<i>errorColor</i>	検証が失敗したときにコンポーネントをハイライトする色。(FlexObject から継承されました)
<i>errorString</i>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
<i>focusEnabled</i>	コンポーネントが選択時にフォーカスを取得できるかどうか。(FlexObject から継承されました)
<i>height</i>	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
<i>id</i>	コンポーネントの ID。(FlexObject から継承されました)
<i>isDropDownOpen</i>	VolumeBar が開いているかどうか。
<i>maximum</i>	有効な最大値。(SparkRange から継承されました)
<i>minimum</i>	有効な最小値。(SparkRange から継承されました)
<i>mouseEnabled</i>	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
<i>muted</i>	コントロールがミュートされているかどうか。
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(FlexObject から継承されました)

名前	説明
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。（ FlexObject から継承されました）
scaleX	水平方向の倍率を指定する係数。（ FlexObject から継承されました）
scaleY	垂直方向の倍率を指定する係数。（ FlexObject から継承されました）
snapInterval	snapInterval。（ SparkRange から継承されました）
stepSize	changeValueByStep() メソッドが呼び出されたときに変更される value プロパティの量。（ SparkRange から継承されました）
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。（ FlexDisplayObject から継承されました）
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。（ FlexDisplayObject から継承されました）
tabIndex	SWF ファイル内のオブジェクトのタブ順序。（ FlexDisplayObject から継承されました）
Text	コントロールのテキスト。（ TestObject から継承されました）
themeColor	コンポーネントのテーマ色。（ FlexObject から継承されました）
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。（ FlexObject から継承されました）
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。（ FlexObject から継承されました）
Value	コントロールの値（たとえば、テキストコントロールのテキスト）。（ TestObject から継承されました）
value	この範囲の現在の値。（ SparkRange から継承されました）
visible	表示オブジェクトが可視かどうか。（ FlexDisplayObject から継承されました）
width	表示オブジェクトの幅（ピクセル単位）。（ FlexDisplayObject から継承されました）
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。（ FlexDisplayObject から継承されました）
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。（ FlexDisplayObject から継承されました）

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Change	ユーザーによる操作の結果として NumericStepper コントロールの値が変更されたときにディスパッチされます。 (SparkTrackBase から継承されました)
ChangeFocus	現在のフォーカスを変更します。 (FlexObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(FlexObject から継承されました)
MuteChange	ユーザーによる操作の結果としてミュートの状態が変更されたときにディスパッチされます。
Open	ユーザーがドロップダウン リストを表示するためにドロップダウン ボタンをクリックするときにディスパッチされます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkTrackBase から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkWindow クラス

説明

Window は、AIR デスクトップ アプリケーションの追加ウィンドウに対する最上位のコンテナです。

継承階層

- [SparkSkinnableContainer](#)
 - SparkWindow

構文

```
'Declaration
Public Class SparkWindow _
Inherits SparkSkinnableContainer _
Implements IMoveable
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alwaysInFront	基になる <code>NativeWindow</code> が常に他のウィンドウの手前に表示されるかどうかを示す値。
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。(FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。(FlexObject から継承されました)

名前	説明
<code>className</code>	<code>Button</code> のようなこのインスタンスのクラスの名前。 (<i>FlexObject</i> から継承されました)
<code>closed</code>	基になるウィンドウが閉じられているかどうかを示す値。
<code>columnNames</code>	データ内のすべての列の名前を含むリスト。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>creationPolicy</code>	このコンポーネントの内容の作成ポリシー。 (<i>SparkSkinnableContainer</i> から継承されました)
<code>currentState</code>	コンポーネントの現在の表示状態。 (<i>FlexObject</i> から継承されました)
<code>deferredContentCreated</code>	遅延コンテンツが作成されたかどうか。 (<i>SparkSkinnableContainer</i> から継承されました)
<code>enabled</code>	コンポーネントがユーザー操作を受け付けることができるかどうか。 (<i>FlexObject</i> から継承されました)
<code>errorColor</code>	検証が失敗したときにコンポーネントをハイライトする色。 (<i>FlexObject</i> から継承されました)
<code>errorString</code>	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。 (<i>FlexObject</i> から継承されました)
<code>firstVisibleRow</code>	最初の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>focusEnabled</code>	コンポーネントが選択時にフォーカスを取得できるかどうか。 (<i>FlexObject</i> から継承されました)
<code>height</code>	表示オブジェクトの高さ (ピクセル単位)。 (<i>FlexDisplayObject</i> から継承されました)
<code>id</code>	コンポーネントの ID。 (<i>FlexObject</i> から継承されました)
<code>isAIRWindow</code>	このウィンドウが AIR ウィンドウかどうかを示す値。
<code>lastVisibleRow</code>	最後の可視の子のインデックス。 (<i>SparkSkinnableContainerBase</i> から継承されました)
<code>maxHeight</code>	親がレイアウトを計算するために使用されるコンポーネントの高さの最大推奨値。
<code>maximizable</code>	ウィンドウを最大化できるかどうかを示す値。
<code>maxWidth</code>	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。
<code>minHeight</code>	親がレイアウトを計算するために使用されるコンポーネントの高さの最小推奨値。
<code>minimizable</code>	ウィンドウを最小化できるかどうかを示す値。
<code>minWidth</code>	親がレイアウトを計算するために使用されるコンポーネントの幅の最小推奨値。
<code>mouseEnabled</code>	このオブジェクトがマウス メッセージを受け取るかどうか。 デフォルト値は <code>true</code> であり、これは表示リスト上の <code>InteractiveObject</code> インスタンスがデフォルトでマウス イベントを受け取ることを意味します。 <code>mouseEnabled</code> を <code>false</code> に設定すると、インスタンスでは、マウ

名前	説明
	イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する <code>mouseEnabled</code> 動作を変更するには、 <code>flash.display.DisplayObjectContainer.mouseChildren</code> を使用します。(<i>FlexDisplayObject</i> から継承されました)
<i>numAutomationChildren</i>	このコンテナに含まれるオートメーションの子の数。(<i>FlexObject</i> から継承されました)
<i>numColumns</i>	利用可能なデータ中の列の合計数。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>numElements</i>	このコンテナに含まれるビジュアル要素の数。(<i>SparkSkinnableContainer</i> から継承されました)
<i>numRows</i>	利用可能なデータ中の行の合計数。(<i>SparkSkinnableContainerBase</i> から継承されました)
<i>percentHeight</i>	コンポーネントの高さを親のサイズに対する比率 (パーセンテージ) で指定する数値。(<i>FlexObject</i> から継承されました)
<i>percentWidth</i>	コンポーネントの幅を親のサイズに対する比率 (パーセンテージ) で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。 <code>width</code> または <code>explicitWidth</code> プロパティを設定すると、このプロパティは NaN にリセットされます。(<i>FlexObject</i> から継承されました)
<i>resizable</i>	ウィンドウをサイズ変更できるかどうかを示す値。
<i>scaleX</i>	水平方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>scaleY</i>	垂直方向の倍率を指定する係数。(<i>FlexObject</i> から継承されました)
<i>showStatusBar</i>	ステータスバーが可視かどうかを示す値。
<i>status</i>	ステータスバーに表示される文字 (表示されている場合)。
<i>systemChrome</i>	ウィンドウが持っているシステム クロム (存在する場合) のタイプ。
<i>tabChildren</i>	オブジェクトの子に対してタブが有効かどうか。デフォルト値は <code>true</code> です。(<i>FlexDisplayObject</i> から継承されました)
<i>tabEnabled</i>	このオブジェクトがタブ順序に含まれるかどうか。(<i>FlexDisplayObject</i> から継承されました)
<i>tabIndex</i>	SWF ファイル内のオブジェクトのタブ順序。(<i>FlexDisplayObject</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>themeColor</i>	コンポーネントのテーマ色。(<i>FlexObject</i> から継承されました)
<i>title</i>	ウィンドウのタイトルバーとタスクバーに表示されるタイトル。
<i>toolTip</i>	ツールチップに表示されるテキスト。デフォルト値は <code>Nothing</code> です。(<i>FlexObject</i> から継承されました)
<i>transparent</i>	ウィンドウを透明にするかどうか。

名前	説明
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)

名前	説明
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。 (SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。 利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。 (FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)

名前	説明
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsActive</i>	ウィンドウがアクティブに設定されているかどうかを返します。(<i>IMoveable</i> から継承されました)
<i>Maximize</i>	ウィンドウを最大化します。(<i>IMoveable</i> から継承されました)
<i>Minimize</i>	ウィンドウを最小化します。(<i>IMoveable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(<i>IMoveable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Restore</i>	ウィンドウを元のサイズに戻します。(<i>IMoveable</i> から継承されました)

名前	説明
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
StateChange	ウィンドウの <code>displaystatus</code> が変更されたときにディスパッチされます。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、 <code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクト

名前	説明
WaitForObject	を表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SparkWindowedApplication クラス

説明

WindowedApplication は、AIR アプリケーション用の Flex アプリケーションを作成するために使用するアプリケーション コンテナを定義します。

継承階層

- [SparkApplication](#)
 - SparkWindowedApplication

構文

```
'Declaration
Public Class SparkWindowedApplication _
Inherits SparkApplication _
Implements IMoveable
```

プロパティ

名前	説明
alpha	指定したオブジェクトのアルファ透過値。(FlexDisplayObject から継承されました)
alwaysInFront	基になる NativeWindow が常に他のウィンドウの手前に表示されるかどうかを示す値。
applicationID	AIR がアプリケーションを識別するために使用する識別子。
autoExit	最後のウィンドウを閉じたときに AIR アプリケーションが終了するか、バックグラウンドで実行を続けるかどうかを示す値。
autoLayout	子の位置やサイズが変更されたときに、大きさやレイアウトの調整が行われるかどうかを示す値。(SparkSkinnableContainer から継承されました)
automationClassName	要素に対するオートメーション クラス名を含んだ文字列。(FlexObject から継承されました)

名前	説明
automationIndex	要素に対するオートメーション インデックスを含んだ文字列。 (FlexObject から継承されました)
automationName	このオブジェクトの識別子として使用することのできる名前。 (FlexObject から継承されました)
className	Button のようなこのインスタンスのクラスの名称。(FlexObject から継承されました)
closed	基になるウィンドウが閉じられているかどうかを示す値。
colorCorrection	ステージの colorCorrection プロパティの値。(SparkApplication から継承されました)
columnNames	データ内のすべての列の名称を含むリスト。 (SparkSkinnableContainerBase から継承されました)
controlBarVisible	コントロールバーが可視かどうか。(SparkApplication から継承されました)
creationPolicy	このコンポーネントの内容の作成ポリシー。 (SparkSkinnableContainer から継承されました)
currentState	コンポーネントの現在の表示状態。(FlexObject から継承されました)
deferredContentCreated	遅延コンテンツが作成されたかどうか。 (SparkSkinnableContainer から継承されました)
enabled	コンポーネントがユーザー操作を受け付けることができるかどうか。 (FlexObject から継承されました)
errorColor	検証が失敗したときにコンポーネントをハイライトする色。 (FlexObject から継承されました)
errorString	コンポーネントが監視され、検証に失敗した場合に、コンポーネントのエラー ヒントとして表示されるテキスト。(FlexObject から継承されました)
firstVisibleRow	最初の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
focusEnabled	コンポーネントが選択時にフォーカスを取得できるかどうか。 (FlexObject から継承されました)
height	表示オブジェクトの高さ (ピクセル単位)。(FlexDisplayObject から継承されました)
id	コンポーネントの ID。(FlexObject から継承されました)
lastVisibleRow	最後の可視の子のインデックス。 (SparkSkinnableContainerBase から継承されました)
maxHeight	親がレイアウトを計算するために使用されるコンポーネントの高さの最大推奨値。
maximizable	ウィンドウを最大化できるかどうかを示す値。
maxWidth	親がレイアウトを計算するために使用されるコンポーネントの幅の最大推奨値。
minHeight	親がレイアウトを計算するために使用されるコンポーネントの高さの最小推奨値。

名前	説明
minimizable	ウィンドウを最小化できるかどうかを示す値。
minWidth	親がレイアウトを計算するために使用されるコンポーネントの幅の最小推奨値。
mouseEnabled	このオブジェクトがマウス メッセージを受け取るかどうか。デフォルト値は true であり、これは表示リスト上の InteractiveObject インスタンスがデフォルトでマウス イベントを受け取ることを意味します。mouseEnabled を false に設定すると、インスタンスでは、マウス イベントを一切受け取りません。表示リスト上のこのインスタンスの子は影響を受けません。表示リスト上のオブジェクトのすべての子に関する mouseEnabled 動作を変更するには、flash.display.DisplayObjectContainer.mouseChildren を使用します。(FlexDisplayObject から継承されました)
numAutomationChildren	このコンテナに含まれるオートメーションの子の数。(FlexObject から継承されました)
numColumns	利用可能なデータ中の列の合計数。(SparkSkinnableContainerBase から継承されました)
numElements	このコンテナに含まれるビジュアル要素の数。(SparkSkinnableContainer から継承されました)
numRows	利用可能なデータ中の行の合計数。(SparkSkinnableContainerBase から継承されました)
percentHeight	コンポーネントの高さを親のサイズに対する比率（パーセンテージ）で指定する数値。(FlexObject から継承されました)
percentWidth	コンポーネントの幅を親のサイズに対する比率（パーセンテージ）で指定する数値。指定できる値は 0 から 100 です。デフォルト値は NaN です。width または explicitWidth プロパティを設定すると、このプロパティは NaN にリセットされます。(FlexObject から継承されました)
resizable	ウィンドウをサイズ変更できるかどうかを示す値。
scaleX	水平方向の倍率を指定する係数。(FlexObject から継承されました)
scaleY	垂直方向の倍率を指定する係数。(FlexObject から継承されました)
showStatusBar	ステータス バーが可視かどうかを示す値。
status	ステータス バーに表示される文字（表示されている場合）。
systemChrome	ウィンドウが持っているシステム クロム（存在する場合）のタイプ。
tabChildren	オブジェクトの子に対してタブが有効かどうか。デフォルト値は true です。(FlexDisplayObject から継承されました)
tabEnabled	このオブジェクトがタブ順序に含まれるかどうか。(FlexDisplayObject から継承されました)
tabIndex	SWF ファイル内のオブジェクトのタブ順序。(FlexDisplayObject から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
themeColor	コンポーネントのテーマ色。(FlexObject から継承されました)
title	ウィンドウのタイトルバーとタスクバーに表示されるタイトル。
toolTip	ツールチップに表示されるテキスト。デフォルト値は Nothing です。(FlexObject から継承されました)
transparent	ウィンドウを透明にするかどうか。
url	このアプリケーションの SWF ファイルがロードされた URL。(SparkApplication から継承されました)
useHandCursor	マウスが要素上に移動したときに、指差しハンドカーソルを表示するかどうか。デフォルト値は true です。useHandCursor が false の場合、代わりに矢印カーソルが表示されます。(FlexObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)
viewSourceURL	アプリケーションのソースを参照できる URL。(SparkApplication から継承されました)
visible	表示オブジェクトが可視かどうか。(FlexDisplayObject から継承されました)
width	表示オブジェクトの幅 (ピクセル単位)。(FlexDisplayObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)
x	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの x 座標。(FlexDisplayObject から継承されました)
y	親 DisplayObjectContainer のローカル座標を基準にした DisplayObject インスタンスの y 座標。(FlexDisplayObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファチャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
ChangeFocus	現在のフォーカスを変更します。(FlexObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DragCancel	ドラッグ操作がキャンセルされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragDrop	ドラッグした項目がドロップされた場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
DragStart	ドラッグ&ドロップ操作が始まった場合にディスパッチされます。(SparkSkinnableContainerBase から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyle	指定したスタイルのスタイル値を返します。利用可能なスタイルについては、Apache Flex SDK ドキュメントを参照してください。(FlexDisplayObject から継承されました)
GetValues	コンポーネントのすべての部分のオートメーション値を含む配列を返します。(SparkSkinnableContainerBase から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	親のコンポーネント内の指定した位置にコンポーネントを移動します。このメソッドを呼び出すことは、コンポーネントの x および y プロパティに値を設定することと同じです。(IMoveable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	このコンポーネントにフォーカスを設定します。コンポーネントはそのサブコンポーネントに順々にフォーカスを渡すことができます。(FlexObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ToString	コントロールの文字列表現を返します。(FlexDisplayObject から継承されました)
Type	ユーザーがキーを押したときにディスパッチされます。(SparkApplication から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Java SWT クラス リファレンス

Java SWT コントロールのテストに利用可能なクラスを一覧します。

CBanner クラス

説明

このクラスは、ワークベンチにおけるツール バー領域とツール バーを切り替えるパースペクティブのレイアウトに使用します。

継承階層

- [Control](#)
 - CBanner

構文

```
'Declaration
Public Class CBanner _
Inherits Control
```


プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Bottom	バナーの下部にあるコントロール。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Left	バナーの左側にあるコントロール。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Right	バナーの右側にあるコントロール。
RightMinimumSize	バナーの右側にあるコントロールの最小サイズ。
RightWidth	バナーの右側にあるコントロールの幅。
Simple	バナーをシンプル、つまり従来の形状でレンダリングするかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

CoolBar クラス

説明

SWT ウィジェットを含むコンテナ クラスで、主要なツールバーです。

継承階層

- [Control](#)
 - CoolBar

構文

```
'Declaration
Public Class CoolBar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	CoolBar の項目の数。
Items	CoolBar の項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

CoolItem クラス

説明

動的に配置することができる CoolBar の領域を表す選択可能なユーザー インターフェイス オブジェクトを含みます。

継承階層

- [Item](#)
 - CoolItem

構文

```
'Declaration
Public Class CoolItem _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Content	動的に配置することができる領域のコンテンツ。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)

名前	説明
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

CTabFolder クラス

説明

拡張タブ フォルダーのクラスです。通常は、装飾されたタブや最小化、最大化、閉じるボタンを持ちます。

継承階層

- [Control](#)
 - CTabFolder

構文

```
'Declaration
Public Class CTabFolder _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	タブ フォルダーの項目の数。
Items	タブ フォルダーの項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択する項目のインデックス。
SelectedItem	選択したタブ項目。
SelectedItemText	選択されている項目のテキスト。項目が選択されていない場合は空の文字列を返します。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	表示されているページを閉じます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
Maximize	CTabFolder をそのフル サイズ (最大化) になるようにサイズを変更します。
Minimize	CTabFolder を可能な最小のサイズになるようにサイズを変更します。
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	CTabFolder を元のサイズに戻します。
Select	CTabFolder から項目を選択します。
SelectList	1 つ以上の項目をタブで選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
	ムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

CTabItem クラス

説明

CTabFolder 内の拡張タブのクラスです。

継承階層

- [Item](#)
 - CTabItem

構文

```
'Declaration
Public Class CTabItem _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Selected	項目が選択されているかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	表示されているタブを閉じます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)

名前	説明
Select	タブを選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイ

名前	説明
	ムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ExpandBar クラス

説明

ヘッダーをクリックすることによって展開したり閉じたりするウィジェットのクラスです。

継承階層

- [Control](#)
 - ExpandBar

構文

```
'Declaration
Public Class ExpandBar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
ExpandedItems	展開される項目のリスト。
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	ExpandBar の項目の数。
Items	ExpandBar の項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Collapse	<code>ExpandBar</code> を折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Expand	<code>ExpandBar</code> を展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ExpandItem クラス

説明

ExpandBar の展開可能な項目を含むクラスです。

継承階層

- [Item](#)
 - ExpandItem

構文

```
'Declaration
Public Class ExpandItem _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Content	項目のコンテンツ。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Expanded	項目が展開されているかどうか。
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Collapse	展開可能な項目を折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Expand	展開可能な項目を展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

HorizontalSash クラス

説明

ウィンドウ ペインの周囲にある水平方向の境界線のクラスです。ドラッグしてウィンドウ サイズを調整することができます。

継承階層

- [Sash](#)
 - HorizontalSash

構文

```
'Declaration  
Public Class HorizontalSash _  
Inherits Sash
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	画面分割の境界のウィンドウ ペインを移動します。 (Sash から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Sash クラス

説明

ウィンドウ ペインの周囲にある境界線のクラスです。ドラッグしてウィンドウ サイズを調整することができます。

継承階層

- [Control](#)
 - Sash
 - [HorizontalSash](#)
 - [VerticalSash](#)

構文

```
'Declaration  
Public Class Sash _  
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Move	画面分割の境界のウィンドウ ペインを移動します。
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SashForm クラス

説明

フォームの周囲にある境界線のクラスです。ドラッグしてフォームのサイズを調整することができます。

継承階層

- [Control](#)
 - SashForm

構文

```
'Declaration
Public Class SashForm _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)

名前	説明
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Horizontal	分割方向が水平かどうか。
MaximizedControl	フォーム上で最大化されている画面分割ウィンドウ。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Vertical	分割方向が垂直かどうか。
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
Weights	SashForm ウィンドウの子の相対的なサイズ。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ScrollableControl クラス

説明

コントロールとして分類されるだけでなく、スクロールバーを含んでいるすべてのコントロールの実装クラスです。

継承階層

- [Control](#)
 - ScrollableControl

構文

```
'Declaration
Public Class ScrollableControl _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)

名前	説明
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForProperty	<p>トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

ScrolledComposite クラス

説明

スクロールバーのクラスです。ユーザーがスクロールバーを使用するときにコンテンツをスクロールします。

継承階層

- [Control](#)
 - ScrolledComposite

構文

```
'Declaration
Public Class ScrolledComposite _
Inherits Control
```

プロパティ

名前	説明
AlwaysShowScrollBars	スクロールバーが表示されるかどうか。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Content	動的に配置することができる領域のコンテンツ。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
Origin	ScrolledComposite の左上隅に現在表示されているコンテンツの位置。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Shell クラス

説明

Java SWT ウィンドウのクラスです。

継承階層

- [Window](#)
 - Shell

構文

```
'Declaration
Public Class Shell _
Inherits Window
```

プロパティ

名前	説明
Application	このウィンドウが属するアプリケーションの名前。(Window から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>Maximize</i>	ウィンドウを最大化します。(<i>IMoveable</i> から継承されました)
<i>Minimize</i>	ウィンドウを最小化します。(<i>IMoveable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	ウィンドウを移動します。(<i>IMoveable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Restore</i>	ウィンドウを元のサイズに戻します。(<i>IMoveable</i> から継承されました)
<i>SetActive</i>	ウィンドウをアクティブにします。(<i>IMoveable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>Size</i>	ウィンドウのサイズを変更します。(<i>IMoveable</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。(<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(<i>IKeyable</i> から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Spinner クラス

説明

アップダウン コントロールの数値を入力したり変更したりするクラスです。

継承階層

- [Control](#)
 - Spinner

構文

```
'Declaration
Public Class Spinner _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
IncrementCount	Spinner コントロールを押下することによって増加する位置の量。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
PageSize	Spinner コントロールの内部ページ サイズ。
Position	アップダウン コントロールの位置。
Range	Spinner コントロールの範囲。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Decrement	Spinner コントロールが押されたときに値を減少させます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Increment	Spinner コントロールが押されたときに値を増加させます。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PageDecrement	Spinner コントロールが押されたときにページ単位で値を減少させます。
PageIncrement	Spinner コントロールが押されたときにページ単位で値を増加させます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SetValue	Spinner コントロールの値を設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

StyledText クラス

説明

テキストの行を表示する編集可能なユーザー インターフェイス オブジェクトのクラスです。

継承階層

- [TextField](#)
 - StyledText

構文

```
'Declaration  
Public Class StyledText _  
Inherits TextField
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Editable	テキストが編集可能かどうか。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
IsPassword	コントロールがパスワード テキスト フィールドであるかどうか。(TextField から継承されました)
MultiLine	コントロールが複数行かどうか。(TextField から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Position	テキスト フィールド内のキャレットの位置。(TextField から継承されました)
SelectedRange	テキスト フィールド内の選択範囲。(TextField から継承されました)
SelectedText	テキスト フィールド内の選択テキスト。テキストが選択されていない場合は空の文字列。(TextField から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ClearText	すべてのテキストをテキスト フィールドから削除します (TextField から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetMultiText	複数行テキスト フィールドの指定された行のテキストを返します。(TextField から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetMultiText</i>	複数行テキスト フィールドの行のすべて、または一部を置換します。(<i>TextField</i> から継承されました)
<i>SetPosition</i>	テキスト フィールド内の挿入位置を設定します。(<i>TextField</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)

名前	説明
SetSelectionRange	単一行または複数行フィールドの指定した範囲を選択します。 (TextField から継承されました)
SetText	テキスト フィールド内のテキストのすべて、または一部を新しいテキストに置換します。(TextField から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTBrowser クラス

説明

ブラウザ ウィジェットのクラスです。

継承階層

- [Control](#)
 - SWTBrowser

構文

```
'Declaration
Public Class SWTBrowser _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)</p>
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTDateTime クラス

説明

カレンダー、日付、または時間コントロールの SWT における日付と時間要素のクラスです。

継承階層

- [Control](#)
 - SWTDateTime

構文

```
'Declaration  
Public Class SWTDateTime _  
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
CurrentDate	現在の日付。
Day	日。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Hours	時。
Minutes	分。
Month	月。
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Seconds	秒。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
Year	年。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保

名前	説明
	<p>存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetHorizontalBar	<p>このコントロールの水平スクロールバーを返します。 (IScrollable から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
GetVerticalBar	<p>このコントロールの垂直スクロールバーを返します。 (IScrollable から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	日付および時間を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTTabControl クラス

説明

SWT TabFolder のクラスです。

継承階層

- [TabControl](#)
 - SWCTabControl

構文

```
'Declaration  
Public Class SWCTabControl _  
Inherits TabControl
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	タブ コントロールの項目の数。(TabControl から継承されました)
Items	タブ コントロールの項目のリスト。(TabControl から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択する項目のインデックス。(TabControl から継承されました)
SelectedItem	選択する項目。(TabControl から継承されました)
SelectedTabItem	選択する項目の名前。
TabItems	タブ コントロールの TabItem のリスト。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル)

名前	説明
	<p>はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetHorizontalBar	<p>このコントロールの水平スクロールバーを返します。 (IScrollable から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
GetVerticalBar	<p>このコントロールの垂直スクロールバーを返します。 (IScrollable から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	現在のページを指定したページに設定します。(TabControl から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTTabItem クラス

説明

SWT TabItem のクラスです。

継承階層

- [Item](#)
 - SWTTabItem

構文

```
'Declaration  
Public Class SWTTabItem _  
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Selected	項目が選択されているかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	タブを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SWTTable クラス

説明

SWT テーブルのクラスです。

継承階層

- [Table](#)
 - SWTTable

構文

```
'Declaration
Public Class SWTTable _
Inherits Table
```

プロパティ

名前	説明
AllowsCheck	コントロールがチェック マークを表示できるかどうか。(Table から継承されました)

名前	説明
AllowsMultiSelect	テーブルが複数項目の選択をサポートするかどうか。(Table から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
ColumnCount	テーブルの列の数。(Table から継承されました)
ColumnItems	列のすべての項目のリスト。(Table から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
RowCount	テーブルの行の数。(Table から継承されました)
RowItems	テーブルのすべての行のリスト。(Table から継承されました)
SelectedIndices	選択した項目の索引。(Table から継承されました)
SelectedItems	選択した項目。(Table から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	テーブルの定義された行のチェック ボックスをチェックします。(Table から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)

名前	説明
DoubleClick	項目をダブルクリックします。(Table から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExtendSelectRow	行の範囲を選択します。(Table から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
FocusRow	テーブルの行にフォーカスを設定します。(Table から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MultiSelectRow	テーブルの行を選択行の集合に追加します。(Table から継承されました)
MultiUnselectRow	行を選択行の集合から削除します。(Table から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SelectRow	テーブルの行を選択します。(Table から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	テーブルの定義された行のチェック ボックスのチェックを外します。(Table から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTTableColumn クラス

説明

SWT テーブルにおける列のクラスです。

継承階層

- [TableColumn](#)
 - SWTTableColumn

構文

```
'Declaration
Public Class SWTTableColumn _
Inherits TableColumn
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
Width	列の幅。(TableColumn から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Resize	列のサイズを変更します。 (TableColumn から継承されました)
Select	列を選択します。 (TableColumn から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTTableRow クラス

説明

SWT テーブルにおける行のクラスです。

継承階層

- [TableRow](#)
 - SWTTableRow

構文

```
'Declaration
Public Class SWTTableRow _
Inherits TableRow
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Checked	行のチェックボックスがチェックされているかどうか。(TableRow から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	行の項目の数。(TableRow から継承されました)
Items	行の項目のリスト。(TableRow から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTTree クラス

説明

SWT ツリー ウィジェットのクラスです。

継承階層

- [Tree](#)
 - SWTTree

構文

```
'Declaration
Public Class SWTTree _
Inherits Tree
```

プロパティ

名前	説明
AllowsCheck	コントロールがチェック マークを表示できるかどうか。(Tree から継承されました)
AllowsMultiSelect	コントロールが複数項目の選択をサポートするかどうか。(Tree から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
ColumnCount	ツリーの列の数。
Columns	ツリーの列のリスト。

名前	説明
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	ツリー上の項目数 (すべての子を含む)。(Tree から継承されました)
ItemPaths	ツリー上の項目のリスト (すべての子を含む)。(Tree から継承されました)
Items	ツリー上の項目のリスト (すべての子を含む)。(Tree から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
SelectedIndex	最初の選択された項目のインデックス。(Tree から継承されました)
SelectedIndices	選択した項目の索引。(Tree から継承されました)
SelectedItem	最初の選択された項目の名前。(Tree から継承されました)
SelectedItems	選択した項目の名前。(Tree から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
VisibleItemPaths	ツリー上の可視の項目のリスト (すべての子を含む)。(Tree から継承されました)
VisibleItems	ツリー上の可視の項目のリスト (すべての子を含む)。(Tree から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	チェック ボックスをチェックします。(Tree から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
Collapse	ツリービュー コントロール上で項目を折りたたみます。(Tree から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClick	項目をダブルクリックします。(Tree から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Expand	ツリー ビュー コントロール上でオブジェクトを展開します。(Tree から継承されました)
ExtendSelect	選択範囲を拡張して項目の範囲を選択します。
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetContentsOfColumn	指定した列のコンテンツをリストとして取得します 最初の列より後ろの列のコンテンツを取得するには、 <code>rawMode</code> と <code>column</code> パラメーターを指定する必要があります。
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)
GetItemPaths	ツリーのすべての項目を項目パスのリストとして返します。(Tree から継承されました)
GetItemRect	TreeView コントロールに対して相対的な項目のサイズと位置を返します。(Tree から継承されました)
GetItemsOfColumn	指定した列の項目をツリーから取得します
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetSelectedTextOfColumn	指定した列から選択したテキストを取得します
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsExpandable	指定した項目が展開可能かどうかを検査します。(Tree から継承されました)
IsExpanded	指定した項目が展開されているかどうかを検査します。(Tree から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MultiSelect	ツリーの項目を選択項目の集合に追加します。
MultiUnselect	ツリーの項目を選択項目の集合から削除します。
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ツリーから項目を選択します。(Tree から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	チェック ボックスのチェックを外します。(Tree から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	タイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SWTTreeColumn クラス

説明

ツリー ウィジェットにおける列のクラスです。

継承階層

- [Item](#)
 - SWTTreeColumn

構文

```
'Declaration
Public Class SWTTreeColumn _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
Width	列の幅。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)

名前	説明
Resize	ツリーの列のサイズを変更します。
Select	ツリーの列を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

VerticalSash クラス

説明

ウィンドウ ペインの周囲にある垂直方向の境界線のクラスです。ドラッグしてウィンドウ サイズを調整することができます。

継承階層

- [Sash](#)
 - VerticalSash

構文

```
'Declaration
Public Class VerticalSash _
Inherits Sash
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	画面分割の境界のウィンドウ ペインを移動します。(Sash から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ViewForm クラス

説明

ワークベンチにおけるビューのラベル/メニュー/ツールバーのローカルバーをレイアウトするクラス。

継承階層

- [Control](#)
 - ViewForm

構文

```
'Declaration
Public Class ViewForm _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Content	動的に配置することができる領域のコンテンツ。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
TopCenter	ペインの中央上にあるウィンドウ。
TopLeft	ペインの左上にあるウィンドウ。
TopRight	ペインの右上にあるウィンドウ。
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

キーワード駆動テスト クラス リファレンス

キーワード駆動テストで利用可能なクラスを一覧します。

ArgumentAttribute クラス

説明

キーワード駆動テストのキーワードとしてマークされるメソッドの引数の名前を明示的に指定するために使用されます。

構文

```
<AttributeUsageAttribute(AttributeTargets.Parameter)>  
Public Class ArgumentAttribute Inherits Attribute
```

プロパティ

名前	説明
Name プロパティ (<i>ArgumentAttribute</i>)	引数の名前を取得します。

例：2つの引数を持つキーワードの作成

引数 username および password を持つ *Login* キーワードを作成するには、次のサンプルコードを使用します。

```
<Keyword(>  
Public Sub Login(<Argument("Name of the user")> username As String,  
<Argument("Password of the user")> password As String)  
    ' keyword implementation  
End Sub
```

例：説明のない引数の使用

引数の説明を使用せずに、引数 username および password を持つ *Login* キーワードを作成するには、次のサンプルコードを使用します。

```
<Keyword(>  
Public Sub Login(username As String, password As String)  
    ' keyword implementation  
End Sub  
  
<Keyword(>  
Public Sub Login(<Argument> username As String, <Argument>  
password As String)  
    ' keyword implementation  
End Sub
```


KeywordAttribute クラス

説明

メソッドがキーワードであることを指定します。キーワードの名前が指定されていない場合、アノテーションメソッドの名前がキーワード名として使用されます。

構文

```
'Declaration
<AttributeUsageAttribute(AttributeTargets.Method)>
Public Class KeywordAttribute Inherits Attribute
```

プロパティ

名前	説明
Name プロパティ (KeywordAttribute)	キーワードの名前を取得します。デフォルト値は空の文字列です。
Description プロパティ (KeywordAttribute)	キーワードの説明を取得します。デフォルト値は空の文字列です。

例：2つの引数を持つキーワードの作成

引数 username および password を持つ *Login* キーワードを作成するには、次のサンプルコードを使用します。

```
<Keyword()>
Public Sub Login(username As String, password As String)
    ' keyword implementation
End Sub
```

例：キーワードの名前の指定

新しいキーワードを作成するときにキーワードの名前を指定するには、次のサンプルコードを使用します。

```
<Keyword("Login user")>
Public Sub Login(username As String, password As String)
    ' keyword implementation
End Sub
```

KeywordGroupAttribute クラス

説明

クラスがキーワードグループであることを指定します。キーワードとしてマークされるこのクラス内のメソッドはキーワードです。これらのメソッドの完全修飾名は、キーワードグループの名前とキーワードの名前で構成されます。キーワードグループの名前がない場合、宣言しているクラスの単純名が代わりに使用されます。

構文

```
'Declaration
<AttributeUsageAttribute(AttributeTargets.Class)>
Public Class KeywordGroupAttribute Inherits Attribute
```

プロパティ

名前	説明
Name プロパティ (KeywordGroupAttribute)	キーワード グループの名前を取得します。デフォルト値は空の文字列です。

例 : キーワード グループの作成

キーワード グループ *LoginPage* を作成するには、次のサンプル コードを使用します。

```
<KeywordGroup()>  
Public Class LoginPage  
    ' keyword methods  
End Class
```

例 : キーワード グループの名前の指定

引数の説明を使用せずに、引数 *username* および *password* を持つ *Login* キーワードを作成するには、次のサンプル コードを使用します。

```
<KeywordGroup("Login Page")>  
Public Class LoginPage  
    ' keyword methods  
End Class
```

Mobile クラス リファレンス

Mobile コントロールのテストに利用可能なクラスを一覧します。

IMobileClickable インターフェイス

説明

クリックを使用するオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class IMobileClickable
```

メソッド

名前	説明
Click	デバイスの指定した座標をクリックします。
DoubleClick	デバイスの指定した座標をダブルクリックします。
LongClick	デバイスの指定した座標をロングクリックします。

IMobileGestures インターフェイス

説明

ロングスワイプを使用するオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class IMobileGestures
```

メソッド

名前	説明
Drag	2つの指定したポイント間でドラッグ操作を実行します。
MultiTouch	指定したポインターを使ったマルチタッチ操作を実行します。すべての指定したポインターに対して、同時にスワイプ操作が実行されません。
PinchIn	各ポインターが他方に向かって、このオブジェクトの端から中央へ移動する2ポインター ジェスチャを実行します。たとえば、イメージを拡大表示する場合に相当します。
PinchOut	各ポインターが他方の反対方向に向かって、このオブジェクトの中央から端へ移動する2ポインター ジェスチャを実行します。たとえば、イメージを縮小表示する場合に相当します。
Swipe	2つの指定したポイント間でスワイプを実行します。
SwipeDown	下スワイプを実行します。
SwipeLeft	左スワイプを実行します。
SwipeRight	右スワイプを実行します。
SwipeUp	上スワイプを実行します。
SwipeWithSegments	指定したポイント間でスワイプを実行します。
TwoPointerMultiTouch	2つのポインターを使ったマルチタッチ操作を実行します。2つのスワイプ操作を同時に実行します。

IMobileKeyable インターフェイス

説明

キーストロークを使用するオブジェクトのインターフェイスです。

構文

```
'Declaration  
Public Class IMobileKeyable
```

メソッド

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。

MobileButton クラス

説明

モバイル ボタンを表します。モバイル ボタンはクリックできます。

継承階層

- [MobileObject](#)
 - MobileButton

構文

```
'Declaration
Public Class MobileButton _
Inherits MobileObject
```

プロパティ

名前	説明
IsEnabled	モバイル オブジェクトが有効かどうか。 (MobileObject から継承されました)
IsFocused	モバイル オブジェクトがフォーカスを持っているかどうか。 このプロパティは iOS ではサポートされません。 (MobileObject から継承されました)
IsSelected	モバイル オブジェクトが選択またはチェックされているかどうか。 (MobileObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	デバイスの指定した座標をクリックします。 (IMobileClickable から継承されました)
DoubleClick	デバイスの指定した座標をダブルクリックします。 (IMobileClickable から継承されました)
Drag	2 つの指定したポイント間でドラッグ操作を実行します。 (IMobileGestures から継承されました)
DragTo	モバイル デバイスに相対的な指定したポイントにこのオブジェクトをドラッグします。 (MobileObject から継承されました)
DragToObject	指定したドラッグ先オブジェクトにこのオブジェクトをドラッグします。 (MobileObject から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
LongClick	デバイスの指定した座標をロングクリックします。 (IMobileClickable から継承されました)
MultiTouch	指定したポインターを使ったマルチタッチ操作を実行します。すべての指定したポインターに対して、同時にスワイプ操作が実行されます。 (IMobileGestures から継承されました)
PinchIn	各ポインターが他方に向かって、このオブジェクトの端から中央へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを拡大表示する場合に相当します。 (IMobileGestures から継承されました)
PinchOut	各ポインターが他方の反対方向に向かって、このオブジェクトの中央から端へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを縮小表示する場合に相当します。 (IMobileGestures から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
Swipe	2 つの指定したポイント間でスワイプを実行します。 (IMobileGestures から継承されました)
SwipeDown	下スワイプを実行します。 (IMobileGestures から継承されました)
SwipeLeft	左スワイプを実行します。 (IMobileGestures から継承されました)
SwipeRight	左スワイプを実行します。 (IMobileGestures から継承されました)
SwipeUp	上スワイプを実行します。 (IMobileGestures から継承されました)
SwipeWithSegments	指定したポイント間でスワイプを実行します。 (IMobileGestures から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TwoPointerMultiTouch	2 つのポインターを使ったマルチタッチ操作を実行します。2 つのスワイプ操作を同時に実行します。 (IMobileGestures から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IMobileKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

MobileDevice クラス

説明

モバイル デバイス。

継承階層

- [TestObject](#)
 - MobileDevice

構文

```
'Declaration
Public Class MobileDevice _
```

Inherits `TestObject`
Implements `IMobileClickable`, `IMobileGestures`, `IMobileKeyable`

プロパティ

名前	説明
DeviceId	現在の接続に対するデバイスの ID。
DisplayHeight	ディスプレイの高さ (ピクセル単位)。
DisplayOrientation	デバイスの現在の向き。
DisplayWidth	ディスプレイの幅 (ピクセル単位)。
IsEmulator	接続したデバイスがエミュレーターかどうか。
Model	接続したデバイスのデバイス名
OperatingSystem	モバイル デバイスのオペレーティング システム (Android または iOS)
OsVersion	接続したデバイスのオペレーティング システムのバージョン。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	デバイスの指定した座標をクリックします。 (IMobileClickable から継承されました)
CloseApp	アプリを閉じてデバイスを開放し、ほかのユーザーが使用できるようにします。
DoubleClick	デバイスの指定した座標をダブルクリックします。 (IMobileClickable から継承されました)
Drag	2 つの指定したポイント間でドラッグ操作を実行します。 (IMobileGestures から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateConnectionString	このモバイル デバイス用の接続文字列を生成します。
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
LongClick	デバイスの指定した座標をロングクリックします。 (IMobileClickable から継承されました)
MultiTouch	指定したポインターを使ったマルチタッチ操作を実行します。すべての指定したポインターに対して、同時にスワイプ操作が実行されます。 (IMobileGestures から継承されました)
PinchIn	各ポインターが他方に向かって、このオブジェクトの端から中央へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを拡大表示する場合に相当します。 (IMobileGestures から継承されました)
PinchOut	各ポインターが他方の反対方向に向かって、このオブジェクトの中央から端へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを縮小表示する場合に相当します。 (IMobileGestures から継承されました)
PressBack	戻るボタンを押します。このメソッドは iOS ではサポートされません。
PressDelete	削除ボタンを押します。
PressEnter	Enter ボタンを押します。
PressHome	ホーム ボタンを押します。
PressKeyCode	特定の Android 固有のキー コードを押します。このメソッドは iOS ではサポートされません。
PressRecentApps	最近使ったアプリケーション ボタンを押します。
Rotate	デバイス画面の回転をシミュレートします。
SetLocation	デバイスの地理的なロケーションを設定します。このメソッドは、Android デバイス (物理デバイスまたはエミュレーター) および iOS シミュレーター上でのみ機能します。
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
Sleep	スクリーンがオンの場合に電源ボタンを押します。このメソッドは iOS ではサポートされません。
Swipe	2 つの指定したポイント間でスワイプを実行します。 (IMobileGestures から継承されました)
SwipeDown	下スワイプを実行します。 (IMobileGestures から継承されました)
SwipeLeft	左スワイプを実行します。 (IMobileGestures から継承されました)
SwipeRight	右スワイプを実行します。 (IMobileGestures から継承されました)
SwipeUp	上スワイプを実行します。 (IMobileGestures から継承されました)
SwipeWithSegments	指定したポイント間でスワイプを実行します。 (IMobileGestures から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TwoPointerMultiTouch	2つのポインターを使ったマルチタッチ操作を実行します。2つのスワイプ操作を同時に実行します。(IMobileGestures から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IMobileKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは5秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WakeUp	スクリーンがオフの場合に電源ボタンを押します。このメソッドは iOS ではサポートされません。

MobileObject クラス

説明

すべてのモバイル オブジェクトの基本クラスです。

継承階層

- [TestObject](#)
 - MobileObject
 - [MobileButton](#)
 - [MobileTextField](#)
 - [MobileWindow](#)

構文

```
'Declaration
Public Class MobileObject _
Inherits TestObject _
Implements IMobileClickable, IMobileGestures, IMobileKeyable
```

プロパティ

名前	説明
IsEnabled	モバイル オブジェクトが有効かどうか。
IsFocused	モバイル オブジェクトがフォーカスを持っているかどうか。このプロパティは iOS ではサポートされません。
IsSelected	モバイル オブジェクトが選択またはチェックされているかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
Click	デバイスの指定した座標をクリックします。(IMobileClickable から継承されました)
DoubleClick	デバイスの指定した座標をダブルクリックします。(IMobileClickable から継承されました)
Drag	2つの指定したポイント間でドラッグ操作を実行します。(IMobileGestures から継承されました)
DragTo	モバイルデバイスに相対的な指定したポイントにこのオブジェクトをドラッグします。
DragToObject	指定したドラッグ先オブジェクトにこのオブジェクトをドラッグします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む)のリストを返します。(TestObject から継承されました)
GetParent	テストアプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
LongClick	デバイスの指定した座標をロングクリックします。(IMobileClickable から継承されました)
MultiTouch	指定したポインターを使ったマルチタッチ操作を実行します。すべての指定したポインターに対して、同時にスワイプ操作が実行されます。(IMobileGestures から継承されました)
PinchIn	各ポインターが他方に向かって、このオブジェクトの端から中央へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを拡大表示する場合に相当します。(IMobileGestures から継承されました)
PinchOut	各ポインターが他方の反対方向に向かって、このオブジェクトの中央から端へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを縮小表示する場合に相当します。(IMobileGestures から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Swipe	2 つの指定したポイント間でスワイプを実行します。(IMobileGestures から継承されました)
SwipeDown	下スワイプを実行します。(IMobileGestures から継承されました)
SwipeLeft	左スワイプを実行します。(IMobileGestures から継承されました)
SwipeRight	左スワイプを実行します。(IMobileGestures から継承されました)
SwipeUp	上スワイプを実行します。(IMobileGestures から継承されました)
SwipeWithSegments	指定したポイント間でスワイプを実行します。(IMobileGestures から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TwoPointerMultiTouch	2つのポインターを使ったマルチタッチ操作を実行します。2つのスワイプ操作を同時に実行します。(IMobileGestures から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IMobileKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは5秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

MobileTextField クラス

説明

テキスト フィールドを表します。

継承階層

- [MobileObject](#)
 - MobileTextField

構文

```
'Declaration  
Public Class MobileTextField _  
Inherits MobileObject
```

プロパティ

名前	説明
IsEnabled	モバイル オブジェクトが有効かどうか。 (MobileObject から継承されました)
IsFocused	モバイル オブジェクトがフォーカスを持っているかどうか。 このプロパティは iOS ではサポートされません。 (MobileObject から継承されました)
IsSelected	モバイル オブジェクトが選択またはチェックされているかどうか。 (MobileObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ClearText	テキスト フィールドのすべてのコンテンツをクリアします。
Click	デバイスの指定した座標をクリックします。 (IMobileClickable から継承されました)
DoubleClick	デバイスの指定した座標をダブルクリックします。 (IMobileClickable から継承されました)

名前	説明
Drag	2つの指定したポイント間でドラッグ操作を実行します。 (IMobileGestures から継承されました)
DragTo	モバイルデバイスに相対的な指定したポイントにこのオブジェクトをドラッグします。(MobileObject から継承されました)
DragToObject	指定したドラッグ先オブジェクトにこのオブジェクトをドラッグします。(MobileObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テストアプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
LongClick	デバイスの指定した座標をロングクリックします。(IMobileClickable から継承されました)
MultiTouch	指定したポインターを使ったマルチタッチ操作を実行します。すべての指定したポインターに対して、同時にスワイプ操作が実行されます。(IMobileGestures から継承されました)
PinchIn	各ポインターが他方に向かって、このオブジェクトの端から中央へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを拡大表示する場合に相当します。(IMobileGestures から継承されました)
PinchOut	各ポインターが他方の反対方向に向かって、このオブジェクトの中央から端へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを縮小表示する場合に相当します。(IMobileGestures から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetText	テキスト フィールド内のテキストを新しいテキストに置換します。
Swipe	2 つの指定したポイント間でスワイプを実行します。(IMobileGestures から継承されました)
SwipeDown	下スワイプを実行します。(IMobileGestures から継承されました)
SwipeLeft	左スワイプを実行します。(IMobileGestures から継承されました)
SwipeRight	左スワイプを実行します。(IMobileGestures から継承されました)
SwipeUp	上スワイプを実行します。(IMobileGestures から継承されました)
SwipeWithSegments	指定したポイント間でスワイプを実行します。(IMobileGestures から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TwoPointerMultiTouch	2つのポインターを使ったマルチタッチ操作を実行します。2つのスワイプ操作を同時に実行します。(IMobileGestures から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IMobileKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドに入力します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

MobileWindow クラス

説明

モバイル ウィンドウ。

継承階層

- [MobileObject](#)
 - MobileWindow

構文

```
'Declaration
Public Class MobileWindow _
Inherits MobileObject
```

プロパティ

名前	説明
IsEnabled	モバイル オブジェクトが有効かどうか。 (MobileObject から継承されました)
IsFocused	モバイル オブジェクトがフォーカスを持っているかどうか。 このプロパティは iOS ではサポートされません。 (MobileObject から継承されました)
IsSelected	モバイル オブジェクトが選択またはチェックされているかどうか。 (MobileObject から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	デバイスの指定した座標をクリックします。 (IMobileClickable から継承されました)
DoubleClick	デバイスの指定した座標をダブルクリックします。 (IMobileClickable から継承されました)

名前	説明
Drag	2つの指定したポイント間でドラッグ操作を実行します。 (IMobileGestures から継承されました)
DragTo	モバイルデバイスに相対的な指定したポイントにこのオブジェクトをドラッグします。(MobileObject から継承されました)
DragToObject	指定したドラッグ先オブジェクトにこのオブジェクトをドラッグします。(MobileObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む)のリストを返します。(TestObject から継承されました)
GetParent	テストアプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
LongClick	デバイスの指定した座標をロングクリックします。(IMobileClickable から継承されました)
MultiTouch	指定したポインターを使ったマルチタッチ操作を実行します。すべての指定したポインターに対して、同時にスワイプ操作が実行されます。(IMobileGestures から継承されました)
PinchIn	各ポインターが他方に向かって、このオブジェクトの端から中央へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを拡大表示する場合に相当します。(IMobileGestures から継承されました)
PinchOut	各ポインターが他方の反対方向に向かって、このオブジェクトの中央から端へ移動する 2 ポインター ジェスチャを実行します。たとえば、イメージを縮小表示する場合に相当します。(IMobileGestures から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Swipe	2 つの指定したポイント間でスワイプを実行します。(IMobileGestures から継承されました)
SwipeDown	下スワイプを実行します。(IMobileGestures から継承されました)
SwipeLeft	左スワイプを実行します。(IMobileGestures から継承されました)
SwipeRight	右スワイプを実行します。(IMobileGestures から継承されました)
SwipeUp	上スワイプを実行します。(IMobileGestures から継承されました)
SwipeWithSegments	指定したポイント間でスワイプを実行します。(IMobileGestures から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TwoPointerMultiTouch	2つのポインターを使ったマルチタッチ操作を実行します。2つのスワイプ操作を同時に実行します。(IMobileGestures から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IMobileKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Rumba クラス リファレンス

Rumba コントロールのテストに利用可能なクラスを一覧します。

Rumba キー

説明

RumbaObject クラスの SendKeys や SynchSendKeys メソッドで使用される Rumba キーのニーモニックに対応する文字列定数です。

以下の文字列定数が含まれます。

文字列定数	キー ニーモニック
At	@@
Alt	@A
AltCursor	@e@1*
Attention	@A@Q
Backspace	@B
Backtab	@B
Clear	@C
Cmd	@A@Y
CursorDown	@V
CursorLeft	@L
CursorRight	@Z
CursorSelect	@e@1*
CursorUp	@U
Delete	@D
Dup	@S@x
End	@q
Enter	@E
EraseEof	@F
EraseEol	@e@d
EraseInput	@A@F
FieldExit	@A@E
FieldMark	@S@y
FieldMinus	@A@-
FieldPlus	@A@+
Help	@H
Hexadecimal	@A@X
Home	@0
Insert	@I
InsertToggle	@A@I
LocalHelp	@e@I

文字列定数

LocalPrint
LocalHelpCursor
LeftTab
NewLine
PageUp
PageDown
Print
PrintScreen
RecordBackspace
Reset
RightTab
Shift
SysRequest
Tab
Test
Usm
AplCutOnOf
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PF1
PF2
PF3
PF4
PF5
PF6
PF7
PF8
PF9

キー ニーモニック

@P
@e@m*
@B
@e@n*
@u
@v
@A@t
@A@T
@A@<
@R
@T
@S
@A@H
@T
@A@C
@e@v
@e@2
@x
@y
@z
@+
@%
@&
@'
@(
@)
@*
@1
@2
@3
@4
@5
@6
@7
@8
@9

文字列定数	キー ニーモニック
PF10	@a
PF11	@b
PF12	@c
PF13	@d
PF14	@e
PF15	@f
PF16	@g
PF17	@h
PF18	@i
PF19	@j
PF20	@k
PF21	@l
PF22	@m
PF23	@n
PF24	@o
Cents	@n@1
VerticalBar	@d@2
Corner	@d@3
Plus	@d@4
Minus	@d@5
Quote	@d@7
HorizontalBar	@d@8
OpenBracket	@d@9
CloseBracket	@d@10
Substitute	@d@0
Synchronous	@d@a

RumbaCharacterAttribute データ型

スクリーン上の文字の文字属性を表します。

次の文字属性を含めることができます。

属性	説明
backgroundColor	文字の背景色。
foregroundColor	文字の前景色。
column	スクリーン上の文字の列。
line	スクリーン上の文字の行。
autoEnter	文字がオートエンター テキストの一部かどうかを定義します。

属性	説明
autoSkip	文字がオートスキップ テキストの一部かどうかを定義します。
blinking	文字が点滅表示かどうかを定義します。
bold	文字が太字かどうかを定義します。
columnSeparator	文字が列セパレータかどうかを定義します。
detectable	文字が検出可能テキストの一部かどうかを定義します。
hidden	文字が非表示テキストの一部かどうかを定義します。
protect	文字が保護テキストの一部かどうかを定義します。
reverseVideo	文字が反転表示かどうかを定義します。
underlined	文字が下線付きかどうかを定義します。

詳細については、Rumba Developer Edition (RDE) ドキュメントの『*IRDEAttribute* プロパティ』を参照してください。

RumbaField クラス

説明

グリーン スクリーン上のフィールドを表します。

継承階層

- [RumbaObject](#)
 - RumbaField
 - [RumbaLabel](#)
 - [RumbaTextField](#)

構文

```
'Declaration
Public Class RumbaField _
Inherits RumbaObject
```

プロパティ

名前	説明
AutoEnter	フィールドが AutoEnter フィールドの場合に true を示す値。
AutoSkip	フィールドが AutoSkip フィールドの場合に true を示す値。
Detectable	フィールドが検出可能な場合に true を示す値。
Dup	フィールドが Dup フィールドの場合に true を示す値。
EndColumn	フィールドが終了する列を示す値。
EndLine	フィールドが終了する行を示す値。
EndPosition	スクリーンが 1 つの連続した文字として見られる場合、1 から始まるフィールドの終了位置です。
Enptui	フィールドが ENPTUI フィールドの場合に true を示す値。

名前	説明
FieldExitRequired	フィールドがスクリーンを終了する前に必要とされる場合に true を示す値。
Hidden	フィールドが非表示かどうかを示す値。
Intense	フィールドが強調されている場合に true を示す値。
Length	フィールドの長さ。
ModifiedDataTag	フィールドが ModifiedDataTag プロパティ セットを持つ場合に true を示す値。
Protected	フィールドが保護されている場合に true を示す値 (RumbaLabel など)。
StartColumn	フィールドが始まる列を示す値。
StartLine	フィールドが始まる行を示す値。
StartPosition	スクリーンが 1 つの連続した文字として見られる場合、1 から始まるフィールドの開始位置です。
Text	コントロールのテキスト。(TestObject から継承されました)
Type	このフィールドに対して許されている入力を表す値。
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetCharacterAttributes</i>	このフィールドの文字の文字属性を取得します。
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SendKeys	キーを送信します。RDE ニーモニックをサポートします。(RumbaObject から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SynchSendKeys	キーを同期して送信します。RDE ニーモニックをサポートします。(RumbaObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
	<p>アウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

RumbaLabel クラス

説明

テキストを取得できるが設定できないスクリーン上のフィールド。

継承階層

- [RumbaField](#)
 - RumbaLabel

構文

```
'Declaration
Public Class RumbaLabel _
Inherits RumbaField
```

プロパティ

名前	説明
AutoEnter	<p>フィールドが <code>AutoEnter</code> フィールドの場合に true を示す値。(RumbaField から継承されました)</p>
AutoSkip	<p>フィールドが <code>AutoSkip</code> フィールドの場合に true を示す値。(RumbaField から継承されました)</p>

名前	説明
<i>Detectable</i>	フィールドが検出可能な場合に true を示す値。(RumbaField から継承されました)
<i>Dup</i>	フィールドが Dup フィールドの場合に true を示す値。(RumbaField から継承されました)
<i>EndColumn</i>	フィールドが終了する列を示す値。(RumbaField から継承されました)
<i>EndLine</i>	フィールドが終了する行を示す値。(RumbaField から継承されました)
<i>EndPosition</i>	スクリーンが 1 つの連続した文字として見られる場合、1 から始まるフィールドの終了位置です。(RumbaField から継承されました)
<i>Enptui</i>	フィールドが ENPTUI フィールドの場合に true を示す値。(RumbaField から継承されました)
<i>FieldExitRequired</i>	フィールドがスクリーンを終了する前に必要とされる場合に true を示す値。(RumbaField から継承されました)
<i>Hidden</i>	フィールドが非表示かどうかを示す値。(RumbaField から継承されました)
<i>Intense</i>	フィールドが強調されている場合に true を示す値。(RumbaField から継承されました)
<i>Length</i>	フィールドの長さ。(RumbaField から継承されました)
<i>ModifiedDataTag</i>	フィールドが ModifiedDataTag プロパティ セットを持つ場合に true を示す値。(RumbaField から継承されました)
<i>Protected</i>	フィールドが保護されている場合に true を示す値 (RumbaLabel など)。(RumbaField から継承されました)
<i>StartColumn</i>	フィールドが始まる列を示す値。(RumbaField から継承されました)
<i>StartLine</i>	フィールドが始まる行を示す値。(RumbaField から継承されました)
<i>StartPosition</i>	スクリーンが 1 つの連続した文字として見られる場合、1 から始まるフィールドの開始位置です。(RumbaField から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Type</i>	このフィールドに対して許されている入力を表す値。(RumbaField から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保

名前	説明
	<p>存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetCharacterAttributes	<p>このフィールドの文字の文字属性を取得します。 (RumbaField から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SendKeys</i>	キーを送信します。RDE ニーモニックをサポートします。(<i>RumbaObject</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>SynchSendKeys</i>	キーを同期して送信します。RDE ニーモニックをサポートします。(<i>RumbaObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

RumbaObject クラス

説明

RumbaObject。

継承階層

- [TestObject](#)
 - RumbaObject
 - [RumbaField](#)
 - [RumbaScreen](#)

構文

```
'Declaration
Public Class RumbaObject _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable
```

プロパティ

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SendKeys	キーを送信します。RDE ニーモニックをサポートします。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SynchSendKeys	キーを同期して送信します。RDE ニーモニックをサポートします。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
	値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

RumbaScreen クラス

説明

グリーン スクリーン全体を表します。

継承階層

- [RumbaObject](#)
 - RumbaScreen

構文

```
'Declaration
Public Class RumbaScreen _
Inherits RumbaObject
```

プロパティ

名前	説明
BackgroundColor	ディスプレイ コントロールの背景色。
CharacterSetId	ディスプレイ コントロールが使用する文字セット変換テーブルを識別する値。
ColumnCount	現在のスクリーンの列数を示す値。
CursorColumn	スクリーン上のカーソルの列の位置。
CursorLine	スクリーン上のカーソルの行の位置。
CursorPosition	スクリーン上の現在のカーソル位置。 X と Y の値は 1 から始まる値です (最初の行または列を表す)。
HostConnected	ホストが接続されているかどうかを示す値。
HostDeviceName	ホスト デバイス名。

名前	説明
HostIpAddress	ホスト IP アドレス。
KeyboardLocked	キーボード入力が抑制されているかどうかを示す値。
LineCount	現在のスクリーンの行数を示す値。
ScreenId	スクリーン ID。
ScreenSize	テキストの行数と列数で表したスクリーンのサイズ。
SessionName	セッション名。
SupportsStructuredFields	ホストが構造化フィールドのデコードをサポートするかどうかを示す値。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Clear	「クリア (CLEAR)」キーを送信してスクリーンをクリアします。 <code>synchSendKeys(RumbaKey.CLEAR)</code> と等価です。
Click	オブジェクトをクリックします。(IClickable から継承されました)
Connect	現在の構成内のホストに接続します。
Disconnect	現在の接続を解除します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetCharacterAttributes</i>	指定した位置から始まる、指定した長さの文字に対する文字属性を取得します。
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetContent</i>	指定したテキスト位置から始まる改行を含まないスクリーンを含んだ文字列を取得します。
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetLines</i>	スクリーンの指定範囲を取得し、選択した行を文字列のリストとして返します。
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Reset	「リセット (RESET)」キーを送信してスクリーンをリセットします。 synchSendKeys(RumbaKey.RESET) と等価です。
SendKeys	キーを送信します。RDE ニーモニックをサポートします。 (RumbaObject から継承されました)
SetCursorPosition	スクリーンの指定した位置にカーソルを移動します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
Submit	「送信 (SUBMIT)」キーを送信して入力したデータをサーバーに送信します。 synchSendKeys(RumbaKey.SUBMIT) と等価です。
SynchSendKeys	キーを同期して送信します。RDE ニーモニックをサポートします。 (RumbaObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForHostConnected	セッションが接続されるまで待機します。
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WriteText	スクリーン上にテキストを書き出します。

RumbaTextField クラス

説明

テキストを取得および設定できるスクリーン上のフィールド。

継承階層

- [RumbaField](#)
 - RumbaTextField

構文

```
'Declaration
Public Class RumbaTextField _
Inherits RumbaField
```

プロパティ

名前	説明
<i>AutoEnter</i>	フィールドが AutoEnter フィールドの場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>AutoSkip</i>	フィールドが AutoSkip フィールドの場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>Detectable</i>	フィールドが検出可能な場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>Dup</i>	フィールドが Dup フィールドの場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>EndColumn</i>	フィールドが終了する列を示す値。 (<i>RumbaField</i> から継承されました)
<i>EndLine</i>	フィールドが終了する行を示す値。 (<i>RumbaField</i> から継承されました)
<i>EndPosition</i>	スクリーンが 1 つの連続した文字として見られる場合、1 から始まるフィールドの終了位置です。 (<i>RumbaField</i> から継承されました)
<i>Enptui</i>	フィールドが ENPTUI フィールドの場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>FieldExitRequired</i>	フィールドがスクリーンを終了する前に必要とされる場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>Hidden</i>	フィールドが非表示かどうかを示す値。 (<i>RumbaField</i> から継承されました)
<i>Intense</i>	フィールドが強調されている場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>Length</i>	フィールドの長さ。 (<i>RumbaField</i> から継承されました)
<i>ModifiedDataTag</i>	フィールドが ModifiedDataTag プロパティ セットを持つ場合に true を示す値。 (<i>RumbaField</i> から継承されました)
<i>Protected</i>	フィールドが保護されている場合に true を示す値 (RumbaLabel など)。 (<i>RumbaField</i> から継承されました)
<i>StartColumn</i>	フィールドが始まる列を示す値。 (<i>RumbaField</i> から継承されました)
<i>StartLine</i>	フィールドが始まる行を示す値。 (<i>RumbaField</i> から継承されました)
<i>StartPosition</i>	スクリーンが 1 つの連続した文字として見られる場合、1 から始まるフィールドの開始位置です。 (<i>RumbaField</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Type</i>	このフィールドに対して許されている入力を表す値。 (<i>RumbaField</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetCharacterAttributes	このフィールドの文字の文字属性を取得します。 (RumbaField から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SendKeys	キーを送信します。RDE ニーモニックをサポートします。(RumbaObject から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
 SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetText	テキスト フィールドのテキストを設定します。
SynchSendKeys	キーを同期して送信します。RDE ニーモニックをサポートします。(RumbaObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SAP クラス リファレンス

SAP コントロールのテストに利用可能なクラスを一覧します。

ISapContextMenuable インターフェイス

説明

コンテキスト メニューの選択をサポートする任意のコンポーネント用のインターフェイスです。

構文

```
'Declaration  
Public Class ISapContextMenuable
```

メソッド

名前	説明
SelectContextMenuItem	コントロールのコンテキスト メニューから項目を選択します。
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキスト メニュー項目を選択できます。
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキスト メニューのメニュー項目を選択します。

SapBarChart クラス

説明

SAP の BarChart クラスです。

継承階層

- [SapShell](#)
 - SapBarChart

構文

```
'Declaration  
Public Class SapBarChart _  
Inherits SapShell _  
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityDescription	シェルのアクセシビリティの説明。
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)

名前	説明
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
ChartCount	グラフの数。
CurrentContextMenu	現在のコンテキスト メニュー。 コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ&ドロップ操作を許可しているかどうか。 (SapShell から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)
WindowHandle	GuiShell に接続しているコントロールのウィンドウ ハンドル。

メソッド

名前	説明
BarCount	指定したグラフの棒の数を返します。

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetBarContent	棒のコンテンツを返します。
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetGridLineContent	グリッド線のコンテンツを返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GridCount	グラフ内のグリッドの数を返します。
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
LinkCount	指定したグラフ内のリンクの数を返します。
SelectContextMenuItem	コントロールのコンテキスト メニューから項目を選択します。(ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキスト メニュー項目を選択できます。(ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキスト メニューのメニュー項目を選択します。(ISapContextMenuable から継承されました)
SendData	サーバーにデータを送信します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapBox クラス

説明

SAP の SapBox クラスです。

継承階層

- [SapComponent](#)
 - SapBox

構文

```
'Declaration
Public Class SapBox _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>
ImageClickFile	<p>指定したイメージをクリックします。 (TestObject から継承されました)</p>
ImageExists	<p>指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)</p>

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
	<p>値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapButton クラス

説明

SapButton クラスはプッシュ ボタンのクラスです。

継承階層

- [SapComponent](#)
 - SapButton

構文

```
'Declaration
Public Class SapButton _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CharHeight	要素の高さ (文字単位)。
CharLeft	要素の左からの座標 (文字単位)。
CharTop	要素の上からの座標 (文字単位)。

名前	説明
CharWidth	要素の幅（文字単位）。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。（ SapComponent から継承されました）
Height	現在のコンポーネントの高さ。（ SapComponent から継承されました）
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。（ SapComponent から継承されました）
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。（ SapComponent から継承されました）
Left	画面上の座標で要素の左からの位置。（ SapComponent から継承されました）
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。（ SapComponent から継承されました）
Name	コンポーネントの名前。（ SapComponent から継承されました）
ScreenLeft	画面上の座標でコンポーネントの左からの位置。（ SapComponent から継承されました）
ScreenTop	画面上の座標でコンポーネントの上からの位置。（ SapComponent から継承されました）
Text	コントロールのテキスト。（ TestObject から継承されました）
Tooltip	ツールチップ テキスト。（ SapComponent から継承されました）
Top	画面上の座標で要素の上からの位置。（ SapComponent から継承されました）
Value	コントロールの値（たとえば、テキスト コントロールのテキスト）。（ TestObject から継承されました）
Width	現在のコンポーネントの幅。（ SapComponent から継承されました）

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます（%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps）。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目（アルファ チャンネル）はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。（ TestObject から継承されました）
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
Select	このメソッドはボタンの手動による押下をエミュレートします。ボタンの押下は、常にサーバーとの通信を発生させ、そのウィンドウレベル以下の要素へのすべての無効な参照がレンダリングされます。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapCalendar クラス

説明

SapCalendar クラスは SapCalendar を表します。

継承階層

- [SapShell](#)
 - SapCalendar

構文

```
'Declaration
Public Class SapCalendar _
Inherits SapShell _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。 コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
Day	現在選択されている月の日。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ & ドロップ操作を許可しているかどうか。(SapShell から継承されました)
EndSelection	選択範囲の終了点。

名前	説明
<i>FirstVisibleDate</i>	カレンダー コントロールの最も早い可視な日付。
<i>Height</i>	現在のコンポーネントの高さ。 (<i>SapComponent</i> から継承されました)
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (<i>SapComponent</i> から継承されました)
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (<i>SapComponent</i> から継承されました)
<i>LastVisibleDate</i>	カレンダー コントロールの最後の可視な日付。
<i>Left</i>	画面上の座標で要素の左からの位置。 (<i>SapComponent</i> から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (<i>SapComponent</i> から継承されました)
<i>Month</i>	現在選択されている年の月。
<i>Name</i>	コンポーネントの名前。 (<i>SapComponent</i> から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。 (<i>SapComponent</i> から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。 (<i>SapComponent</i> から継承されました)
<i>SelectionInterval</i>	現在選択されている範囲。
<i>StartSelection</i>	選択範囲の開始点。
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Week</i>	現在選択されている年の週。
<i>Width</i>	現在のコンポーネントの幅。 (<i>SapComponent</i> から継承されました)
<i>Year</i>	現在選択されている年。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ContextMenu	コンテキストメニューを開きます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む)のリストを返します。(TestObject から継承されました)
GetParent	テストアプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
SelectContextMenuItem	コントロールのコンテキストメニューから項目を選択します。 (ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。 (ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。 (ISapContextMenuable から継承されました)
SetFirstVisibleDate	カレンダー コントロールの最も早い可視な日付を設定します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetFocusDate	フォーカスを持つ日付を設定します。
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SetSelectionInterval	指定した範囲を選択します。
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapChart クラス

説明

SAP の Chart クラスです。

継承階層

- [SapShell](#)
 - SapChart

構文

```
'Declaration
Public Class SapChart _
Inherits SapShell _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityDescription	シェルのアクセシビリティの説明。
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)

名前	説明
<i>AccessibilityTextOnRequest</i>	アクセシビリティ サポートに関する追加のテキスト。 (<i>SapComponent</i> から継承されました)
<i>AccessibilityTooltip</i>	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Changeable</i>	オブジェクトが変更可能かどうか。(<i>SapComponent</i> から継承されました)
<i>CurrentContextMenu</i>	現在のコンテキスト メニュー。コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
<i>DefaultTooltip</i>	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(<i>SapComponent</i> から継承されました)
<i>DragDropSupported</i>	シェルがドラッグ&ドロップ操作を許可しているかどうか。 (<i>SapShell</i> から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(<i>SapComponent</i> から継承されました)
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (<i>SapComponent</i> から継承されました)
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(<i>SapComponent</i> から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。(<i>SapComponent</i> から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(<i>SapComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。(<i>SapComponent</i> から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(<i>SapComponent</i> から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(<i>SapComponent</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(<i>SapComponent</i> から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(<i>SapComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>SapComponent</i> から継承されました)
<i>WindowHandle</i>	GuiShell に接続しているコントロールのウィンドウ ハンドル。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SelectContextMenuItem	コントロールのコンテキスト メニューから項目を選択します。(ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキスト メニュー項目を選択できます。(ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキスト メニューのメニュー項目を選択します。(ISapContextMenuable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ValueChange	グラフの値を変更します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapCheckBox クラス

説明

SapCheckBox クラスは、チェック ボックス コントロールのクラスです。

継承階層

- [SapComponent](#)
 - SapCheckBox

構文

```
'Declaration
Public Class SapCheckBox _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	<p>アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)</p>

名前	説明
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CharHeight	要素の高さ (文字単位)。
CharLeft	要素の左からの座標 (文字単位)。
CharTop	要素の上からの座標 (文字単位)。
CharWidth	要素の幅 (文字単位)。
Checked	チェックボックスがチェックされているかどうか。
ColorIndex	この要素のリスト色のインデックス。
ColorIntensified	この要素の画面描画に対して INTENSIFIED フラグが設定されているかどうか。
ColorInverse	この要素の画面描画に対して INVERSE 色スタイルが設定されているかどうか。
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Flushing	値の変更がラウンドトリップするかどうか。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsLeftLabel	コンポーネントが 'assign left' フラグを持つかどうか。
IsListElement	要素が画面でなく ABAP リスト上にあるかどうか。
IsRightLabel	コンポーネントが 'assign right' フラグを持つかどうか。
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
RowText	現在のコンポーネントを含んでいる行全体のテキスト。
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	チェック ボックスをチェックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Uncheck	チェック ボックスのチェックを外します。

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapColorSelector クラス

説明

SapColorSelector クラスは SapColorSelector を表します。

継承階層

- [SapShell](#)
 - SapColorSelector

構文

```
'Declaration
Public Class SapColorSelector _
Inherits SapShell _
Implements ISapContextMenuable
```


プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。 コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ&ドロップ操作を許可しているかどうか。 (SapShell から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ChangeSelection	指定したインデックスの位置の色を選択します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)

名前	説明
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SelectContextMenuItem	コントロールのコンテキスト メニューから項目を選択します。(ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキスト メニュー項目を選択できます。(ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキスト メニューのメニュー項目を選択します。(ISapContextMenuable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定し

名前	説明
WaitForDisappearance	<p>ない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapComboBox クラス

説明

SAP の ComboBox クラスです。

継承階層

- [SapComponent](#)
 - SapComboBox

構文

```
'Declaration
Public Class SapComboBox _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CharHeight	要素の高さ (文字単位)。
CharLeft	要素の左からの座標 (文字単位)。
CharTop	要素の上からの座標 (文字単位)。
CharWidth	要素の幅 (文字単位)。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Flushing	値の変更がラウンド トリップするかどうか。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsLeftLabel	コンポーネントが 'assign left' フラグを持つかどうか。
IsListBoxActive	コンボ ボックスのリスト ボックスが現在開かれているかどうか。
IsRightLabel	コンポーネントが 'assign right' フラグを持つかどうか。
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Key	現在選択されている項目のキー。
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
Required	コンポーネントが画面に必要な値かどうか。
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCurrentEntryKey	現在のエントリのキーを取得します。
GetCurrentEntryValue	現在のエントリの値を取得します。
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetEntryKeys	すべてのエントリ キーのリストを取得します。
GetEntryValues	すべてのエントリ値のリストを取得します。

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
SelectItem	指定した項目を選択します。
SelectKey	指定したキーを持つリスト項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapComponent クラス

説明

SapComponent クラスは SAP の基本クラスです。

継承階層

- [TestObject](#)
 - SapComponent

- [SapBox](#)
- [SapButton](#)
- [SapCheckBox](#)
- [SapComboBox](#)
- [SapContainer](#)
- [SapContainerShell](#)
- [SapCustomControl](#)
- [SapDockShell](#)
- [SapLabel](#)
- [SapMenu](#)
- [SapMenubar](#)
- [SapOkCodeField](#)
- [SapRadioButton](#)
- [SapScrollContainer](#)
- [SapShell](#)
- [SapSimpleContainer](#)
- [SapSplitterContainer](#)
- [SapStatusbar](#)
- [SapTab](#)
- [SapTable](#)
- [SapTabStrip](#)
- [SapTextField](#)
- [SapTitlebar](#)
- [SapToolbar](#)
- [SapUserArea](#)
- [SapWindow](#)

構文

```
'Declaration
Public Class SapComponent _
Inherits TestObject _
Implements IFocusable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。
Changeable	オブジェクトが変更可能かどうか。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。
Height	現在のコンポーネントの高さ。
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。

名前	説明
<i>Left</i>	画面上の座標で要素の左からの位置。
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。
<i>Name</i>	コンポーネントの名前。
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。
<i>Top</i>	画面上の座標で要素の上からの位置。
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapContainer クラス

説明

SAP の Container クラスです。

継承階層

- [SapComponent](#)
 - SapContainer

構文

```
'Declaration  
Public Class SapContainer _  
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
WaitForObject	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapContainerShell クラス

説明

SAP の ContainerShell クラスです。

継承階層

- [SapComponent](#)
 - SapContainerShell

構文

```
'Declaration
Public Class SapContainerShell _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)

名前	説明
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 (TestObject から継承されました)

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapContextMenu クラス

説明

SAP コンテキスト メニューのクラスです。

SAP コンテキスト メニューの項目を決定する方法については、『[Determining the item names and item texts of a SapContextMenu](#)』を参照してください。

継承階層

- [SapMenu](#)
 - SapContextMenu

構文

```
'Declaration
Public Class SapContextMenu _
Inherits SapMenu
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)

名前	説明
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Select	メニューを選択します。(SapMenu から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapCustomControl クラス

説明

SAP の CustomControl クラスです。

継承階層

- [SapComponent](#)
 - SapCustomControl

構文

```
'Declaration  
Public Class SapCustomControl _  
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)

名前	説明
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>SapComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapDockShell クラス

説明

SAP の DockShell クラスです。

継承階層

- [SapComponent](#)
 - SapDockShell

構文

```
'Declaration
Public Class SapDockShell _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)

名前	説明
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapGridView クラス

説明

SAP の GridView クラス。 Get で始まる名前のメソッドのいくつかは、タイプ文字列の入力パラメーターとして column を必要とします。 このパラメーターは個々の GridView 要素によって異なります。 このパラメーターの値を取得するには、必要な項目の操作を記録し、記録したパラメーターを使用します。

継承階層

- [SapShell](#)
 - SapGridView

構文

```
'Declaration
Public Class SapGridView _
Inherits SapShell _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
ColumnCount	列の数。
ColumnOrder	すべての列識別子とその表示順で含まれる文字列のリスト。
CurrentCellColumn	現在の列。
CurrentCellRow	現在の行。

名前	説明
CurrentContextMenu	現在のコンテキストメニュー。コンテキストメニューがシェルオブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップテキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ&ドロップ操作を許可しているかどうか。(SapShell から継承されました)
FirstVisibleColumn	グリッドビューのスクロール可能領域の最初の可視な列。固定された列は無視されます。
FirstVisibleRow	グリッドの最初の可視な行のインデックス。
FrozenColumnCount	水平方向のスクロールから除外される列の数。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
RowCount	行の数。
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
SelectedCells	選択したセルを含む文字列のリスト。それぞれの書式は "<行>,<列 ID>" となります。
SelectedColumns	現在選択されている列の文字列のリスト。
SelectedRows	行インデックス番号、またはインデックス範囲のカンマ区切りリスト ("1,4,6-8" など)。
SelectionMode	選択モード。有効な値は、"RowsAndColumns"、"ListboxSingle"、"ListboxMultiple"、"Free" です。
Text	コントロールのテキスト。(TestObject から継承されました)
Title	グリッド コントロールのタイトル。
ToolbarButtonCount	区切りを含むツールバーのボタンの数。
Tooltip	ツールチップテキスト。(SapComponent から継承されました)

名前	説明
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VisibleRowCount	グリッド ビューの可視な行の数。
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ClearSelection	すべての行、列、セルの選択を削除します。
Click	指定したセルをクリックします。
ClickCurrentCell	現在のセルをクリックします。
ContextMenu	コンテキスト メニューを要求します。
CurrentCellMoved	この関数は、他のセルが現在のセルになったことをサーバーに通知します。現在のセルが変更された場合に関数が呼び出されます。
DeleteRows	指定した行を削除します。
DeselectColumn	選択した列のリストから指定した列を削除します。
DoubleClick	指定したセルをダブルクリックします。
DoubleClickCurrentCell	現在のセルをダブルクリックします。
DuplicateRows	指定した範囲の行を複製します
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetCellColor	セルの色の識別子を返します。
GetCellHeight	セルの高さ (ピクセル単位) を取得します。
GetCellIcon	指定したセルのアイコン文字列を取得します。
GetCellLeft	セルの左からの位置を取得します。
GetCellMaxLength	セルの最大の長さをバイト数で取得します。
GetCellState	指定したセルの状態を取得します。有効な値は、Normal、Error、Warning、Info です。
GetCellTooltip	指定したセルのツールチップを取得します。
GetCellTop	指定したセルの上からの位置を取得します。
GetCellType	指定したセルのセルタイプを取得します。
GetCellValue	指定したセルの値を取得します。
GetCellWidth	指定したセルの幅を取得します。
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetColorInfo	セルの色の説明を返します。
GetColumnDataType	列のデータ型を取得します。
GetColumnPosition	指定した列の位置を取得します。
GetColumnSortType	列のソート種別を取得します。有効な値は、None、Ascending、Descending です。
GetColumnTitles	指定した列のすべての列タイトルを取得します。
GetColumnTooltip	指定した列のツールチップ テキストを取得します。
GetColumnTotalType	列の合計種別を取得します。有効な値は、None、Total、Subtotal です。
GetDisplayedColumnTitle	指定した列の現在表示されているタイトルを取得します。
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetRowTotalLevel	行のレベルを取得します。
GetSymbolInfo	セルの記号の説明を返します。
GetToolBarButtonIcon	指定したツールバー ボタンのアイコン名を取得します。
GetToolBarButtonId	指定したツールバー ボタンの ID を取得します。
GetToolBarButtonText	指定したツールバー ボタンのテキストを取得します。
GetToolBarButtonTooltip	指定したツールバー ボタンのツールチップを取得します
GetToolBarButtonType	指定したツールバーのボタンの種類を取得します。有効な値は、"Button"、"ButtonAndMenu"、"Menu"、"Separator"、"Group"、"CheckBox" です。
GetToolBarFocusButton	現在フォーカスを持つツールバー ボタンの位置を取得します
HasCellF4Help	指定したセルがヘルプが割り当てられた値を持つ場合に true を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
InsertRows	行の範囲を挿入します
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsCellChangeable	指定したセルが変更可能な場合に true を返します。
IsCellCheckBoxChecked	指定した位置にあるチェック ボックスがチェックされている場合に true を返します。
IsCellHotspot	セルがリンクである場合に true を返します。
IsCellSymbol	セルのテキストが SAP シンボル フォントで表示されている場合に true を返します。

名前	説明
<i>IsCellTotalExpander</i>	指定したセルに合計エクスペンダ ボタンがある場合に true を返します。
<i>IsColumnFiltered</i>	指定した列がフィルタリングされている場合に true を返します。
<i>IsColumnKey</i>	指定した列がキー列である場合に true を返します。
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>IsToolBarButtonChecked</i>	指定したツール バー ボタンがチェックされている場合に true を返します。
<i>IsToolBarButtonEnabled</i>	指定したボタンが有効な場合に true を返します。
<i>IsTotalRowExpanded</i>	指定した合計行が展開されているかどうかを取得します。
<i>ModifyCell</i>	指定したセルの値を変更します。
<i>ModifyCheckBox</i>	指定したチェック ボックス セルの値を変更します。
<i>MoveRows</i>	fromRow 以上、toRow 以下のインデックスを持つ行を destRow の位置に移動させます。
<i>PressButton</i>	指定したセルにあるボタンを押します。
<i>PressButtonCurrentCell</i>	現在のセルにあるボタンをクリックします。
<i>PressColumnHeader</i>	指定した列のヘッダーをクリックします。
<i>PressEnter</i>	Enter キーを押します。
<i>PressF1</i>	F1 キーを押します。
<i>PressF4</i>	F4 キーを押します。
<i>PressToolBarButton</i>	グリッド ビューのツール バーの指定したボタンをクリックします。
<i>PressToolBarContextButton</i>	グリッド ビューのツール バーの指定したボタンのコンテキスト メニューを開きます。
<i>PressTotalRow</i>	指定した合計行の行ボタンを押します。
<i>PressTotalRowCurrentCell</i>	現在のセルにある展開ボタンを押します。
<i>SelectAll</i>	グリッドのコンテンツ全体 (たとえば、すべての行と列) を選択します。
<i>SelectColumn</i>	選択した列のリストに指定した列を追加します。
<i>SelectContextMenuItem</i>	コントロールのコンテキスト メニューから項目を選択します。 (<i>ISapContextMenuable</i> から継承されました)
<i>SelectContextMenuItemByPosition</i>	このメソッドは項目の位置を利用してコンテキスト メニュー項目を選択できます。(<i>ISapContextMenuable</i> から継承されました)
<i>SelectContextMenuItemByText</i>	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキスト メニューのメニュー項目を選択します。 (<i>ISapContextMenuable</i> から継承されました)
<i>SelectionChanged</i>	選択範囲が変更されたことをサーバーに通知します
<i>SelectToolBarMenuItem</i>	グリッド ビューのツール バーのコンテキスト メニューから項目を選択します。

名前	説明
SetColumnOrder	列の順序を設定します。
SetColumnWidth	指定した列の幅を設定します。
SetCurrentCell	指定したセルを現在のセルとして設定します。
SetCurrentCellColumn	現在のセルの列を指定します。
SetCurrentCellRow	現在のセルの行を設定します。
SetFirstVisibleColumn	最初の可視な列を設定します。
SetFirstVisibleRow	最初の可視な行を設定します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SetSelectedRows	指定した行を選択します。
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TriggerModified	セルの複数の変更をサーバーに通知します。一般に、ModifyCell への複数の呼び出しの跡に、このメソッドは呼び出されます。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
	<p>値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapHorizontalScrollBar クラス

説明

水平スクロールバーのクラスです。

継承階層

- [SapScrollbar](#)
 - SapHorizontalScrollBar

構文

```
'Declaration
Public Class SapHorizontalScrollBar _
Inherits SapScrollbar
```

プロパティ

名前	説明
Maximum	スクロールバーの最大値。(SapScrollbar から継承されました)
Minimum	スクロールバーの最小値。(SapScrollbar から継承されました)
PageSize	スクロールバーのページサイズ。(SapScrollbar から継承されました)
Position	スクロールバーの現在の位置。(SapScrollbar から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
ScrollTo	指定した位置にスクロールします。(SapScrollbar から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapHTMLViewer クラス

説明

SAP の HTMLView クラスです。

継承階層

- [SapShell](#)
 - SapHTMLViewer

構文

```
'Declaration
Public Class SapHTMLViewer _
Inherits SapShell _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DocumentComplete	ドキュメントが完全かどうか。

名前	説明
DragDropSupported	シェルがドラッグ&ドロップ操作を許可しているかどうか。 (SapShell から継承されました)
Height	現在のコンポーネントの高さ。 (SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。 (SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (SapComponent から継承されました)
Name	コンポーネントの名前。 (SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。 (SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。 (SapComponent から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Tooltip	ツールチップ テキスト。 (SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。 (SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (SapComponent から継承されました)
WindowHandle	GuiShell に接続しているコントロールのウィンドウ ハンドル。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ContextMenu	コンテキスト メニューの要求をエミュレートします。バックエンドによって提供されるコンテキスト メニューに対してのみ適用されます。

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
SapEvent	この関数は HTML フォームをバックエンドに送信します。
SelectContextMenuItem	コントロールのコンテキストメニューから項目を選択します。 (ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。 (ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。 (ISapContextMenuable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapLabel クラス

説明

SapLabel クラスはすべての SAP ラベルのクラスです。

継承階層

- [SapComponent](#)
 - SapLabel

構文

```
'Declaration
Public Class SapLabel _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CharLeft	ラベルの左からの座標 (文字単位)。

名前	説明
<i>CharTop</i>	ラベルの上からの座標 (文字単位)。
<i>CharWidth</i>	要素の幅 (文字単位)。
<i>ColorIndex</i>	ラベルの色インデックス。
<i>ColorInverse</i>	ラベルに対して INVERSE 色スタイルが設定されているかどうか。
<i>DefaultTooltip</i>	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>DisplayedText</i>	ラベルのテキスト。
<i>Height</i>	現在のコンポーネントの高さ。 (<i>SapComponent</i> から継承されました)
<i>Highlighted</i>	ラベルに対して 'highlighted' フラグが割り当てられているかどうか。
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (<i>SapComponent</i> から継承されました)
<i>IsHotspot</i>	ラベルに対するマウス クリックがラウンド トリップするかどうか。
<i>IsLeftLabel</i>	コンポーネントに 'assign left' フラグが割り当てられているかどうか。
<i>IsListElement</i>	要素が ABAP リスト上にあるかどうか。
<i>IsRightLabel</i>	コンポーネントに 'assign right' フラグが割り当てられているかどうか。
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (<i>SapComponent</i> から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。 (<i>SapComponent</i> から継承されました)
<i>Maxlength</i>	ラベルのテキストの最大の長さ (コード単位)。
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (<i>SapComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。 (<i>SapComponent</i> から継承されました)
<i>Numerical</i>	ラベルに数値だけを含めることができるかどうか。
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。 (<i>SapComponent</i> から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)

名前	説明
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>ShowContextMenu</i>	コントロールのコンテキストメニューを表示します。(<i>SapComponent</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション

名前	説明
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapMenu クラス

説明

SAP の Menu クラスです。

継承階層

- [SapComponent](#)
 - SapMenu
 - [SapContextMenu](#)

構文

```
'Declaration
Public Class SapMenu _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)

名前	説明
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
Select	メニューを選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に

名前	説明
WaitForObject	対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapMenubar クラス

説明

SAP の MenuBar クラスです。

継承階層

- [SapComponent](#)
 - SapMenubar

構文

```
'Declaration
Public Class SapMenubar _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)

名前	説明
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)

名前	説明
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapNetPlan クラス

説明

SAP の NetChart クラスです。

継承階層

- [SapShell](#)
 - SapNetPlan

構文

```
'Declaration  
Public Class SapNetPlan _  
Inherits SapShell _  
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityDescription	シェルのアクセシビリティの説明。
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。 コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ & ドロップ操作を許可しているかどうか。 (SapShell から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
LinkCount	ネットのリンクの数。

名前	説明
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>NodeCount</i>	ネットのノードの数。
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)
<i>WindowHandle</i>	GuiShell に接続しているコントロールのウィンドウ ハンドル。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetLinkContent	リンクのコンテンツを返します。
GetNodeContent	ノードのコンテンツを返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SelectContextMenuItem	コントロールのコンテキスト メニューから項目を選択します。(ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキスト メニュー項目を選択できます。(ISapContextMenuable から継承されました)

名前	説明
SelectContextMenuByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。 (ISapContextMenuable から継承されました)
SendData	この関数は、コントロール側でトリガーされたアクションそれぞれの出力をエミュレートします。アクションの結果はサーバーに送信されます。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapOfficeIntegration クラス

説明

SAP の OfficeIntegration クラスです。

継承階層

- [SapShell](#)
 - SapOfficeIntegration

構文

```
'Declaration
Public Class SapOfficeIntegration _
Inherits SapShell _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。 コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ & ドロップ操作を許可しているかどうか。(SapShell から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)

名前	説明
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (<i>SapComponent</i> から継承されました)
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (<i>SapComponent</i> から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。 (<i>SapComponent</i> から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (<i>SapComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。 (<i>SapComponent</i> から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。 (<i>SapComponent</i> から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。 (<i>SapComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
SelectContextMenuItem	コントロールのコンテキスト メニューから項目を選択します。 (ISapContextMenuable から継承されました)

名前	説明
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。(ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。(ISapContextMenuable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapOkCodeField クラス

説明

SapOkCodeField クラスは SAP OkCodeField のクラスです。

継承階層

- [SapComponent](#)
 - SapOkCodeField

構文

```
'Declaration
Public Class SapOkCodeField _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)

名前	説明
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>Opened</i>	GuiOkCodeField が折りたたまれているかどうか。
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Close</i>	GuiOkCodeField を折りたたみます。
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Open	GuiOkCodeField を展開します。
PressF1	GuiOkCodeField にフォーカスがある間に F1 キーを押すことをエミュレートします。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 SetText 	コントロールのテキストを指定したテキストに設定します。
 ShowContextMenu 	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapPicture クラス

説明

SAP 画像のクラスです。

継承階層

- [SapShell](#)
 - SapPicture

構文

```
'Declaration  
Public Class SapPicture _  
Inherits SapShell _  
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DisplayMode	画像の表示モード ("Normal"、"Stretch"、"Fit"、"NormalCenter"、"FitCenter" など)。
DragDropSupported	シェルがドラッグ & ドロップ操作を許可しているかどうか。 (SapShell から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
Icon	SAPGUI アイコン コード ("@01@" など)。
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)

名前	説明
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Url	画像の URL
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	画像をクリックします
ClickControlArea	指定した位置をクリックします
ClickPictureArea	画像領域の指定した位置をクリックします
ContextMenu	この関数は指定した位置にコンテキスト メニューを開きます。座標は、画面上に表示される画像コントロールに対してピクセル単位で指定します。
DoubleClick	画像をダブルクリックします
DoubleClickControlArea	指定した位置をダブルクリックします
DoubleClickPictureArea	画像領域の指定した位置をダブルクリックします

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>SelectContextMenuItem</i>	コントロールのコンテキストメニューから項目を選択します。 (<i>ISapContextMenuable</i> から継承されました)
<i>SelectContextMenuItemByPosition</i>	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。 (<i>ISapContextMenuable</i> から継承されました)
<i>SelectContextMenuItemByText</i>	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。 (<i>ISapContextMenuable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>ShowContextMenu</i>	コントロールのコンテキストメニューを表示します。 (<i>SapComponent</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForObject</i>	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	タイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapRadioButton クラス

説明

SapRadioButton クラスはラジオ ボタン コントロールのクラスです。

継承階層

- [SapComponent](#)
 - SapRadioButton

構文

```
'Declaration
Public Class SapRadioButton _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CharHeight	要素の高さ (文字単位)
CharLeft	要素の左からの座標 (文字単位)

名前	説明
<i>CharTop</i>	要素の上からの座標 (文字単位)
<i>CharWidth</i>	要素の幅 (文字単位)
<i>DefaultTooltip</i>	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Flushing</i>	値の変更がラウンド トリップするかどうか
<i>Height</i>	現在のコンポーネントの高さ。 (<i>SapComponent</i> から継承されました)
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (<i>SapComponent</i> から継承されました)
<i>IsLeftLabel</i>	コンポーネントが 'assign left' フラグを持つかどうか
<i>IsRightLabel</i>	コンポーネントが 'assign right' フラグを持つかどうか
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (<i>SapComponent</i> から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。 (<i>SapComponent</i> から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (<i>SapComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。 (<i>SapComponent</i> から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。 (<i>SapComponent</i> から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Selected</i>	ラジオ ボタンが選択されているかどうか
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。 (<i>SapComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保

名前	説明
	<p>存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>
ImageClickFile	<p>指定したイメージをクリックします。 (TestObject から継承されました)</p>
ImageExists	<p>指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)</p>
ImageExistsFile	<p>指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)</p>

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
Select	ラジオ ボタンを選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前に

名前	説明
WaitForProperty	<p>トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapScrollbar クラス

説明

SAP の Scrollbar クラスです。

継承階層

- [TestObject](#)
 - SapScrollbar
 - [SapHorizontalScrollBar](#)
 - [SapVerticalScrollBar](#)

構文

```
'Declaration
Public Class SapScrollbar _
Inherits TestObject
```

プロパティ

名前	説明
Maximum	スクロールバーの最大値。
Minimum	スクロールバーの最小値。
PageSize	スクロールバーのページ サイズ。
Position	スクロールバーの現在の位置。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
ScrollTo	指定した位置にスクロールします。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に

名前	説明
WaitForObject	対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapScrollContainer クラス

説明

SAP の UserArea クラスです。

継承階層

- [SapComponent](#)
 - SapScrollContainer

構文

```
'Declaration
Public Class SapScrollContainer _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)

名前	説明
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)

名前	説明
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapShell クラス

説明

SAP の SapShell クラスです。

継承階層

- [SapComponent](#)
 - SapShell
 - [SapBarChart](#)
 - [SapCalendar](#)
 - [SapChart](#)
 - [SapColorSelector](#)
 - [SapGridView](#)
 - [SapHTMLViewer](#)
 - [SapNetPlan](#)
 - [SapOfficeIntegration](#)
 - [SapPicture](#)
 - [SapTextEdit](#)
 - [SapToolbarControl](#)
 - [SapTree](#)

構文

```
'Declaration  
Public Class SapShell _  
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ & ドロップ操作を許可しているかどうか。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)

名前	説明
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (<i>SapComponent</i> から継承されました)
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (<i>SapComponent</i> から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。 (<i>SapComponent</i> から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (<i>SapComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。 (<i>SapComponent</i> から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。 (<i>SapComponent</i> から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。 (<i>SapComponent</i> から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。 (<i>SapComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。 (<i>SapComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 ShowContextMenu 	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapSimpleContainer クラス

説明

SAP の SimpleContainer クラスです。

継承階層

- [SapComponent](#)
 - SapSimpleContainer

構文

```
'Declaration  
Public Class SapSimpleContainer _  
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapSplitterContainer クラス

説明

SAP スプリッタ コンテナのクラスです。

継承階層

- [SapComponent](#)
 - SapSplitterContainer

構文

```
'Declaration
Public Class SapSplitterContainer _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
WaitForObject	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapStatusbar クラス

説明

SAP のステータスバー クラスです。

継承階層

- [SapComponent](#)
 - SapStatusbar

構文

```
'Declaration
Public Class SapStatusbar _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)

名前	説明
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
MessageId	ABAP メッセージ呼び出しで使用されたメッセージ クラスの名前
MessageNumber	ABAP メッセージ呼び出しで使用されたメッセージ番号
MessageType	ステータス バーに表示されたメッセージの種類 (たとえば、S (Success)、W (Warning)、E (Error)、A (Abort)、I (Information))
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
DoubleClick	ステータスバーをダブルクリックします
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>ShowContextMenu</i>	コントロールのコンテキストメニューを表示します。 (<i>SapComponent</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(<i>TestObject</i> から継承されました)
<i>WaitForObject</i>	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapTab クラス

説明

SAP の Tab クラスです。

継承階層

- [SapComponent](#)
 - SapTab

構文

```
'Declaration
Public Class SapTab _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)

名前	説明
<i>Left</i>	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Select	タブを選択します
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapTable クラス

説明

SapTable クラスは SAP テーブルのクラスです。

継承階層

- [SapComponent](#)
 - SapTable

構文

```
'Declaration  
Public Class SapTable _  
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。 (SapComponent から継承されました)
CharHeight	要素の高さ (文字単位)。
CharLeft	要素の左からの座標 (文字単位)。
CharTop	要素の上からの座標 (文字単位)。
CharWidth	要素の幅 (文字単位)。
ColumnCount	テーブルの列の数。
ColumnSelectionMode	列の選択モード。
CurrentColumn	現在の列のインデックス。
CurrentRow	現在の行のインデックス。
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。 (SapComponent から継承されました)
Height	現在のコンポーネントの高さ。 (SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)

名前	説明
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>RowCount</i>	テーブルの行の数 (不可視の行を含む)。
<i>RowSelectionMode</i>	行の選択モード。
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>TableFieldName</i>	テーブルのフィールド名。
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>VisibleRowCount</i>	テーブルの可視な行の数。
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>ConfigureLayout</i>	テーブルのレイアウトを変更するために、レイアウトの設定ダイアログを開きます。
<i>DeselectAllColumns</i>	この関数は、1 ステップですべての列の選択を解除できるボタンを持つテーブル コントロールに対して使用することができます。
<i>DeselectColumn</i>	指定した列の選択を解除します。

名前	説明
DeselectRow	指定した行の選択を解除します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetCell	可視なテーブルセルを返します。
GetCellAbsolute	テーブルセルを返します。行が表示されていない場合、スクロールして表示します。
GetCellsForRow	指定した行のセルを返します。
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetColumnTitle	このメソッドは指定した列のタイトルを返します。
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む)のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVisibleCellsForColumn	指定した列の可視なセルを返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsColumnSelected</i>	指定した列が選択されている場合に true を返します。
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>IsRowSelected</i>	指定した行が選択されている場合に true を返します。
<i>ReorderTable</i>	このメソッドは列の順序を並べ替えます。
<i>SelectAllColumns</i>	この関数は、1 ステップですべての列を選択できるボタンを持つテーブル コントロールに対して使用することができます。
<i>SelectColumn</i>	指定した列を選択します。
<i>SelectRow</i>	指定した行を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>ShowContextMenu</i>	コントロールのコンテキスト メニューを表示します。(<i>SapComponent</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SapTabStrip クラス

説明

SAP のタブ ペイン クラスです。

継承階層

- [SapComponent](#)
 - SapTabStrip

構文

```
'Declaration
Public Class SapTabStrip _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	<p>アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)</p>

名前	説明
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。 (SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。 (SapComponent から継承されました)
Height	現在のコンポーネントの高さ。 (SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。 (SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。 (SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。 (SapComponent から継承されました)
Name	コンポーネントの名前。 (SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。 (SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。 (SapComponent から継承されました)
SelectedTab	現在選択されているタブ。
Tabs	タブ ペインのすべてのタブ。
Text	コントロールのテキスト。 (TestObject から継承されました)
Tooltip	ツールチップ テキスト。 (SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。 (SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>
ImageClickFile	<p>指定したイメージをクリックします。 (TestObject から継承されました)</p>
ImageExists	<p>指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)</p>

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForObject	値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapTextEdit クラス

説明

SAP の TextEdit クラスです。

継承階層

- [SapShell](#)
 - SapTextEdit

構文

```
'Declaration
Public Class SapTextEdit _
Inherits SapShell
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentColumn	現在キャレットがある列のインデックス。
CurrentContextMenu	現在のコンテキスト メニュー。コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
CurrentLine	現在キャレットがある行のインデックス。

名前	説明
<i>DefaultTooltip</i>	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
<i>DragDropSupported</i>	シェルがドラッグ&ドロップ操作を許可しているかどうか。(SapShell から継承されました)
<i>FirstVisibleLine</i>	テキスト編集コントロールの上端に表示されている最初の可視な行。
<i>Height</i>	現在のコンポーネントの高さ。(SapComponent から継承されました)
<i>IconName</i>	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
<i>LastVisibleLine</i>	テキスト編集コントロールの下端に表示されている最後の可視な行。
<i>Left</i>	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
<i>LineCount</i>	テキスト編集コントロールのすべての行の数。
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>SelectedText</i>	現在選択されているテキスト。
<i>SelectionEndColumn</i>	選択範囲の終了点の列インデックス。
<i>SelectionEndLine</i>	選択範囲の終了点の行インデックス。
<i>SelectionIndexEnd</i>	テキストの選択範囲全体の終了インデックス。
<i>SelectionIndexStart</i>	テキストの選択範囲全体の開始インデックス。
<i>SelectionStartColumn</i>	テキストの選択範囲の列の開始インデックス。
<i>SelectionStartLine</i>	選択範囲の開始点の行インデックス。
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ContextMenu	コンテキスト メニューを要求します。
DoubleClick	テキスト編集コントロールをダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetLineText	指定した行のテキストを返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsBreakpointLine	指定した行にブレークポイントが含まれているかどうかを取得します。
IsCommentLine	指定した行がコメント行である場合に true を返します。
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
IsHighlightedLine	指定した行がハイライトされている場合に true を返します。
IsProtectedLine	指定した行が保護されている場合に true を返します。
PressF1	F1 キーを押します。
PressF4	F4 キーを押します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionIndexes	指定した範囲のテキストを選択します。
SetText	テキスト編集コントロールに指定したテキストを設定します。
ShowContextMenu	コントロールのコンテキスト メニューを表示します。(SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapTextField クラス

説明

SapTextfield はユーザーが変更可能なテキストを保持する、単一行および複数行フィールドのクラスです。

継承階層

- [SapComponent](#)
 - SapTextField

構文

```
'Declaration  
Public Class SapTextField _  
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DisplayedText	画面上に表示されるテキスト (前後の空白を含む)。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsPassword	テキスト フィールドがパスワード テキスト フィールドであるかどうか。
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

名前	説明
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetPasswordText</i>	暗号化されたパスワードをオブジェクトに送信します。
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>SetText</i>	コントロールのテキストを指定したテキストに設定します。
<i>ShowContextMenu</i>	コントロールのコンテキストメニューを表示します。(<i>SapComponent</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapTitlebar クラス

説明

SAP の Titlebar クラスです。

継承階層

- [SapComponent](#)
 - SapTitlebar

構文

```
'Declaration
Public Class SapTitlebar _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)

名前	説明
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケータで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケータで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケータを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
ShowContextMenu	コントロールのコンテキスト メニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapToolbar クラス

説明

SAP の Toolbar クラスです。

継承階層

- [SapComponent](#)
 - SapToolbar

構文

```
'Declaration
Public Class SapToolbar _
Inherits SapComponent
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプ of データ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)

名前	説明
<i>IsSymbolFont</i>	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
<i>Left</i>	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)

名前	説明
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapToolBarControl クラス

説明

SAP の ToolbarControl クラスです。

継承階層

- [SapShell](#)
 - SapToolBarControl

構文

```
'Declaration  
Public Class SapToolBarControl _  
Inherits SapShell _  
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。 (SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。 (SapComponent から継承されました)
ButtonCount	区切りを含むツール バーのボタンの数。
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキストメニュー。コンテキストメニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ&ドロップ操作を許可しているかどうか。 (SapShell から継承されました)
FocusedButton	現在フォーカスを持つボタンの 0 を基準としたインデックス。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。 (SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)

名前	説明
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)
<i>WindowHandle</i>	GuiShell に接続しているコントロールのウィンドウ ハンドル。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetButtonIcon	指定したツールバー ボタンのアイコン名を取得します。
GetButtonType	指定したツールバーのボタンの種類を取得します。有効な値は、"Button"、"ButtonAndMenu"、"Menu"、"Separator"、"Group"、"CheckBox" です
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsButtonChecked	ボタンが現在チェックされている (押されている) かどうかを取得します。
IsButtonEnabled	ボタンを押すことができるかどうかを取得します。
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
PressContextButton	指定した ID を持つコンテキスト ボタンを押します。

名前	説明
SelectButton	指定した ID を持つボタンの押下をエミュレートします。
SelectContextMenuItem	コントロールのコンテキストメニューから項目を選択します。 (ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。 (ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。 (ISapContextMenuable から継承されました)
SelectMenuItem	指定した ID を持つメニュー項目の選択をエミュレートします。
SelectMenuItemByText	メニュー項目のテキストによるメニュー項目の選択をエミュレートします。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForObject	トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapTree クラス

説明

SapTree クラスはすべての SAP ツリーのクラスです。Get で始まる名前メソッドのいくつかは、タイプ文字列の入力パラメーターとして nodeKey と ItemName を必要とします。これらのパラメーターは個々の SapTree 要素によって異なります。これらのパラメーターの値を取得するには、必要な項目の操作を記録し、記録したパラメーターを使用します。

継承階層

- [SapShell](#)
 - SapTree

構文

```
'Declaration
Public Class SapTree _
Inherits SapShell _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityDescription	シェルアクセシビリティの説明。
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)

名前	説明
ColumnHeaders	列のタイトルのコレクション。
ColumnNames	列名のコレクション。
ColumnOrder	列順序。
ColumnTitles	列のタイトルのコレクション。
CurrentContextMenu	現在のコンテキストメニュー。コンテキストメニューがシェルオブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップテキスト。(SapComponent から継承されました)
DragDropSupported	シェルがドラッグ&ドロップ操作を許可しているかどうか。(SapShell から継承されました)
FocusedNodeKey	フォーカスを持つノードのキー。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
HierarchyHeaderWidth	階層ヘッダーの幅 (ピクセル単位)。
HierarchyTitle	階層タイトル。
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)
Modified	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
Name	コンポーネントの名前。(SapComponent から継承されました)
NodeKeys	アプリケーションに依存したルート ノードのノード キーまたはツリーのすべてのノードのオード キーのどちらかを含んでいるコレクション。
ScreenLeft	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
ScreenTop	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
SelectedItemColumn	現在選択されている項目列のキー。
SelectedItemNode	現在選択されている項目ノードのキー。
SelectedNode	現在選択されているノードのキー。
SelectedNodes	ツリー内のすべての選択されたノードのノード キーを含むコレクション。
SelectionMode	選択モード。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Tooltip	ツールチップ テキスト。(SapComponent から継承されました)
Top	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
TopNode	トップ ノードのキー。
TreeType	ツリーの種類 (単純ツリー、リスト ツリー、列ツリー)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ChangeCheckbox	このメソッドはチェックボックスの状態の変更をエミュレートします。
CollapseNode	この関数は nodeKey で指定したキーでノードを閉じます。
DefaultContextMenu	このメソッドは、ツリー コントロール全体に対してコンテキスト メニューを要求します。
DoubleSelectItem	この関数はテキスト項目のダブルクリックをエミュレートします。
DoubleSelectNode	この関数はノードのダブルクリックをエミュレートします。
EnsureVisibleHorizontalItem	この関数は項目が表示されるまでツリーを水平方向にスクロールさせます。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExpandNode	この関数は nodeKey で指定したキーでノードを展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
FindNodeKeyByPath	指定したパス (2¥1¥2 など) のノード キーを返します。
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetAbapImage	ABAP 画像を取得します。
GetCheckBoxState	チェック ボックスの状態 (チェックされている場合は true、チェックされていない場合は false) を取得します。
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetColumnIndexFromName	指定した名前の列インデックスを取得します。
GetColumnKeys	指定した列のすべての項目のキーを取得します。
GetColumnNameFromName	指定した名前の列タイトルを取得します。
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHierarchyLevel	指定したキーの階層レベルを取得します。
GetItemHeight	現在の項目の高さを返します。
GetItemLeft	コンポーネントの左からの位置を取得します。
GetItemStyle	指定した項目のスタイルを取得します。
GetItemText	複数列のツリーの場合、この関数は、パラメーターで指定したキーと名前によって項目のテキストを返します。
GetItemTextColor	項目のテキスト色を返します。
GetItemTooltip	項目のツールチップ テキストを返します。
GetItemTop	コンポーネントの上からの位置を取得します。
GetItemType	指定したノードの種類を返します。
GetItemWidth	現在の項目の幅を返します。
GetListTreeNodeItemCount	ノード項目の数を返します。
GetNextNodeKey	次のノードのキーを返します。
GetNodeAbapImage	指定したノードの ABAP 画像を返します。
GetNodeChildrenCount	この関数は、指定したノードの子の数を返します。
GetNodeChildrenCountByPath	この関数は、パス パラメーター (2¥1¥2 など) によって指定されたノードの子の数を返します。
GetNodeHeight	現在のノードの高さを返します。
GetNodeIndex	指定したノードのインデックスを返します。
GetNodeItemHeaders	指定したノードのヘッダーを返します。
GetNodeLeft	コンポーネントの左からの位置を取得します。

名前	説明
GetNodePathByKey	ノードのキー (2¥1¥2 など) を指定してパスを取得します。
GetNodeStyle	ノードのスタイルを返します。
GetNodeText	この関数は、指定したキーによるノードのテキストを返します。
GetNodeTextByPath	指定したパスによって定義されるノードのテキストを返します。
GetNodeTextColor	ノードのテキスト色を返します。
GetNodeTooltip	ノードのツールチップ テキストを返します。
GetNodeTop	コンポーネントの上からの位置を取得します。
GetNodeWidth	現在のノードの幅を返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetParentNodeKey	指定したキーによるノードの親ノードのキーを返します。
GetPreviousNodeKey	指定したキーによるノードの前のノードのキーを返します。
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetStyleDescription	指定したスタイルの説明を返します。
GetSubNodesKeys	指定したキーのノードのすべてのサブノードのキーのコレクションを返します。
HeaderContextMenu	このメソッドは、ヘッダーに対してコンテキスト メニューを要求します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>IsFolder</i>	指定したノードがフォルダーかどうかを返します。
<i>IsFolderExpandable</i>	フォルダーが展開可能かどうかを返します。
<i>IsFolderExpanded</i>	フォルダーが展開されているかどうかを返します。
<i>IsHighLighted</i>	ノードがハイライトされているかどうかを返します。
<i>ItemContextMenu</i>	このメソッドは、項目に対してコンテキストメニューを要求します。
<i>NodeContextMenu</i>	このメソッドは、ノードに対してコンテキストメニューを要求します。
<i>PressKey</i>	このメソッドはキーの押下をエミュレートします。
<i>SelectButton</i>	このメソッドはボタンの押下をエミュレートします。
<i>SelectColumn</i>	この関数は列の選択範囲に列を追加します。ノードまたは項目の選択は削除されます。
<i>SelectContextMenuItem</i>	コントロールのコンテキストメニューから項目を選択します。 (<i>ISapContextMeneuable</i> から継承されました)
<i>SelectContextMenuItemByPosition</i>	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。(<i>ISapContextMeneuable</i> から継承されました)
<i>SelectContextMenuItemByText</i>	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。 (<i>ISapContextMeneuable</i> から継承されました)
<i>SelectHeader</i>	このメソッドはヘッダーのクリックをエミュレートします。
<i>SelectItem</i>	この関数は項目の選択をエミュレートします。この選択によって、すべてのほかの選択は削除されます。
<i>SelectLink</i>	この関数は、リンクのトリガーをエミュレートします。
<i>SelectNode</i>	nodeKey キーを持つノードをノードの選択範囲に追加します。
<i>SetCheckBoxState</i>	指定したノードのチェックボックスの状態を変更します。
<i>SetColumnOrder</i>	ツリーの列の順序を設定します。
<i>SetColumnWidth</i>	この関数は列の幅をピクセル単位で設定します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetHierarchyHeaderWidth</i>	階層ヘッダーの幅 (ピクセル単位) を設定します。
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>SetSelectedNode</i>	指定したノードを選択します。
<i>SetTopNode</i>	指定したキーをトップノードとして設定します。

名前	説明
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
UnselectAll	すべての選択を削除します。
UnselectColumn	この関数は列の選択範囲から列を削除します。
UnselectNode	nodeKey キーを持つノードをノードの選択範囲から削除します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイム

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapUserArea クラス

説明

SAP の UserArea クラスです。

継承階層

- SapComponent
 - SapUserArea

構文

```
'Declaration
Public Class SapUserArea _
Inherits SapComponent _
Implements ISapContextMenuable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
CurrentContextMenu	現在のコンテキスト メニュー。コンテキスト メニューがシェル オブジェクトで利用可能な場合にのみ設定されます。
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)

名前	説明
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
SelectContextMenuItem	コントロールのコンテキストメニューから項目を選択します。(ISapContextMenuable から継承されました)
SelectContextMenuItemByPosition	このメソッドは項目の位置を利用してコンテキストメニュー項目を選択できます。(ISapContextMenuable から継承されました)
SelectContextMenuItemByText	メニュー項目や利用可能な高レベルのメニューのテキストを使用してコンテキストメニューのメニュー項目を選択します。(ISapContextMenuable から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
ShowContextMenu	コントロールのコンテキストメニューを表示します。 (SapComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイム

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapVerticalScrollBar クラス

説明

垂直スクロールバーのクラスです。

継承階層

- [SapScrollbar](#)
 - SapVerticalScrollBar

構文

```
'Declaration
Public Class SapVerticalScrollBar _
Inherits SapScrollbar
```

プロパティ

名前	説明
Maximum	スクロールバーの最大値。(SapScrollbar から継承されました)
Minimum	スクロールバーの最小値。(SapScrollbar から継承されました)
PageSize	スクロールバーのページサイズ。(SapScrollbar から継承されました)
Position	スクロールバーの現在の位置。(SapScrollbar から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
ScrollTo	指定した位置にスクロールします。(SapScrollbar から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイム

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SapWindow クラス

説明

SapWindow クラスはトップ レベル ウィンドウのクラスです。

継承階層

- [SapComponent](#)
 - SapWindow

構文

```
'Declaration
Public Class SapWindow _
Inherits SapComponent _
Implements IMoveable
```

プロパティ

名前	説明
AccessibilityText	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTextOnRequest	アクセシビリティ サポートに関する追加のテキスト。(SapComponent から継承されました)
AccessibilityTooltip	アクセシビリティ サポートに関する追加のツールチップ テキスト。(SapComponent から継承されました)
ButtonbarVisible	アプリケーション ツールバー (SAP GUI の下のツールバー) が可視かどうか。
Changeable	オブジェクトが変更可能かどうか。(SapComponent から継承されました)
DefaultTooltip	指定した画面要素タイプのデータ辞書で定義された簡単なテキストから生成されるツールチップ テキスト。(SapComponent から継承されました)
GuiFocus	現在フォーカスを持つ SAP GUI 内の要素。
Height	現在のコンポーネントの高さ。(SapComponent から継承されました)
IconName	オブジェクトにアイコンが割り当てられている場合は、このプロパティはアイコンの名前を、それ以外の場合は空の文字列。(SapComponent から継承されました)
IsSymbolFont	コンポーネントのテキストが SAP シンボル フォントで表示されているかどうか。(SapComponent から継承されました)
Left	画面上の座標で要素の左からの位置。(SapComponent から継承されました)

名前	説明
<i>Modified</i>	状態がユーザーによって変更され、この変更がまだ SAP システムに送信されていないかどうか。(SapComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(SapComponent から継承されました)
<i>ScreenLeft</i>	画面上の座標でコンポーネントの左からの位置。(SapComponent から継承されました)
<i>ScreenTop</i>	画面上の座標でコンポーネントの上からの位置。(SapComponent から継承されました)
<i>StatusbarVisible</i>	ステータスバーが可視かどうか。
<i>SystemFocus</i>	サーバー アプリケーションの視点からフォーカスを持つ SAP GUI 内の要素。
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>TitlebarVisible</i>	タイトルバーが可視かどうか。
<i>ToolbarVisible</i>	ツールバーが可視かどうか。
<i>Tooltip</i>	ツールチップ テキスト。(SapComponent から継承されました)
<i>Top</i>	画面上の座標で要素の上からの位置。(SapComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(SapComponent から継承されました)
<i>WindowHandle</i>	GuiShell に接続しているコントロールのウィンドウ ハンドル。
<i>WindowState</i>	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)
<i>WorkingPaneHeight</i>	現在の作業ペインの高さ。
<i>WorkingPaneWidth</i>	現在の作業ペインの幅。

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Close</i>	ウィンドウを閉じます。(IMoveable から継承されました)
<i>CloseSynchron</i>	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)

名前	説明
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメータが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsActive</i>	ウィンドウがアクティブに設定されているかどうかを返します。(<i>IMoveable</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>IsVkeyAllowed</i>	このメソッドは、指定した仮想キーが現在の画面上で実行できる場合に true を返します。
<i>JumpBackward</i>	ウィンドウに対して Ctrl+Shift+Tab キーを実行して、1 ブロック後ろに移動します。
<i>JumpForward</i>	ウィンドウに対して Ctrl+Tab キーを実行して、1 ブロック前に移動します。
<i>Maximize</i>	ウィンドウを最大化します。(<i>IMoveable</i> から継承されました)
<i>Minimize</i>	ウィンドウを最小化します。(<i>IMoveable</i> から継承されました)
<i>Move</i>	ウィンドウを移動します。(<i>IMoveable</i> から継承されました)
<i>ResizeWorkingPane</i>	ResizeWorkingPane 関数は、文字単位で指定された幅と高さを利用可能な作業領域のウィンドウ サイズを変更します。
<i>ResizeWorkingPaneEx</i>	ResizeWorkingPaneEx 関数は、ピクセルで指定された幅と高さを利用可能な作業領域のウィンドウ サイズを変更します。
<i>Restore</i>	ウィンドウを元のサイズに戻します。(<i>IMoveable</i> から継承されました)
<i>SendVKey</i>	システムに仮想キーを送信します。
<i>SetActive</i>	ウィンドウをアクティブにします。(<i>IMoveable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>ShowContextMenu</i>	コントロールのコンテキストメニューを表示します。(<i>SapComponent</i> から継承されました)
<i>Size</i>	ウィンドウのサイズを変更します。(<i>IMoveable</i> から継承されました)

名前	説明
TabBackward	ウィンドウに対して Shift+Tab キーを実行して、1 要素後ろに移動します。
TabForward	ウィンドウに対して Tab キーを実行して、1 要素前に移動します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Silverlight クラス リファレンス

Silverlight コントロールのテストに利用可能なクラスを一覧します。

SLApplication クラス

説明

Silverlight コントロールのコンテナ。

継承階層

- [SLBase](#)
 - SLApplication

構文

```
'Declaration  
Public Class SLApplication _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLAutoCompleteBox クラス

説明

ユーザー入力可能なテキスト ボックスとテキスト ボックスへの入力を基にした有効な一致項目を含んだドロップ ダウンを提供するコントロールを表します。

継承階層

- [SLBase](#)

- SLAutoCompleteBox

構文

```
'Declaration
Public Class SLAutoCompleteBox _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
HorizontallyScrollable	コンボ ボックスを水平方向にスクロールできるかどうかを示す値。
HorizontalScrollPercent	現在の水平方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
IsOpen	オート コンプリート ボックスが開いているかどうかを示す値。
Items	現在の一致項目を含んだリスト。コントロールのテキスト ボックスに入力された項目がない場合は、空のリストを返します。
ItemTexts	現在の一致項目のテキストを含んだリスト。コントロールのテキスト ボックスに入力された項目がない場合は、空のリストを返します。
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
VerticallyScrollable	コンボ ボックスを垂直方向にスクロールできるかどうかを示す値。
VerticalScrollPercent	現在の垂直方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	一致項目を含んだドロップダウン リストが開いている場合に閉じます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollHorizontal</i>	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を水平方向にスクロールします。
<i>ScrollVertical</i>	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を垂直方向にスクロールします。
<i>Select</i>	オート コンプリート ボックスに指定したテキストを入力して、一致した項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetHorizontalScrollPercent</i>	コンボ ボックス内のコンテンツ領域全体のパーセンテージとして、水平スクロール位置を設定します。
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>SetVerticalScrollPercent</i>	コンボ ボックス内のコンテンツ領域全体のパーセンテージとして、垂直スクロール位置を設定します。

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLBase クラス

説明

Silverlight 要素の Silverlight フレームワーク レベルのプロパティ、イベント、およびメソッドのセットを提供します。

継承階層

- [TestObject](#)
 - SLBase
 - [SLApplication](#)
 - [SLAutoCompleteBox](#)
 - [SLButton](#)
 - [SLCalendar](#)
 - [SLCalendarButton](#)
 - [SLCalendarDayButton](#)
 - [SLCheckBox](#)
 - [SLComboBox](#)
 - [SLComboBoxItem](#)
 - [SLDataGrid](#)
 - [SLDataGridCell](#)
 - [SLDataGridDetails](#)
 - [SLDataGridRow](#)
 - [SLDataPager](#)
 - [SLDatePicker](#)
 - [SLDescriptionViewer](#)
 - [SLFrame](#)
 - [SLGridSplitter](#)
 - [SLGroup](#)
 - [SLHeader](#)
 - [SLHeaderItem](#)
 - [SLHorizontalScrollBar](#)
 - [SLHyperlinkButton](#)
 - [SLImage](#)
 - [SLListBox](#)
 - [SLListItem](#)
 - [SLMediaElement](#)
 - [SLMenu](#)
 - [SLMenuBar](#)
 - [SLMenuItem](#)
 - [SLMultiScaleImage](#)
 - [SLPane](#)
 - [SLPasswordBox](#)
 - [SLPopup](#)
 - [SLProgressBar](#)
 - [SLRadioButton](#)
 - [SLRepeatButton](#)
 - [SLRichTextBox](#)

- [SLSeparator](#)
- [SLSlider](#)
- [SLSpinner](#)
- [SLSplitButton](#)
- [SLStatusBar](#)
- [SLTabControl](#)
- [SLTabItem](#)
- [SLTable](#)
- [SLTextBlock](#)
- [SLTextBox](#)
- [SLThumb](#)
- [SLTitleBar](#)
- [SLToggleButton](#)
- [SLToolBar](#)
- [SLToolTip](#)
- [SLTreeView](#)
- [SLTreeViewItem](#)
- [SLValidationSummary](#)
- [SLVerticalScrollBar](#)
- [SLWindow](#)

構文

```
'Declaration
Public Class SLBase _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。
ClassName	要素の単純クラス名。
ControlType	要素に対するコントロールの種類を含んだ文字列。
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。
IsEnabled	要素が有効かどうかを示す値。
Name	要素に対する UI オートメーション名を含んだ文字列。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類 1 つを指定して、指定した種類の子のみを返すこともできます。
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLButton クラス

説明

ボタン コントロールを識別します。

継承階層

- [SLBase](#)
 - SLButton

構文

```
'Declaration  
Public Class SLButton _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRawChildren</i>	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(<i>SLBase</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	ボタンをクリックします。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SLCalendar クラス

説明

カレンダー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLCalendar

構文

```
'Declaration
Public Class SLCalendar _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	<p>要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)</p>

名前	説明
CanSelectMultiple	複数の日付を選択可能かどうか
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
DisplayMode	カレンダーで月、年、10年のいずれを表示するかを指定する値。
HelpText	要素に対するヘルプテキスト(ツールチップテキストなど)を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelectionRequired	少なくとも1つの日付を選択しなければならないかどうか
Name	要素に対するUIオートメーション名を含んだ文字列。(SLBase から継承されました)
SelectedDate	現在表示されているビューで選択されている日付。
SelectedDates	現在表示されているビューで選択されている日付のリスト。
SupportedDisplayModes	サポートされる表示モードのリスト。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値(たとえば、テキストコントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップイメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます(%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは32ビットRGB順で保存されます。それぞれのピクセルの4バイト目(アルファチャンネル)はパディングのため、イメージの解析時には無視されます。.NETでは、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウスボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeoutパラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐにfalseが返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SelectDate	指定した日付を選択します。
SelectDates	指定した日付を選択します。
SelectRange	日付の範囲を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SwitchToDecadeView	カレンダー表示モードを '10 年' に切り替えます (サポートされている場合)。
SwitchToMonthView	カレンダー表示モードを '月' に切り替えます (サポートされている場合)。
SwitchToYearView	カレンダー表示モードを '年' に切り替えます (サポートされている場合)。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLCalendarButton クラス

説明

カレンダー ボタンを識別します。

継承階層

- [SLBase](#)
 - SLCalendarButton

構文

```
'Declaration
Public Class SLCalendarButton _
Inherits SLBase
```

プロパティ

名前	説明
<i>AutomationId</i>	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (<i>SLBase</i> から継承されました)
<i>ClassName</i>	要素の単純クラス名。 (<i>SLBase</i> から継承されました)
<i>ControlType</i>	要素に対するコントロールの種類を含んだ文字列。 (<i>SLBase</i> から継承されました)
<i>HelpText</i>	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (<i>SLBase</i> から継承されました)
<i>IsEnabled</i>	要素が有効かどうかを示す値。 (<i>SLBase</i> から継承されました)
<i>IsSelected</i>	カレンダー ボタンが選択されているかどうかを示す値。
<i>Name</i>	要素に対する UI オートメーション名を含んだ文字列。 (<i>SLBase</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ボタンをクリックします。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	ムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLCalendarDayButton クラス

説明

カレンダーの日付ボタンを識別します。

継承階層

- [SLBase](#)
 - SLCalendarDayButton

構文

```
'Declaration
Public Class SLCalendarDayButton _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelected	カレンダーの日付ボタンが選択されているかどうかを示す値。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)

名前	説明
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ボタンをクリックします。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLCheckBox クラス

説明

チェック ボックス コントロールを識別します。

継承階層

- [SLBase](#)
 - SLCheckBox

構文

```
'Declaration
Public Class SLCheckBox _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToggleState	チェック ボックスのトグル状態
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル)

名前	説明
	<p>はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
<code>Check</code>	<p>チェック ボックスをチェックします。</p>
<code>Click</code>	<p>オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)</p>
<code>DoubleClick</code>	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)</p>
<code>Exists</code>	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)</p>
<code>Exists</code>	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)</p>
<code>Find</code>	<p>XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)</p>
<code>FindAll</code>	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)</p>
<code>GenerateLocator</code>	<p>このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)</p>
<code>GetChildren</code>	<p>このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)</p>
<code>GetDynamicMethodList</code>	<p>この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (<code>TestObject</code> から継承されました)</p>
<code>GetParent</code>	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (<code>TestObject</code> から継承されました)</p>
<code>GetProperty</code>	<p>指定したプロパティの値を返します。 (<code>TestObject</code> から継承されました)</p>
<code>GetPropertyList</code>	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (<code>TestObject</code> から継承されました)</p>
<code>GetRawChildren</code>	<p>この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。 (<code>SLBase</code> から継承されました)</p>
<code>GetRect</code>	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (<code>TestObject</code> から継承されました)</p>
<code>HighlightObject</code>	<p>このオブジェクトをハイライトします。 (<code>TestObject</code> から継承されました)</p>

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetIndeterminate</i>	チェック ボックスを不確定状態に設定します。
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Toggle	次の順番でサイクリックにチェック ボックスの状態を切り替えます： オン、オフ、不確定（サポートされる場合）。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	チェック ボックスのチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject スタートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLComboBox クラス

説明

コンボ ボックス コントロールを識別します。

Silverlight のコンボ ボックスでは、その項目についての情報を取得する前に一度開いておく必要があります。

継承階層

- [SLBase](#)
 - SLComboBox

構文

```
'Declaration  
Public Class SLComboBox _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
HorizontallyScrollable	コンボ ボックスを水平方向にスクロールできるかどうかを示す値。
HorizontalScrollPercent	現在の水平方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
IsOpen	コンボ ボックスが開いているかどうかを示す値。
Items	項目の完全なリスト。
ItemTexts	項目テキストの完全なリスト。
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
SelectedIndex	選択されている項目のインデックス。選択範囲が空の場合は -1 を返します。
SelectedItem	選択されている項目。選択範囲が空の場合は Nothing を返します。
SelectedItemText	選択されている項目のテキスト。選択範囲が空の場合は空の文字列を返します。
Text	コントロールのテキスト。 (TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VerticallyScrollable	コンボ ボックスを垂直方向にスクロールできるかどうかを示す値。
VerticalScrollPercent	現在の垂直方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	コンボ ボックスを閉じます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Open	コンボ ボックスを開きます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollHorizontal	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を水平方向にスクロールします。
ScrollVertical	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を垂直方向にスクロールします。
Select	コンボ ボックスの項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetHorizontalScrollPercent	コンボ ボックス内のコンテンツ領域全体のパーセンテージとして、水平スクロール位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetVerticalScrollPercent	コンボ ボックス内のコンテンツ領域全体のパーセンテージとして、垂直スクロール位置を設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLComboBoxItem クラス

説明

コンボ ボックス項目コントロールを識別します。

継承階層

- [SLBase](#)
 - SLComboBoxItem

構文

```
'Declaration
Public Class SLComboBoxItem _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelected	コンボ ボックス項目が選択されているかどうかを示す値。

名前	説明
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	コンボ ボックス項目を選択します。

名前	説明
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLDataGrid クラス

説明

データグリッド コントロールを識別します。

データグリッドに垂直方向のスクロールバーがある場合、データグリッドの表示領域にない行の情報(テキスト、キャプションなど)は、操作対象になりません。このような行に対して操作したり、情報を取得するためには、まずビューにスクロールして表示させる必要があります。

継承階層

- [SLBase](#)
 - SLDataGrid

構文

```
'Declaration
Public Class SLDataGrid _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
CanSelectMultiple	データグリッドが複数の行の同時選択を許可するかどうか。
ClassName	要素の単純クラス名。(SLBase から継承されました)
ColumnCount	データグリッドの列数。
ColumnHeaders	すべての列に対するヘッダー項目を含んだリスト
ColumnHeaderTexts	すべての列に対するヘッダー項目のテキストを含んだリスト
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
HorizontallyScrollable	データグリッドを水平方向にスクロールできるかどうかを示す値。
HorizontalScrollPercent	現在の水平方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)

名前	説明
RowCount	データグリッドの行数。
Rows	データグリッドの行のリスト。
RowTexts	データグリッド中の行テキストのリスト。すべての行のテキストが文字列として返されます。各セルのテキストは空白文字で分離されず。
SelectedCell	現在選択されているセル。セルが選択されていない場合は <code>Nothing</code>
SelectedCellText	現在選択されているセルのテキスト。セルが選択されていない場合は空の文字列。
SelectedRow	現在選択されている行。行が選択されていない場合は <code>Nothing</code>
SelectedRows	現在選択されている行のリスト。行が選択されていない場合は空のリスト
SelectedRowText	現在選択されている行のテキスト。行が選択されていない場合は空の文字列。
SelectedRowTexts	現在選択されている行のテキスト。行が選択されていない場合は空のリスト。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VerticallyScrollable	データグリッドを垂直方向にスクロールできるかどうかを示す値。
VerticalScrollPercent	現在の垂直方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRawChildren</i>	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(<i>SLBase</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollHorizontal	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を水平方向にスクロールします。
ScrollIntoView	指定した行またはセルをビューにスクロールします。
ScrollVertical	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を垂直方向にスクロールします。
Select	指定した行を選択します。
SelectCell	指定したセルを選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetHorizontalScrollPercent	データ グリッド内のコンテンツ領域全体のパーセンテージとして、水平スクロール位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetVerticalScrollPercent	データ グリッド内のコンテンツ領域全体のパーセンテージとして、垂直スクロール位置を設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLDataGridCell クラス

説明

データ グリッド コントロールの行内のセルを識別します。

継承階層

- [SLBase](#)
 - SLDataGridCell

構文

```
'Declaration  
Public Class SLDataGridCell _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
IsSelected	セルが選択されているかどうかを示す値
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	セルをビューにスクロールします。
<i>Select</i>	セルを選択します
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
	<p>アウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SLDataGridDetails クラス

説明

データグリッド行の詳細を表示するコントロールを識別します。

継承階層

- [SLBase](#)
 - SLDataGridDetails

構文

```
'Declaration
Public Class SLDataGridDetails _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)

名前	説明
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLDataGridRow クラス

説明

データグリッド内の行を識別します。

継承階層

- [SLBase](#)
 - SLDataGridRow

構文

```
'Declaration
Public Class SLDataGridRow _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
Cells	データグリッド行のセル。
CellTexts	データグリッド行のセルのテキスト。
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelected	行が選択されているかどうかを示す値。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
RowHeader	行のヘッダー
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	行をビューにスクロールします。
Select	行を選択します
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLDataPager クラス

説明

データのコレクションをページングするためのデータ ページャを識別します。

継承階層

- [SLBase](#)
 - SLDataPager

構文

```
'Declaration
Public Class SLDataPager _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
CurrentPage	現在のページ。
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
PageCount	ページ数。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MoveToFirstPage	データ ページの最初のページに移動します。
MoveToLastPage	データ ページの最後のページに移動します。
MoveToNextPage	データ ページの次のページに移動します。
MoveToPage	データ ページの指定したページに移動します。
MoveToPreviousPage	データ ページの前のページに移動します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLDatePicker クラス

説明

ユーザーが日付を選択できるコントロールを表します。

継承階層

- [SLBase](#)
 - SLDatePicker

構文

```
'Declaration  
Public Class SLDatePicker _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsCalendarOpen	ドロップダウン カレンダーが開いているかどうか。
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
SelectedDate	選択されている日付。日付が選択されていない場合は <code>Nothing</code> を返します。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
CloseCalendar	ドロップダウン カレンダーを閉じます。

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenCalendar	ドロップダウン カレンダーを開きます。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLDescriptionViewer クラス

説明

説明ビューアー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLDescriptionViewer

構文

```
'Declaration
Public Class SLDescriptionViewer _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForObject	トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLFrame クラス

説明

フレーム コントロールを識別します。

継承階層

- [SLBase](#)
 - SLFrame

構文

```
'Declaration
Public Class SLFrame _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)

名前	説明
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLGridSplitter クラス

説明

グリッド スプリッタ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLGridSplitter

構文

```
'Declaration
Public Class SLGridSplitter _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MoveHorizontal	スプリッタを水平方向に移動します。
MoveVertical	スプリッタを垂直方向に移動します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SLGroup クラス

説明

他のコントロールのコンテナとして動作するグループ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLGroup

構文

```
'Declaration  
Public Class SLGroup _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLHeader クラス

説明

行または列のラベルを表すビジュアル コンテナを提供するコントロールを識別します。

継承階層

- [SLBase](#)
 - SLHeader

構文

```
'Declaration
Public Class SLHeader _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HeaderItems	このヘッダー コントロールのヘッダー項目を含むリスト

名前	説明
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLHeaderItem クラス

説明

行または列のビジュアル ラベルを提供するコントロールを識別します。

継承階層

- [SLBase](#)
 - SLHeaderItem

構文

```
'Declaration
Public Class SLHeaderItem _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	<p>この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類 1 つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)</p>
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	ヘッダー項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLHorizontalScrollBar クラス

説明

水平方向のスクロールバーを識別します。

継承階層

- [SLBase](#)
 - SLHorizontalScrollBar

構文

```
'Declaration
Public Class SLHorizontalScrollBar _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Maximum	最大スクロール位置
Minimum	最小スクロール位置
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
ScrollPosition	現在のスクロール位置
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Scroll	スクロール バーを指定した量スクロールします
ScrollToMaximum	スクロール バーを右端にスクロールします。
ScrollToMinimum	スクロール バーを左端にスクロールします。
ScrollToPosition	スクロール バーを指定した位置にスクロールします
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLHyperlinkButton クラス

説明

ハイパーリンク ボタン コントロールを識別します。

継承階層

- [SLBase](#)
 - SLHyperlinkButton

構文

```
'Declaration
Public Class SLHyperlinkButton _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	ハイパーリンク ボタンをクリックします。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	タイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLImage クラス

説明

イメージ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLImage

構文

```
'Declaration
Public Class SLImage _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)

名前	説明
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLListBox クラス

説明

リスト ボックス コントロールを識別します。

継承階層

- [SLBase](#)
 - SLListBox

構文

```
'Declaration
Public Class SLListBox _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
CanSelectMultiple	リスト ボックスが複数の子要素の同時選択を許可するかどうか。
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
HorizontallyScrollable	リスト ボックスを水平方向にスクロールできるかどうかを示す値。
HorizontalScrollPercent	現在の水平方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Items	リスト ボックス項目のリスト。
ItemTexts	すべての項目テキストのリスト。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
SelectedIndex	現在選択されている最初の項目のインデックス。選択範囲が空の場合は -1 を返します。
SelectedIndices	現在選択されている項目のインデックス。項目が現在選択されていない場合は、空のリストが返されます。
SelectedItem	現在選択されている最初の項目。選択範囲が空の場合は Nothing を返します

名前	説明
SelectedItems	現在選択されている項目。項目が現在選択されていない場合は、空のリストが返されます。
SelectedItemText	最初の選択された項目のテキスト。
SelectedItemTexts	現在選択されている項目のテキスト。項目が現在選択されていない場合は、空のリストが返されます。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VerticallyScrollable	リスト ボックスを垂直方向にスクロールできるかどうかを示す値。
VerticalScrollPercent	現在の垂直方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DeselectAll	すべてのリスト項目の選択を解除します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollHorizontal	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を水平方向にスクロールします。
ScrollVertical	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を垂直方向にスクロールします。
Select	指定したリスト ボックス項目を選択します。
SelectAll	リスト内のすべての項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetHorizontalScrollPercent	リスト ボックス内のコンテンツ領域全体のパーセンテージとして、水平スクロール位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetVerticalScrollPercent	リスト ボックス内のコンテンツ領域全体のパーセンテージとして、垂直スクロール位置を設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLListItem クラス

説明

リスト項目コントロールを識別します。

継承階層

- [SLBase](#)
 - SLListItem

構文

```
'Declaration
Public Class SLListItem _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)

名前	説明
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelected	リスト項目が選択されているかどうかを示す値。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Deselect	項目の選択を解除します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	リスト項目をビューにスクロールします。
Select	リスト項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForObject	値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLMediaElement クラス

説明

オーディオとビデオまたはそのいずれかを含むコントロールを表します。

継承階層

- [SLBase](#)
 - SLMediaElement

構文

```
'Declaration
Public Class SLMediaElement _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)

名前	説明
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLMenu クラス

説明

メニュー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLMenu

構文

```
'Declaration
Public Class SLMenu _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。(<i>IKeyable</i> から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLMenuBar クラス

説明

メニュー バー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLMenuBar

構文

```
'Declaration  
Public Class SLMenuBar _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLMenuItem クラス

説明

メニュー項目コントロールを識別します。

継承階層

- [SLBase](#)
 - SLMenuItem

構文

```
'Declaration
Public Class SLMenuItem _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)

名前	説明
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLMultiScaleImage クラス

説明

ユーザーがズームやパンに対応した多重解像度イメージを開くことができますようにします。

継承階層

- [SLBase](#)
 - SLMultiScaleImage

構文

```
'Declaration
Public Class SLMultiScaleImage _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類 1 つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLPane クラス

説明

スクロールバーを持つことが可能なペインを識別します。

継承階層

- [SLBase](#)
 - SLPane

構文

```
'Declaration  
Public Class SLPane _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
HorizontallyScrollable	ペインを水平方向にスクロールできるかどうかを示す値。
HorizontalScrollPercent	現在の水平方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
VerticallyScrollable	ペインを垂直方向にスクロールできるかどうかを示す値。
VerticalScrollPercent	現在の垂直方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollHorizontal	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を水平方向にスクロールします。
ScrollVertical	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を垂直方向にスクロールします。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetHorizontalScrollPercent	ペイン内のコンテンツ領域全体のパーセンテージとして、水平スクロール位置を設定します。
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SetVerticalScrollPercent	ペイン内のコンテンツ領域全体のパーセンテージとして、垂直スクロール位置を設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SLPasswordBox クラス

説明

パスワード ボックス コントロールを識別します。

継承階層

- [SLBase](#)
 - SLPasswordBox

構文

```
'Declaration  
Public Class SLPasswordBox _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
IsReadOnly	パスワード ボックスが読み取り専用かどうか
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをパスワード フィールドから削除します
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)

名前	説明
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRawChildren</i>	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(<i>SLBase</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetPosition	パスワード フィールド内の挿入位置を設定します。
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SetSelection	パスワード ボックス内のテキストの範囲を選択します。
SetText	パスワード フィールド内のテキストを指定したテキストで置換します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定し

名前	説明
WaitForDisappearance	<p>ない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SLPopup クラス

説明

ポップアップ コントロールを識別します。コンテキスト メニューもポップアップ コントロールとして扱われます。

継承階層

- [SLBase](#)
 - SLPopup

構文

```
'Declaration
Public Class SLPopup _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
	トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLProgressBar クラス

説明

進行状況バー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLProgressBar

構文

```
'Declaration
Public Class SLProgressBar _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Maximum	進行状況の最大値
Minimum	進行状況の最小値
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)

名前	説明
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLRadioButton クラス

説明

ユーザーがオプションのグループから単一のオプションを選択できるボタンを表します。

継承階層

- SLBase
 - SLRadioButton

構文

```
'Declaration
Public Class SLRadioButton _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelected	ラジオ ボタンが選択されているかどうかを示す値。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル)

名前	説明
	<p>はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケータを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRawChildren	<p>この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	任意の選択された項目の選択を解除し、現在の要素を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SLRepeatButton クラス

説明

リピート ボタン コントロールを識別します。

継承階層

- [SLBase](#)
 - SLRepeatButton

構文

```
'Declaration  
Public Class SLRepeatButton _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	ボタンをクリックします。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLRichTextBox クラス

説明

リッチ テキスト ボックス コントロールを識別します。

継承階層

- [SLBase](#)
 - SLRichTextBox

構文

```
'Declaration
Public Class SLRichTextBox _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)

名前	説明
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
SelectedText	選択したテキスト
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Clear	すべてのテキストをテキスト ボックスから削除します
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetPosition	テキスト ボックス内の挿入位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelection	テキスト ボックス内のテキストの範囲を選択します。
SetText	テキスト ボックスの内容を指定したテキストで置換します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLSeparator クラス

説明

セパレータ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLSeparator

構文

```
'Declaration
Public Class SLSeparator _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLSlider クラス

説明

進行状況バー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLSlider

構文

```
'Declaration  
Public Class SLSlider _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
LargeChange	PAGE DOWN キーが押された場合など、大きい単位での変更が行われる際に、Value プロパティに加算または減算される値。
Maximum	進行状況の最大値
Minimum	進行状況の最小値
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
SmallChange	矢印キーが押された場合など、小さい単位での変更が行われる際に、Value プロパティに加算または減算される値。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DecreaseLarge	Slider の値を大きい単位で減らします。
DecreaseSmall	Slider の値を小さい単位で減らします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>IncreaseLarge</i>	Slider の値を大きい単位で増やします。
<i>IncreaseSmall</i>	Slider の値を小さい単位で増やします。
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>SetValue</i>	スライダの値を設定します。
<i>SetValueToMaximum</i>	Slider を最大の位置に設定します。

名前	説明
SetValueToMinimum	Slider を最小の位置に設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLSpinner クラス

説明

スピン コントロールを識別します。

Spinner コントロール型は一連のボタンで構成されます。ユーザーはこれらのボタンを使用して、一連の項目の中から項目を選択したり、範囲内の数値を設定したりできます。

継承階層

- [SLBase](#)
 - SLSpinner

構文

```
'Declaration  
Public Class SLSpinner _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
NumericValue	スピン コントロールの現在の数値
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。(<i>IKeyable</i> から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLSplitButton クラス

説明

分割ボタン コントロールを識別します。

継承階層

- [SLBase](#)
 - SLSplitButton

構文

```
'Declaration  
Public Class SLSplitButton _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRawChildren</i>	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(<i>SLBase</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLStatusBar クラス

説明

ステータスバー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLStatusBar

構文

```
'Declaration
Public Class SLStatusBar _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)

名前	説明
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLTabControl クラス

説明

タブ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLTabControl

構文

```
'Declaration
Public Class SLTabControl _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Items	タブ項目の完全なリスト。
ItemTexts	タブ項目テキストの完全なリスト。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
SelectedIndex	選択されている項目のインデックス。選択範囲が空の場合は -1 を返します。
SelectedItem	選択されている項目。選択範囲が空の場合は <code>Nothing</code> を返します。
SelectedItemText	選択されている項目のテキスト。選択範囲が空の場合は空の文字列を返します。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)

名前	説明
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	タブ コントロールの項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLTabItem クラス

説明

タブ項目コントロールを識別します。

継承階層

- SLBase
 - SLTabItem

構文

```
'Declaration
Public Class SLTabItem _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsSelected	タブ項目が選択されているかどうかを示す値。
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル)

名前	説明
	<p>はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetDynamicMethodList	<p>この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)</p>
GetParent	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)</p>
GetProperty	<p>指定したプロパティの値を返します。 (TestObject から継承されました)</p>
GetPropertyList	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)</p>
GetRawChildren	<p>この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)</p>
GetRect	<p>このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)</p>
HighlightObject	<p>このオブジェクトをハイライトします。 (TestObject から継承されました)</p>
ImageClick	<p>指定したイメージ資産をクリックします。 (TestObject から継承されました)</p>

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Select</i>	タブ項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SLTable クラス

説明

テーブルを識別します。

継承階層

- [SLBase](#)
 - SLTable

構文

```
'Declaration  
Public Class SLTable _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLTextBlock クラス

説明

少量のフロー コンテンツを表示するコントロールを識別します。

継承階層

- [SLBase](#)
 - SLTextBlock

構文

```
'Declaration
Public Class SLTextBlock _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)

名前	説明
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLTextBox クラス

説明

テキスト ボックス コントロールを識別します。

継承階層

- [SLBase](#)
 - SLTextBox

構文

```
'Declaration
Public Class SLTextBox _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
IsReadOnly	テキスト ボックスが読み取り専用かどうか
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをテキスト フィールドから削除します
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetPosition	テキスト フィールド内の挿入位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelection	テキスト ボックス内のテキストの範囲を選択します。

名前	説明
SetText	テキスト フィールド内のテキストを指定したテキストで置換します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

例

```
textBox.SetText("some text")
textBox.SetSelection(0, 4) ' 最初の文字から 4 文字を選択
textBox.TypeKeys("<Ctrl+C>") ' 選択したテキストをクリップボードにコピー
Dim selectedText = Clipboard.GetText() ' クリップボードからテキストを取得
```

SLThumb クラス

説明

サム コントロールを識別します。

継承階層

- [SLBase](#)
 - SLThumb

構文

```
'Declaration
Public Class SLThumb _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル)

名前	説明
	はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SLTitleBar クラス

説明

ウィンドウのタイトルバーを識別します。

継承階層

- [SLBase](#)
 - SLTitleBar

構文

```
'Declaration  
Public Class SLTitleBar _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRawChildren</i>	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(<i>SLBase</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)
<i>WaitForDisappearance</i>	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLToggleButton クラス

説明

トグル ボタン コントロールを識別します。

継承階層

- [SLBase](#)
 - SLToggleButton

構文

```
'Declaration
Public Class SLToggleButton _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)

名前	説明
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToggleState	ボタンのトグル状態
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Toggle	次の順番でサイクリックにボタンの状態を切り替えます： オン、オフ、不確定 (サポートされる場合)。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

SLToolBar クラス

説明

ツールバー コントロールを識別します。

継承階層

- [SLBase](#)
 - SLToolBar

構文

```
'Declaration
Public Class SLToolBar _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLToolTip クラス

説明

ツールチップ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLToolTip

構文

```
'Declaration  
Public Class SLToolTip _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLTreeView クラス

説明

項目を展開したり折りたたんだりできるツリー構造で階層データを表示するコントロールを表します。

継承階層

- [SLBase](#)
 - SLTreeView

構文

```
'Declaration
Public Class SLTreeView _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
HorizontallyScrollable	ツリー ビューを水平方向にスクロールできるかどうかを示す値。
HorizontalScrollPercent	現在の水平方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
RootItems	すべてのルート ツリー項目の完全なリスト。
RootItemTexts	すべてのルート ツリー項目のテキストの完全なリスト。
SelectedItem	選択されている項目。選択範囲が空の場合は <code>Nothing</code> を返します。
SelectedItemPath	選択された項目パスを取得します。選択されていない場合は空の項目パス。
SelectedItemText	選択されている項目のテキスト。選択範囲が空の場合は空の文字列を返します。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
VerticallyScrollable	ツリー ビューを垂直方向にスクロールできるかどうかを示す値。
VerticalScrollPercent	現在の垂直方向スクロールの位置。有効なスクロール位置が存在しない場合は -1 を返します。
VisibleItemPaths	ツリーに表示される項目の項目パス。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
Collapse	指定した項目を折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Expand	指定した項目を展開します。
ExpandAll	ツリー内のすべての項目を展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類 の 1 つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollHorizontal</i>	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を水平方向にスクロールします。
<i>ScrollVertical</i>	指定した ScrollAmount の分、コンテンツ領域の現在の可視領域を垂直方向にスクロールします。
<i>Select</i>	指定した項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetHorizontalScrollPercent</i>	ツリー ビュー内のコンテンツ領域全体のパーセンテージとして、水平スクロール位置を設定します。
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>SetVerticalScrollPercent</i>	ツリー ビュー内のコンテンツ領域全体のパーセンテージとして、垂直スクロール位置を設定します。

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLTreeViewItem クラス

説明

ツリー コントロール内の選択可能な項目を表します。

継承階層

- [SLBase](#)
 - SLTreeViewItem

構文

```
'Declaration  
Public Class SLTreeViewItem _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
IsExpanded	ツリー項目内の入れ子になった項目が展開されているか、折りたたまれているか。
IsLeaf	ツリー ビュー項目がリーフ項目かどうかを示す値。
IsSelected	ツリー項目が選択されているかどうかを示す値。
Items	項目の完全なリスト。
ItemTexts	項目テキストの完全なリスト。
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Collapse	項目を折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Expand	項目を展開します。
ExpandSubTree	このツリー項目の下のサブツリーを展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類を 1 つを指定して、指定した種類の子のみを返すこともできます。 (SLBase から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	ツリー項目をビューにスクロールします
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLValidationSummary クラス

説明

検証概要コントロールを識別します。

継承階層

- [SLBase](#)
 - SLValidationSummary

構文

```
'Declaration  
Public Class SLValidationSummary _  
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLVerticalScrollBar クラス

説明

垂直方向のスクロールバーを識別します。

継承階層

- [SLBase](#)
 - SLVerticalScrollBar

構文

```
'Declaration
Public Class SLVerticalScrollBar _
Inherits SLBase
```


プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。 (SLBase から継承されました)
ClassName	要素の単純クラス名。 (SLBase から継承されました)
ControlType	要素に対するコントロールの種類を含んだ文字列。 (SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。 (SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。 (SLBase から継承されました)
Maximum	最大スクロール位置
Minimum	最小スクロール位置
Name	要素に対する UI オートメーション名を含んだ文字列。 (SLBase から継承されました)
ScrollPosition	現在のスクロール位置
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。 このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。 MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Scroll	スクロール バーを指定した量スクロールします
ScrollToMaximum	スクロール バーを最下部にスクロールします。
ScrollToMinimum	スクロール バーを先頭にスクロールします。
ScrollToPosition	スクロール バーを指定した位置にスクロールします
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

SLWindow クラス

説明

ウィンドウ コントロールを識別します。

継承階層

- [SLBase](#)
 - SLWindow

構文

```
'Declaration
Public Class SLWindow _
Inherits SLBase
```

プロパティ

名前	説明
AutomationId	要素に対する UI オートメーション識別子 (ID) を含んだ文字列。(SLBase から継承されました)
ClassName	要素の単純クラス名。(SLBase から継承されました)

名前	説明
ControlType	要素に対するコントロールの種類を含んだ文字列。(SLBase から継承されました)
HelpText	要素に対するヘルプ テキスト (ツールチップ テキストなど) を含んだ文字列。(SLBase から継承されました)
IsEnabled	要素が有効かどうかを示す値。(SLBase から継承されました)
Name	要素に対する UI オートメーション名を含んだ文字列。(SLBase から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRawChildren	この要素の生の子要素を取得します。このメソッドは、他の方法では子として認識されない要素にアクセスするために使用することができます。MSUIA コントロールの種類の一つを指定して、指定した種類の子のみを返すこともできます。(SLBase から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してくだ

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Java AWT および Swing クラス リファレンス

Java AWT および Swing コントロールのテストに利用可能なクラスを一覧します。

AbstractButton クラス

説明

ボタンやメニュー項目に共通の動作を定義します。

継承階層

- [JComponent](#)
 - AbstractButton
 - [JButton](#)
 - [JMenuItem](#)
 - [JToggleButton](#)

構文

```
'Declaration
Public Class AbstractButton _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)

名前	説明
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。 (<i>AWTComponent</i> から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(<i>AWTComponent</i> から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。 (<i>AWTComponent</i> から継承されました)
<i>Font</i>	コンポーネントのフォント。(<i>AWTComponent</i> から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(<i>AWTComponent</i> から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(<i>AWTComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。(<i>AWTComponent</i> から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (<i>INativeWindow</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>ToolTipText</i>	Swing コンポーネントのツールチップ テキスト。(<i>JComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>Select</i>	ボタンを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

Applet クラス

説明

ブラウザの Java アプレットを表します。

継承階層

- [AWTContainer](#)
 - Applet

構文

```
'Declaration
Public Class Applet _
Inherits AWTContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)

名前	説明
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(TestObject から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む)のリストを返します。(TestObject から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(TestObject から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(TestObject から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(TestObject から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(TestObject から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

AppletContainer クラス

説明

ブラウザのアプレットのルート要素です。

継承階層

- [AWTContainer](#)
 - AppletContainer

構文

```
'Declaration
Public Class AppletContainer _
Inherits AWTContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)

名前	説明
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

AWTButton クラス

説明

プッシュ ボタンを表します。

継承階層

- [AWTComponent](#)
 - AWTButton

構文

```
'Declaration
Public Class AWTButton _
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)

名前	説明
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
Select	ボタンを選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの

名前	説明
WaitForDisappearance	'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTCanvas クラス

説明

アプリケーションが描画したりユーザーからのイベントを補足できる画面上の空の矩形領域を表します。

継承階層

- [AWTComponent](#)
 - AWTCanvas

構文

```
'Declaration
Public Class AWTCanvas _
Inherits AWTComponent
```


プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。 (AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。 (AWTComponent から継承されました)
Background	コンポーネントの背景色。 (AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。 (AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTCheckbox クラス

説明

チェック ボックスを表します。

継承階層

- [AWTComponent](#)
 - AWTCheckbox

構文

```
'Declaration
Public Class AWTCheckbox _
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	チェック ボックスがチェックされているかどうか。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	チェック ボックスをチェックします。

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Toggle	チェック ボックスの状態を切り替えます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Uncheck	チェック ボックスのチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTCheckboxMenuItem クラス

説明

メニュー項目に挿入できるチェック ボックス コントロールを表します。

継承階層

- [AWTMenuItem](#)
 - AWTCheckboxMenuItem

構文

```
'Declaration  
Public Class AWTCheckboxMenuItem _  
Inherits AWTMenuItem
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTMenuItem から継承されました)
Checked	メニュー項目のチェック ボックスがチェックされているかどうか。
Enabled	メニュー項目が有効かどうか。(AWTMenuItem から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	メニュー項目のチェック ボックスをチェックします。
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	メニュー項目を選択します。(AWTMenuItem から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Toggle	メニュー項目のチェック ボックスの状態を切り替えます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	メニュー項目のチェック ボックスをチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTChoice クラス

説明

コンボ ボックスを表します。

継承階層

- [AWTComponent](#)
 - AWTChoice

構文

```
'Declaration
Public Class AWTChoice _
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)

名前	説明
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
ItemCount	コンボ ボックスの項目の数。
Items	コンボ ボックス項目のリスト。
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
SelectedIndex	選択されている項目のインデックス。項目が現在選択されていない場合には -1 (4Test の場合は 0) を返します。
SelectedItem	選択されている項目。項目が現在選択されていない場合には、NULL が返されます。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetItem</i>	指定したインデックスの項目を返します。
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
Select	コンボ ボックスで指定した項目を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの

名前	説明
WaitForDisappearance	'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTComponent クラス

説明

AWTComponent は Java AWT の基本クラスです。

継承階層

- [TestObject](#)
 - AWTComponent
 - [AWTButton](#)
 - [AWTCanvas](#)
 - [AWTCheckbox](#)
 - [AWTChoice](#)
 - [AWTContainer](#)
 - [AWTLabel](#)
 - [AWTList](#)
 - [AWTRadioButton](#)

- [AWTScrollbar](#)
- [AWTTextComponent](#)

構文

```
'Declaration
Public Class AWTComponent _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable, INativeWindow
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。
AccessibleRole	コンポーネントのアクセス可能なロール。
Background	コンポーネントの背景色。
Cursor	コンポーネントに設定されているカーソルの名前。
Enabled	コンポーネントが有効かどうか。
Focusable	コンポーネントをフォーカスすることができるかどうか。
Font	コンポーネントのフォント。
Foreground	コンポーネントの前景色。
Height	現在のコンポーネントの高さ。
Name	コンポーネントの名前。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの

名前	説明
WaitForDisappearance	'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTContainer クラス

説明

他の AWT コンポーネントを含めることができるコンポーネントを表します。

継承階層

- [AWTComponent](#)
 - AWTContainer
 - [Applet](#)
 - [AppletContainer](#)
 - [AWTScrollPane](#)
 - [AWTWindow](#)
 - [JComponent](#)
 - [OracleFormsContainer](#)
 - [SplitPaneDivider](#)

構文

```
'Declaration  
Public Class AWTContainer _  
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTDialog クラス

説明

ダイアログ ボックスを表します。

継承階層

- [AWTWindow](#)
 - AWTDialog
 - [JDialog](#)

構文

```
'Declaration  
Public Class AWTDialog _  
Inherits AWTWindow
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Modal	ダイアログがモーダルかどうか。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Resizable	ユーザーがダイアログのサイズを変更できるかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Title	ダイアログのタイトル。
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Close	ウィンドウを閉じます。 (IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。 (IMoveable から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。 (IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)

名前	説明
getParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。 (IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTFrame クラス

説明

タイトルと境界線を持つトップレベル ウィンドウを表します。

継承階層

- [AWTWindow](#)
 - AWTFrame
 - [JFrame](#)

構文

```
'Declaration
Public Class AWTFrame _
Inherits AWTWindow
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)

名前	説明
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Resizable	ユーザーがフレームのサイズを変更できるかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Title	フレームのタイトル。
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 Size 	ウィンドウのサイズを変更します。(IMoveable から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTHorizontalScrollbar クラス

説明

水平方向のスクロールバーを表します。

継承階層

- [AWTScrollbar](#)
 - AWTHorizontalScrollbar

構文

```
'Declaration
Public Class AWTHorizontalScrollbar _
Inherits AWTScrollbar
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
PageSize	スクロールバーのブロック増分値。(IAWTScroller から継承されました)
Position	スクロールバーの現在の位置。(IAWTScroller から継承されました)
Range	スクロールバーの範囲値。(IAWTScroller から継承されました)
StepSize	スクロールバーの単位増分値。(IAWTScroller から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
Page	指定したページ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTLabel クラス

説明

コンテナ内に配置されるテキストのクラスを表します。

継承階層

- [AWTComponent](#)
 - AWTLabel

構文

```
'Declaration
Public Class AWTLabel _
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Alignment	ラベルの配置。
Background	コンポーネントの背景色。(AWTComponent から継承されました)

名前	説明
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
WaitForObject	<p>ウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

AWTList クラス

説明

リスト ボックスを表します。

継承階層

- [AWTComponent](#)
 - AWTList

構文

```
'Declaration
Public Class AWTList _
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>ItemCount</i>	リスト項目の数。
<i>Items</i>	リスト項目のリスト。
<i>MultipleMode</i>	リストが複数選択を許可するかどうか。
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
<i>SelectedIndex</i>	現在選択されている項目のインデックス。項目が現在選択されていない場合には -1 (4Test の場合は 0) を返します。
<i>SelectedIndices</i>	現在選択されている項目のインデックス。項目が現在選択されていない場合は、空のリストが返されます。
<i>SelectedItem</i>	現在選択されているリスト項目。項目が現在選択されていない場合には、NULL が返されます。
<i>SelectedItems</i>	現在選択されているリスト項目 (複数)。項目が現在選択されていない場合は、空のリストが返されます。
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保

名前	説明
	<p>存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Deselect	指定したリスト項目の選択を解除します。
DeselectAll	すべてのリスト項目の選択を解除します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClickItem	リスト項目をダブルクリックします。
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetItem	指定したインデックスのリスト項目を返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>Select</i>	指定した項目を選択します。
<i>SelectAll</i>	リスト内のすべての項目を選択します。
<i>SelectRange</i>	指定した範囲のすべての項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTMenu クラス

説明

メニュー バーからアクセスされるメニュー コンポーネントを表します。

継承階層

- [AWTMenuItem](#)
 - AWTMenu

構文

```
'Declaration  
Public Class AWTMenu _  
Inherits AWTMenuItem
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。 (AWTMenuItem から継承されました)
Enabled	メニュー項目が有効かどうか。 (AWTMenuItem から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	メニュー項目を選択します。(AWTMenuItem から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTMenuComponent クラス

説明

すべてのメニュー関連コントロールの基本クラスを表します。

継承階層

- [TestObject](#)
 - AWTMenuComponent
 - [AWTMenuItem](#)

構文

```
'Declaration
Public Class AWTMenuComponent _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable
```

プロパティ

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTMenuItem クラス

説明

メニュー項目を表します。

継承階層

- [AWTMenuItemComponent](#)
 - AWTMenuItem
 - [AWTCheckboxMenuItem](#)
 - [AWTMenu](#)

構文

```
'Declaration  
Public Class AWTMenuItem _  
Inherits AWTMenuItemComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。
Enabled	メニュー項目が有効かどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	メニュー項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	ムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTRadioButton クラス

説明

ラジオ ボタンを表します。

継承階層

- [AWTComponent](#)
 - AWTRadioButton

構文

```
'Declaration
Public Class AWTRadioButton _
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	ラジオ ボタンがチェックされているかどうか。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)

名前	説明
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。 (<i>AWTComponent</i> から継承されました)
<i>Font</i>	コンポーネントのフォント。 (<i>AWTComponent</i> から継承されました)
<i>Foreground</i>	コンポーネントの前景色。 (<i>AWTComponent</i> から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。 (<i>AWTComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。 (<i>AWTComponent</i> から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (<i>INativeWindow</i> から継承されました)
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。 (<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Check</i>	ラジオ ボタンをチェックします。
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)

名前	説明
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTScrollbar クラス

説明

スクロールバーを表します。このクラスはリスト ボックス上のスクロールバーのような、コントロールの部品であるスクロールバーを含みます。

継承階層

- [AWTComponent](#)
 - AWTScrollbar
 - [AWTHorizontalScrollbar](#)
 - [AWTVerticalScrollbar](#)

構文

```
'Declaration
Public Class AWTScrollbar _
Inherits AWTComponent _
Implements IAWTScroller
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロール バーのブロック増分値。(IAWTScroller から継承されました)
Position	スクロール バーの現在の位置。(IAWTScroller から継承されました)
Range	スクロール バーの範囲値。(IAWTScroller から継承されました)
StepSize	スクロール バーの単位増分値。(IAWTScroller から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
Page	指定したページ数を現在のスクロール位置に加算します。 (IAWTScroller から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。 (IAWTScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTScrollPane クラス

説明

単一の子コンポーネントに対する水平方向または垂直方向の自動スクロールを実装するクラスを表します。

継承階層

- [AWTContainer](#)
 - AWTScrollPane

構文

```
'Declaration
Public Class AWTScrollPane _
Inherits AWTContainer _
Implements IAWTScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Position	スクロール可能コンポーネントの現在のスクロール位置。(IAWTScrollable から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	水平方向のスクロール バーを取得します。 (IAWTScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetVerticalBar	垂直方向のスクロールバーを取得します。(IAWTScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScrollable から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTTextArea クラス

説明

テキスト領域を表します。

継承階層

- [AWTTextComponent](#)
 - AWTTextArea

構文

```
'Declaration
Public Class AWTTextArea _
Inherits AWTTextComponent _
Implements IAWTScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(AWTTextComponent から継承されました)
Columns	テキスト領域の列の数。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。(AWTTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)

名前	説明
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Position	スクロール可能コンポーネントの現在のスクロール位置。 (IAWTScrollable から継承されました)
Rows	テキスト領域の行の数。
SelectedText	現在選択されているテキスト。(AWTTextComponent から継承されました)
SelectionEnd	選択されたテキストの終了位置。(AWTTextComponent から継承されました)
SelectionStart	選択されたテキストの開始位置。(AWTTextComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Clear	すべてのテキストをテキスト コンポーネントから削除します (AWTTextComponent から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	水平方向のスクロールバーを取得します。(IAWTScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	垂直方向のスクロールバーを取得します。(IAWTScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Insert	指定したテキストを指定した位置に挿入します。

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReplaceRange	指定した開始位置と終了位置の間にあるテキストを、指定した置換テキストで置き換えます。
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。 (IAWTScrollable から継承されました)
SelectAll	テキスト コンポーネント内のすべてのテキストを選択します。 (AWTTextComponent から継承されました)
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。 (AWTTextComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。 (AWTTextComponent から継承されました)
SetText	テキスト コンポーネントのテキストを指定したテキストに設定します。 (AWTTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTTextComponent クラス

説明

AWTTextField クラス、および AWTTextArea クラスの基本クラスを表します。

継承階層

- [AWTComponent](#)
 - AWTTextComponent
 - [AWTTextArea](#)
 - [AWTTextField](#)

構文

```
'Declaration  
Public Class AWTTextComponent _  
Inherits AWTComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
<i>SelectedText</i>	現在選択されているテキスト。
<i>SelectionEnd</i>	選択されたテキストの終了位置。
<i>SelectionStart</i>	選択されたテキストの開始位置。
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Clear</i>	すべてのテキストをテキスト コンポーネントから削除します
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのテキストを選択します。
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。
SetText	テキスト コンポーネントのテキストを指定したテキストに設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTTextField クラス

説明

ユーザーが変更可能なテキストを保持する、単一行および複数行のテキスト フィールド コントロールを表します。

継承階層

- [AWTTextComponent](#)
 - AWTTextField

構文

```
'Declaration
Public Class AWTTextField _
Inherits AWTTextComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(AWTTextComponent から継承されました)

名前	説明
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
EchoChar	実際に入力される文字の代わりに表示される文字。
EchoCharSet	EchoChar が 0 以外の値に設定されているかどうか
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。 (AWTTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。 (AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
IsPassword	テキスト フィールドがパスワード テキスト フィールドであるかどうか。
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
SelectedText	現在選択されているテキスト。 (AWTTextComponent から継承されました)
SelectionEnd	選択されたテキストの終了位置。 (AWTTextComponent から継承されました)
SelectionStart	選択されたテキストの開始位置。 (AWTTextComponent から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(<code>TestObject</code> から継承されました)
<code>Clear</code>	すべてのテキストをテキスト コンポーネントから削除します (<code>AWTTextComponent</code> から継承されました)
<code>Click</code>	オブジェクトをクリックします。(<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	マウス ボタンをオブジェクト上でダブルクリックします。(<code>IClickable</code> から継承されました)
<code>Exists</code>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(<code>TestObject</code> から継承されました)
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<code>TestObject</code> から継承されました)
<code>Find</code>	XPath ロケーターで指定したオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>FindAll</code>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。(<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。(<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。(<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetRect</code>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<code>TestObject</code> から継承されました)
<code>HighlightObject</code>	このオブジェクトをハイライトします。(<code>TestObject</code> から継承されました)
<code>ImageClick</code>	指定したイメージ資産をクリックします。(<code>TestObject</code> から継承されました)
<code>ImageClickFile</code>	指定したイメージをクリックします。(<code>TestObject</code> から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのテキストを選択します。(AWTTextComponent から継承されました)
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。(AWTTextComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。(AWTTextComponent から継承されました)

名前	説明
SetText	テキスト コンポーネントのテキストを指定したテキストに設定します。(AWTTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTVerticalScrollbar クラス

説明

垂直方向のスクロールバーを表します。

継承階層

- [AWTScrollbar](#)
 - AWTVerticalScrollbar

構文

```
'Declaration
Public Class AWTVerticalScrollbar _
Inherits AWTScrollbar
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーのブロック増分値。(IAWTScroller から継承されました)

名前	説明
<i>Position</i>	スクロールバーの現在の位置。(<i>IAWTScroller</i> から継承されました)
<i>Range</i>	スクロールバーの範囲値。(<i>IAWTScroller</i> から継承されました)
<i>StepSize</i>	スクロールバーの単位増分値。(<i>IAWTScroller</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
Page	指定したページ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

AWTWindow クラス

説明

境界線やメニューバーを持たないトップレベルウィンドウを表します。

継承階層

- [AWTContainer](#)
 - AWTWindow
 - [AWTDialog](#)
 - [AWTFrame](#)
 - [JWindow](#)

構文

```
'Declaration
Public Class AWTWindow _
Inherits AWTContainer _
Implements IMoveable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)
<i>WindowState</i>	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>Close</i>	ウィンドウを閉じます。(IMoveable から継承されました)
<i>CloseSynchron</i>	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 Size 	ウィンドウのサイズを変更します。(IMoveable から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

BasicArrowButton クラス

説明

基本方向のうちの 1 つの矢印を持ったプッシュ ボタンを表します。

継承階層

- [JButton](#)
 - BasicArrowButton

構文

```
'Declaration
Public Class BasicArrowButton _
Inherits JButton
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Direction	矢印が向いている方向。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	ボタンを選択します。(AbstractButton から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

IAWTScrollable インターフェイス

説明

2 つのスクロールバーを持つことができる AWT オブジェクトのインターフェイスを表します。

継承階層

- [IBaseScrollable](#)
 - IAWTScrollable

構文

```
'Declaration
Public Class IAWTScrollable _
Inherits IBaseScrollable
```

プロパティ

名前	説明
Position	スクロール可能コンポーネントの現在のスクロール位置。 (IBaseScrollable から継承されました)

メソッド

名前	説明
GetHorizontalBar	水平方向のスクロールバーを取得します。
GetVerticalBar	垂直方向のスクロールバーを取得します。
ScrollTo	指定した位置にスクロールします。(IBaseScrollable から継承されました)

IAWTScroller インターフェイス

説明

スクロール可能な AWT オブジェクトのインターフェイスを表します。

構文

```
'Declaration  
Public Class IAWTScroller
```

プロパティ

名前	説明
PageSize	スクロールバーのブロック増分値。
Position	スクロールバーの現在の位置。
Range	スクロールバーの範囲値。
StepSize	スクロールバーの単位増分値。

メソッド

名前	説明
Page	指定したページ数を現在のスクロール位置に加算します。
ScrollTo	指定した位置にスクロールします。
Step	指定したステップ数を現在のスクロール位置に加算します。

IBaseScrollable インターフェイス

説明

重複を避けるために他のスクロール可能なコントロールのインターフェイスを表します。

構文

```
'Declaration  
Public Class IBaseScrollable
```

プロパティ

名前	説明
Position	スクロール可能コンポーネントの現在のスクロール位置。

メソッド

名前	説明
ScrollTo	指定した位置にスクロールします。

IOracleFormsMenuBase インターフェイス

説明

ユーザーが選択する選択肢を定義するコントロールを表します。

構文

```
'Declaration  
Public Class IOracleFormsMenuBase
```

プロパティ

名前	説明
Checked	メニュー項目がチェックされているかどうか。

IOracleFormsScrollable インターフェイス

説明

スクロール可能な Oracle Forms オブジェクトのインターフェイスを表します。

構文

```
'Declaration  
Public Class IOracleFormsScrollable
```

プロパティ

名前	説明
HorizontalScrollbarPageSize	スクロールバーのブロック増分値。
HorizontalScrollbarPosition	スクロールバーの現在の位置。
HorizontalScrollbarRange	スクロールバーの範囲値。
HorizontalScrollbarStepSize	スクロールバーの単位増分値。
VerticalScrollbarPageSize	スクロールバーのブロック増分値。
VerticalScrollbarPosition	スクロールバーの現在の位置。
VerticalScrollbarRange	スクロールバーの範囲値。
VerticalScrollbarStepSize	スクロールバーの単位増分値。

メソッド

名前	説明
GetHorizontalScrollbar	水平方向のスクロールバーを取得します。
GetVerticalScrollbar	垂直方向のスクロールバーを取得します。
HorizontalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。
HorizontalScrollbarScrollTo	指定した位置にスクロールします。
HorizontalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。
HorizontalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。
HorizontalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。
VerticalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。
VerticalScrollbarScrollTo	指定した位置にスクロールします。
VerticalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。
VerticalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。
VerticalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。

IOracleFormsScroller インターフェイス

説明

スクロール可能な Oracle Forms オブジェクトのインターフェイスを表します。

継承階層

- [IAWTScroller](#)
 - IOracleFormsScroller

構文

```
'Declaration  
Public Class IOracleFormsScroller _  
Inherits IAWTScroller
```

プロパティ

名前	説明
PageSize	スクロールバーのブロック増分値。(IAWTScroller から継承されました)
Position	スクロールバーの現在の位置。(IAWTScroller から継承されました)
Range	スクロールバーの範囲値。(IAWTScroller から継承されました)
StepSize	スクロールバーの単位増分値。(IAWTScroller から継承されました)

メソッド

名前	説明
Page	指定したページ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScroller から継承されました)
ScrollToMax	スクロールバーの最大の位置にスクロールします。
ScrollToMin	スクロールバーの最小の位置にスクロールします。
Step	指定したステップ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)

JButton クラス

説明

プッシュ ボタンを表します。

継承階層

- [AbstractButton](#)

- JButton
 - [BasicArrowButton](#)

構文

```
'Declaration
Public Class JButton _
Inherits AbstractButton
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
<code>Click</code>	オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)</p>
<code>Exists</code>	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)</p>
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)
<code>Find</code>	<p>XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)</p>
<code>FindAll</code>	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)</p>
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。 (<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (<code>TestObject</code> から継承されました)
<code>GetRect</code>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (<code>TestObject</code> から継承されました)
<code>HighlightObject</code>	このオブジェクトをハイライトします。 (<code>TestObject</code> から継承されました)
<code>ImageClick</code>	指定したイメージ資産をクリックします。 (<code>TestObject</code> から継承されました)

名前	説明
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキスト メニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>Select</i>	ボタンを選択します。(<i>AbstractButton</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JCheckBox クラス

説明

チェック ボックスを表します。

継承階層

- [JToggleButton](#)
 - JCheckBox

構文

```
'Declaration  
Public Class JCheckBox _  
Inherits JToggleButton
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Selected	トグル ボタンが選択されているかどうか。(JToggleButton から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Check	チェック ボックスをチェックします。
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	ボタンを選択します。(AbstractButton から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Toggle	ボタンの状態を切り替えます。(JToggleButton から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	チェック ボックスのチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オ

名前	説明
	プシオン OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JCheckBoxMenuItem クラス

説明

メニュー項目に含めることができるチェック ボックス コントロールを表します。

継承階層

- JMenuItem
 - JCheckBoxMenuItem

構文

```
'Declaration
Public Class JCheckBoxMenuItem _
Inherits JMenuItem
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	メニュー項目のチェック ボックスがチェックされているかどうか。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	ボタンを選択します。(AbstractButton から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキストフィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JColorChooser クラス

説明

色の選択を表します。

継承階層

- [JComponent](#)
 - JColorChooser

構文

```
'Declaration
Public Class JColorChooser _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Color	カラーチューザから現在の色の値。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)

名前	説明
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetColor	カラーチューザの現在の色を指定した色に設定します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JComboBox クラス

説明

コンボ ボックスを表します。

継承階層

- [JComponent](#)
 - JComboBox

構文

```
'Declaration
Public Class JComboBox _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	コンボ ボックスが編集可能かどうか。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)

名前	説明
<i>Font</i>	コンポーネントのフォント。(<i>AWTComponent</i> から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(<i>AWTComponent</i> から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(<i>AWTComponent</i> から継承されました)
<i>ItemCount</i>	リスト項目の数。
<i>Items</i>	すべての含まれる項目のリスト。
<i>Name</i>	コンポーネントの名前。(<i>AWTComponent</i> から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (<i>INativeWindow</i> から継承されました)
<i>SelectedIndex</i>	現在選択されている項目のインデックス。
<i>SelectedItem</i>	現在選択されている項目。
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>ToolTipText</i>	Swing コンポーネントのツールチップ テキスト。(<i>JComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetItem	指定したインデックスの項目を返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>Select</i>	指定した項目を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JComponent クラス

説明

トップレベル コンテナーを除くすべてのコンポーネントのクラスを表します。

継承階層

- [AWTContainer](#)
 - JComponent
 - [AbstractButton](#)
 - [JColorChooser](#)
 - [JComboBox](#)
 - [JLabel](#)
 - [JLayeredPane](#)
 - [JList](#)
 - [JMenuBar](#)
 - [JPanel](#)
 - [JPopupMenu](#)
 - [JProgressBar](#)

- [JRootPane](#)
- [JScrollBar](#)
- [JScrollPane](#)
- [JSlider](#)
- [JSpinner](#)
- [JSplitPane](#)
- [JTabbedPane](#)
- [JTable](#)
- [JTableHeader](#)
- [JTextComponent](#)
- [JToolBar](#)
- [JTree](#)
- [JViewport](#)

構文

```
'Declaration
Public Class JComponent _
Inherits AWTContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JDesktopPane クラス

説明

デスクトップ ペインを表します。

継承階層

- [JLayeredPane](#)
 - JDesktopPane

構文

```
'Declaration
Public Class JDesktopPane _
Inherits JLayeredPane
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキストフィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JDialog クラス

説明

ダイアログ ボックスを表します。

継承階層

- [AWTDialog](#)
 - JDialog

構文

```
'Declaration
Public Class JDialog _
Inherits AWTDialog
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)

名前	説明
Modal	ダイアログがモーダルかどうか。(AWTDialog から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Resizable	ユーザーがダイアログのサイズを変更できるかどうか。(AWTDialog から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Title	ダイアログのタイトル。(AWTDialog から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsActive</i>	ウィンドウがアクティブに設定されているかどうかを返します。 (<i>IMoveable</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>Maximize</i>	ウィンドウを最大化します。(<i>IMoveable</i> から継承されました)
<i>Minimize</i>	ウィンドウを最小化します。(<i>IMoveable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	ウィンドウを移動します。(<i>IMoveable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Restore</i>	ウィンドウを元のサイズに戻します。(<i>IMoveable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>SetActive</i>	ウィンドウをアクティブにします。(<i>IMoveable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>Size</i>	ウィンドウのサイズを変更します。(<i>IMoveable</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JEditorPane クラス

説明

さまざまな種類のコンテンツを編集できるテキスト コンポーネントを表します。

継承階層

- [JTextComponent](#)
 - JEditorPane
 - [JTextPane](#)

構文

```
'Declaration  
Public Class JEditorPane _  
Inherits JTextComponent _
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(JTextComponent から継承されました)
ContentType	このエディターが現在処理するように設定されているコンテンツ タイプ。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。(JTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
FormattedText	エディター ペインのテキスト。
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Page	表示されている現在の URL。
SelectedText	現在選択されているテキスト。(JTextComponent から継承されました)

名前	説明
SelectionEnd	選択範囲の終了位置。 (JTextComponent から継承されました)
SelectionStart	選択範囲の開始位置。 (JTextComponent から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。 (JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをテキスト コンポーネントから削除します (JTextComponent から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのコンテンツを選択します。(JTextComponent から継承されました)
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。(JTextComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。(JTextComponent から継承されました)
SetText	テキスト コンテンツを指定したテキストに設定します。(JTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JFrame クラス

説明

Java Swing のサポートを追加する AWTFrame を拡張したバージョンのクラスを表します。

継承階層

- [AWTFrame](#)
 - JFrame

構文

```
'Declaration
Public Class JFrame _
Inherits AWTFrame
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)

名前	説明
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Resizable	ユーザーがフレームのサイズを変更できるかどうか。(AWTFrame から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Title	フレームのタイトル。(AWTFrame から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 Size 	ウィンドウのサイズを変更します。(IMoveable から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JHorizontalScrollBar クラス

説明

水平方向のスクロールバーを表します。

継承階層

- [JScrollBar](#)
 - JHorizontalScrollBar

構文

```
'Declaration
Public Class JHorizontalScrollBar _
Inherits JScrollBar
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
PageSize	スクロールバーのブロック増分値。(IAWTScroller から継承されました)
Position	スクロールバーの現在の位置。(IAWTScroller から継承されました)
Range	スクロールバーの範囲値。(IAWTScroller から継承されました)
StepSize	スクロールバーの単位増分値。(IAWTScroller から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)

名前	説明
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)

名前	説明
Page	指定したページ数を現在のスクロール位置に加算します。 (IAWTScroller から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。 (IAWTScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。 (IAWTScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JLabel クラス

説明

簡単なテキスト、画像、または両方を表示する表示領域を表します。

継承階層

- [JComponent](#)
 - JLabel

構文

```
'Declaration
Public Class JLabel _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
Alignment	ラベルの配置。
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JLayeredPane クラス

説明

階層ペインを表します。

継承階層

- [JComponent](#)
 - JLayeredPane
 - [JDesktopPane](#)

構文

```
'Declaration
Public Class JLayeredPane _
Inherits JComponent
```


プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JList クラス

説明

リスト ボックスを表します。

継承階層

- [JComponent](#)
 - JList

構文

```
'Declaration
Public Class JList _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
ItemCount	リスト項目の数。
Items	すべての含まれるリスト項目のリスト。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択されたリスト項目のインデックス。
SelectedIndices	選択されたリスト項目のインデックスのリスト。
SelectedItem	選択したリスト項目の名前またはインデックス。
SelectedItems	選択されたリスト項目の名前またはインデックスのリスト。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
Click	オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)
ClickItem	指定したリスト項目をクリックします。
Deselect	指定したリスト項目の選択を解除します。
DeselectAll	すべてのリスト項目の選択を解除します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)
DoubleClickItem	リスト項目をダブルクリックします。
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)</p>
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
FindItem	リスト内の項目のインデックスを返します。
GenerateLocator	このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (<code>TestObject</code> から継承されました)
GetItem	指定したインデックスの項目を返します。
GetItemRect	リストに対して相対的な項目のサイズと位置を返します。
GetItemValue	リスト項目の値オブジェクトを返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<code>TestObject</code> から継承されました)
GetProperty	指定したプロパティの値を返します。 (<code>TestObject</code> から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)

名前	説明
ScrollItemIntoView	項目をビューにスクロールします。
Select	リスト項目を選択します。
SelectAll	すべてのリスト要素を選択します。
SelectRange	リスト要素の指定した範囲を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JMenu クラス

説明

メニューを表します。メニューとは子項目を持つメニュー項目です。

継承階層

- [JMenuItem](#)
 - JMenu

構文

```
'Declaration
Public Class JMenu _
Inherits JMenuItem
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)

名前	説明
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	ボタンを選択します。(AbstractButton から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	タイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JMenuBar クラス

説明

メニューバーを表します。

継承階層

- [JComponent](#)
 - JMenuBar

構文

```
'Declaration
Public Class JMenuBar _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)

名前	説明
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。 (JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
	<p>ウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JMenuItem クラス

説明

メニュー項目を表します。

継承階層

- [AbstractButton](#)
 - JMenuItem
 - [JCheckBoxMenuItem](#)
 - [JMenu](#)
 - [JRadioButtonMenuItem](#)

構文

```
'Declaration
Public Class JMenuItem _
Inherits AbstractButton
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)

名前	説明
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>Select</i>	ボタンを選択します。 (<i>AbstractButton</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JPanel クラス

説明

軽量コンポーネントのコンテナ クラスを表します。

継承階層

- [JComponent](#)
 - JPanel

構文

```
'Declaration
Public Class JPanel _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JPasswordField クラス

説明

元の文字を表示しない単一行フィールドを表します。

継承階層

- [JTextField](#)
 - JPasswordField

構文

```
'Declaration
Public Class JPasswordField _
Inherits JTextField
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Alignment	テキスト フィールドの配置。(JTextField から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(JTextComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
EchoChar	実際に入力される文字の代わりに表示される文字。
EchoCharSet	EchoChar が 0 以外の値に設定されているかどうか
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。(JTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。(INativeWindow から継承されました)
SelectedText	現在選択されているテキスト。(JTextComponent から継承されました)
SelectionEnd	選択範囲の終了位置。(JTextComponent から継承されました)
SelectionStart	選択範囲の開始位置。(JTextComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをテキスト コンポーネントから削除します (JTextComponent から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのコンテンツを選択します。(JTextComponent から継承されました)
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。(JTextComponent から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。(JTextComponent から継承されました)
SetText	テキスト コンテンツを指定したテキストに設定します。(JTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JPopupMenu クラス

説明

コンテキストメニューを表します。

継承階層

- [JComponent](#)
 - JPopupMenu

構文

```
'Declaration
Public Class JPopupMenu _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)

名前	説明
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForProperty	<p>トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JProgressBar クラス

説明

操作の進行状況を表示します。

継承階層

- [JComponent](#)
 - JProgressBar

構文

```
'Declaration
Public Class JProgressBar _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)

名前	説明
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Indeterminate	進行状況バーが不確定モードかどうか。
Maximum	進行状況バーの最大値。
Minimum	進行状況バーの最小値。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
ProgressString	現在の進行状況の文字列表現。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
	<p>アウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JRadioButton クラス

説明

ラジオ ボタンを表します。

継承階層

- [JToggleButton](#)
 - JRadioButton

構文

```
'Declaration
Public Class JRadioButton _
Inherits JToggleButton
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Selected	トグル ボタンが選択されているかどうか。(JToggleButton から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	ラジオ ボタンをチェックします。
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
Select	ボタンを選択します。 (AbstractButton から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Toggle	ボタンの状態を切り替えます。 (JToggleButton から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JRadioButtonMenuItem クラス

説明

ラジオ ボタン メニュー項目を表します。

継承階層

- [JMenuItem](#)
 - JRadioButtonMenuItem

構文

```
'Declaration  
Public Class JRadioButtonMenuItem _  
Inherits JMenuItem
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	メニュー項目のラジオ ボタンがチェックされているかどうか。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
<code>Click</code>	オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)</p>
<code>Exists</code>	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)</p>
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)
<code>Find</code>	<p>XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)</p>
<code>FindAll</code>	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)</p>
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。 (<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (<code>TestObject</code> から継承されました)
<code>GetRect</code>	このオブジェクトの大きさや位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (<code>TestObject</code> から継承されました)
<code>HighlightObject</code>	このオブジェクトをハイライトします。 (<code>TestObject</code> から継承されました)
<code>ImageClick</code>	指定したイメージ資産をクリックします。 (<code>TestObject</code> から継承されました)
<code>ImageClickFile</code>	指定したイメージをクリックします。 (<code>TestObject</code> から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	ボタンを選択します。(AbstractButton から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JRootPane クラス

説明

ルート ペインを表します。

継承階層

- [JComponent](#)
 - JRootPane

構文

```
'Declaration  
Public Class JRootPane _  
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JScrollBar クラス

説明

スクロールバーを表します。

継承階層

- [JComponent](#)
 - JScrollBar
 - [JHorizontalScrollBar](#)
 - [JVerticalScrollBar](#)

構文

```
'Declaration
Public Class JScrollBar _
Inherits JComponent _
Implements IAWTScroller
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーのブロック増分値。(IAWTScroller から継承されました)

名前	説明
<i>Position</i>	スクロールバーの現在の位置。(<i>IAWTScroller</i> から継承されました)
<i>Range</i>	スクロールバーの範囲値。(<i>IAWTScroller</i> から継承されました)
<i>StepSize</i>	スクロールバーの単位増分値。(<i>IAWTScroller</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>ToolTipText</i>	Swing コンポーネントのツールチップ テキスト。(<i>JComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
Page	指定したページ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JScrollPane クラス

説明

ペイン自身よりも大きくなるコンテンツをスクロールすることができるスクロール ペインを表します。

継承階層

- [JComponent](#)
 - JScrollPane

構文

```
'Declaration
Public Class JScrollPane _
Inherits JComponent _
Implements IBaseScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)

名前	説明
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Position	スクロール可能コンポーネントの現在のスクロール位置。 (IBaseScrollable から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	水平方向のスクロールバーを取得します。
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	垂直方向のスクロールバーを取得します。
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。 (IBaseScrollable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JSlider クラス

説明

範囲内の値を選択できるスライダ コントロールを表します。

継承階層

- [JComponent](#)
 - JSlider

構文

```
'Declaration
Public Class JSlider _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Inverted	スライダーに表示される値の範囲が反転している場合、値は true になります。
Maximum	スライダーによってサポートされる最大値。
Minimum	スライダーによってサポートされる最小値。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Position	スライダーの位置。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetPosition	スライダーの現在の位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JSpinner クラス

説明

シーケンスの値を反復処理できるスピナー コントロールを表します。

継承階層

- [JComponent](#)
 - JSpinner

構文

```
'Declaration  
Public Class JSpinner _  
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Next</i>	Value プロパティによって返されるオブジェクトの次にくるシーケンス内のオブジェクトにスピナーを設定します。
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Previous</i>	Value プロパティによって返されるオブジェクトの前にあるシーケンス内のオブジェクトにスピナーを設定します。
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JSplitPane クラス

説明

水平方向または垂直方向に 2 つのコンポーネントに分割するスプリット ペイン コントロールを表します。

継承階層

- [JComponent](#)
 - JSplitPane

構文

```
'Declaration
Public Class JSplitPane _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
DividerLocation	現在の分割線の位置。
DividerSize	分割線のサイズ。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
MaximumDividerLocation	分割線の最大位置。
MinimumDividerLocation	分割線の最小位置。

名前	説明
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetPosition	分割線の位置を設定します。
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JTabbedPane クラス

説明

タブ付きペインを表します。

継承階層

- [JComponent](#)
 - JTabbedPane

構文

```
'Declaration
Public Class JTabbedPane _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)

名前	説明
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedTab	現在選択されているタブ。
SelectedTabIndex	現在選択されているタブのインデックス。タブが選択されていない場合は -1 (4Test の場合は 0)。
TabCount	タブの数。
TabRunCount	タブを表示するために現在使用されているタブの並びの数。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)

名前	説明
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetTabIndex	タブ インデックスを返します。
GetTabTitle	タブ タイトルを返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)

名前	説明
<i>IsTabEnabled</i>	タブが現在有効かどうかを返します。
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>Select</i>	指定したタブを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。(<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JTable クラス

説明

行と列を持つテーブル コントロールを表します。

継承階層

- [JComponent](#)
 - JTable

構文

```
'Declaration
Public Class JTable _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
AllCellTexts	テーブルのすべてのセルのテキスト値。セルは行でグループ化され、各行にはそのセルのテキストが含まれます。
AllCellValues	テーブルのすべてのセルの値。セルは行でグループ化され、各行にはそのセルの値が含まれます。
Background	コンポーネントの背景色。(AWTComponent から継承されました)
ColumnCount	テーブルの列の数。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
EditingColumn	現在編集集中のセルを含む列の名前またはインデックス。
EditingRow	現在編集集中のセルを含む行のインデックス。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
RowCount	テーブルに表示することのできる行の数。
SelectedColumn	選択した列のインデックスまたは名前。列が選択されていない場合には -1 (4Test の場合は 0) が返されます。
SelectedColumns	すべての選択されている列のインデックスのリスト。
SelectedRow	選択した行のインデックス。行が選択されていない場合は -1 (4Test の場合は 0) が返されます。
SelectedRows	すべての選択されている行のインデックスのリスト。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
ClearSelection	すべての選択した列と行の選択を解除します。
Click	オブジェクトをクリックします。 (IClickable から継承されました)
ClickCell	テーブル内の指定したセルをクリックします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClickCell	テーブル内の指定したセルをダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
FindCell	指定したテキストを含むセルを返します。
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetCellText	セル テキストを返します。
GetCellValue	セルの値を返します。
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetColumnName	指定したインデックスの列の名前を返します。
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsCellEditable	指定したセルが編集可能な場合に true を返します。
IsCellSelected	指定したセルが選択されている場合に true、それ以外は false を返します。
IsColumnSelected	指定した列が選択されている場合に true を返します。
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
IsRowSelected	指定した行が選択されている場合に true を返します。
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MoveColumn	ターゲット列の位置に列を移動します。
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ResizeColumn	指定した列のサイズを変更します。
ScrollCellIntoView	指定したセルをビューにスクロールします。
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テーブル内のすべての行、列、およびセルを選択します。
SelectColumn	指定した列を選択します。
SelectRow	指定した行を選択します。
SetCellValue	テーブル内の指定したセルに指定した値を設定します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
WaitForObject	<p>タイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JTableHeader クラス

説明

テーブル コンポーネントのテーブル ヘッダー クラスを表します。

継承階層

- [JComponent](#)
 - JTableHeader

構文

```
'Declaration
Public Class JTableHeader _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>ToolTipText</i>	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>ClickColumnHeader</i>	テーブルの列ヘッダーをクリックします。
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JTextArea クラス

説明

書式なしテキストを表示する複数行テキスト領域コントロールを表します。

継承階層

- [JTextComponent](#)
 - JTextArea

構文

```
'Declaration
Public Class JTextArea _
Inherits JTextComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(JTextComponent から継承されました)
Columns	テキスト領域の列の数。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。(JTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Rows	テキスト領域の行の数。
SelectedText	現在選択されているテキスト。(JTextComponent から継承されました)
SelectionEnd	選択範囲の終了位置。(JTextComponent から継承されました)
SelectionStart	選択範囲の開始位置。(JTextComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをテキスト コンポーネントから削除します (JTextComponent から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Insert	指定したテキストを指定した位置に挿入します。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReplaceRange	指定した範囲内のテキストを指定した新しいテキストで置き換えます。
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのコンテンツを選択します。(JTextComponent から継承されました)

名前	説明
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。(JTextComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。(JTextComponent から継承されました)
SetText	テキスト コンテンツを指定したテキストに設定します。(JTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JTextComponent クラス

説明

グラフィカルに表現される属性でマークすることができるテキスト コントロールを表します。

継承階層

- [JComponent](#)
 - JTextComponent
 - [JEditorPane](#)
 - [JTextArea](#)
 - [JTextField](#)

構文

```
'Declaration
Public Class JTextComponent _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)

名前	説明
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。 (<i>AWTComponent</i> から継承されました)
<i>Font</i>	コンポーネントのフォント。 (<i>AWTComponent</i> から継承されました)
<i>Foreground</i>	コンポーネントの前景色。 (<i>AWTComponent</i> から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。 (<i>AWTComponent</i> から継承されました)
<i>Name</i>	コンポーネントの名前。 (<i>AWTComponent</i> から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (<i>INativeWindow</i> から継承されました)
<i>SelectedText</i>	現在選択されているテキスト。
<i>SelectionEnd</i>	選択範囲の終了位置。
<i>SelectionStart</i>	選択範囲の開始位置。
<i>Text</i>	コントロールのテキスト。 (<i>TestObject</i> から継承されました)
<i>ToolTipText</i>	Swing コンポーネントのツールチップ テキスト。 (<i>JComponent</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Width</i>	現在のコンポーネントの幅。 (<i>AWTComponent</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (<i>TestObject</i> から継承されました)
<i>Clear</i>	すべてのテキストをテキスト コンポーネントから削除します
<i>Click</i>	オブジェクトをクリックします。 (<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SelectAll</i>	テキスト コンポーネント内のすべてのコンテンツを選択します。
<i>SetCaretPosition</i>	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>SetSelectionRange</i>	指定した範囲のテキストを選択します。
<i>SetText</i>	テキスト コンテンツを指定したテキストに設定します。
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JTextField クラス

説明

ユーザーが変更可能なテキストを保持する、単一行のテキスト フィールド コントロールを表します。

継承階層

- [JTextComponent](#)
 - JTextField
 - [JPasswordField](#)

構文

```
'Declaration
Public Class JTextField _
Inherits JTextComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Alignment	テキスト フィールドの配置。
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(JTextComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。(JTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedText	現在選択されているテキスト。(JTextComponent から継承されました)
SelectionEnd	選択範囲の終了位置。(JTextComponent から継承されました)
SelectionStart	選択範囲の開始位置。(JTextComponent から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをテキスト コンポーネントから削除します (JTextComponent から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのコンテンツを選択します。(JTextComponent から継承されました)
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。(JTextComponent から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。(JTextComponent から継承されました)
SetText	テキスト コンテンツを指定したテキストに設定します。(JTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JTextPane クラス

説明

グラフィカルに表現される属性でマークすることができるテキスト ペイン コントロールを表します。

継承階層

- [JEditorPane](#)
 - JTextPane

構文

```
'Declaration
Public Class JTextPane _
Inherits JEditorPane
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置。(JTextComponent から継承されました)
CharacterAttributes	キャレットの位置で有効になっている文字属性。
ContentType	このエディターが現在処理するように設定されているコンテンツ タイプ。(JEditorPane から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト コンポーネントのテキストをユーザーが変更できるかどうか。(JTextComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)

名前	説明
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
FormattedText	エディター ペインのテキスト。 (JEditorPane から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
LogicalStyle	キャレットの位置で有効になっている現在の段落に割り当てられている論理書式。
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Page	表示されている現在の URL。 (JEditorPane から継承されました)
ParagraphAttributes	キャレットの位置で有効になっている現在の段落の属性。
SelectedText	現在選択されているテキスト。 (JTextComponent から継承されました)
SelectionEnd	選択範囲の終了位置。 (JTextComponent から継承されました)
SelectionStart	選択範囲の開始位置。 (JTextComponent から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。 (JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Clear	すべてのテキストをテキスト コンポーネントから削除します (JTextComponent から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SelectAll	テキスト コンポーネント内のすべてのコンテンツを選択します。 (JTextComponent から継承されました)
SetCaretPosition	テキスト コンポーネントのテキスト挿入キャレットの位置を設定します。(JTextComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetSelectionRange	指定した範囲のテキストを選択します。(JTextComponent から継承されました)
SetText	テキスト コンテンツを指定したテキストに設定します。 (JTextComponent から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)

名前	説明
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JToggleButton クラス

説明

2 つの状態を持つボタンを表します。

継承階層

- [AbstractButton](#)
 - JToggleButton
 - [JCheckBox](#)
 - [JRadioButton](#)

構文

```
'Declaration  
Public Class JToggleButton _  
Inherits AbstractButton
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Selected	トグル ボタンが選択されているかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>Select</i>	ボタンを選択します。(<i>AbstractButton</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Toggle	ボタンの状態を切り替えます。
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JToolBar クラス

説明

ツールバーを表します。

継承階層

- [JComponent](#)
 - JToolBar

構文

```
'Declaration  
Public Class JToolBar _  
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)

名前	説明
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイ

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JTree クラス

説明

ツリーを表します。

継承階層

- [JComponent](#)
 - JTree

構文

```
'Declaration
Public Class JTree _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
IsEditable	ツリーが編集可能かどうか。
IsEditing	ツリーが編集中かどうか。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
SelectedItemPath	選択された項目パス。
SelectedItemPaths	選択された項目パスのリスト。
SelectedItemText	選択された項目のテキスト。
SelectionCount	選択されたツリー ノードの数。
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VisibleItemPaths	ツリーに表示される項目の項目パス。
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CancelEditing	現在編集集中のセッションをキャンセルします。
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ClearSelection	選択範囲をクリアします。
Click	オブジェクトをクリックします。(IClickable から継承されました)
ClickNode	項目パスによって指定されたツリー ノードをクリックします。
Collapse	指定した項目パスによって識別されるツリー ノードを折りたたみます。
CollapseAll	すべてのツリー ノードを折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClickNode	項目パスによって指定されたツリー ノードをダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Expand	指定した項目パスによって識別されるツリー ノードを展開します。
ExpandAll	すべてのツリー ノードを展開します。
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetItemRect	ツリーに対して相対的な項目のサイズと位置を返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsExpanded	指定した項目パスが展開されている場合に 'true' を返します。
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
IsPathEditable	項目パスが編集可能かどうかを取得します。
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
ScrollPathIntoView	指定した項目パスを表示できる位置にスクロールします。
Select	指定した項目パスによって識別されるツリー ノードを選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
StartEditing	指定した項目パスまたは現在の選択項目の編集を開始します。
StopEditing	現在編集集中のセッションを停止します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JVerticalScrollBar クラス

説明

垂直方向のスクロールバーを表します。

継承階層

- [JScrollBar](#)
 - JVerticalScrollBar

構文

```
'Declaration  
Public Class JVerticalScrollBar _  
Inherits JScrollBar
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーのブロック増分値。(IAWTScroller から継承されました)
Position	スクロールバーの現在の位置。(IAWTScroller から継承されました)
Range	スクロールバーの範囲値。(IAWTScroller から継承されました)
StepSize	スクロールバーの単位増分値。(IAWTScroller から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
Page	指定したページ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IAWTScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 Step 	指定したステップ数を現在のスクロール位置に加算します。(IAWTScroller から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

JViewport クラス

説明

Swing ビューポートを表します。

継承階層

- [JComponent](#)
 - JViewport

構文

```
'Declaration
Public Class JViewport _
Inherits JComponent
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
ToolTipText	Swing コンポーネントのツールチップ テキスト。(JComponent から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
ViewRectangle	ビューの可視部分をビューの座標。
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
getParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

JWindow クラス

説明

ユーザーのデスクトップ上のどこにでも表示できるコンテナを表します。

継承階層

- [AWTWindow](#)
 - JWindow

構文

```
'Declaration
Public Class JWindow _
Inherits AWTWindow _
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)

名前	説明
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。(IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。(IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>Maximize</i>	ウィンドウを最大化します。(<i>IMoveable</i> から継承されました)
<i>Minimize</i>	ウィンドウを最小化します。(<i>IMoveable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	ウィンドウを移動します。(<i>IMoveable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Restore</i>	ウィンドウを元のサイズに戻します。(<i>IMoveable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>SetActive</i>	ウィンドウをアクティブにします。(<i>IMoveable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>Size</i>	ウィンドウのサイズを変更します。(<i>IMoveable</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。(<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。(<i>IKeyable</i> から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsApplication クラス

説明

Oracle Forms アプリケーションを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsApplication

構文

```
'Declaration  
Public Class OracleFormsApplication _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SplashScreenRunning	アプリケーションが現在スプラッシュ スクリーンを表示するかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル)

名前	説明
	はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

OracleFormsButton クラス

説明

プッシュ ボタンを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsButton

構文

```
'Declaration  
Public Class OracleFormsButton _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Default	ボタンがデフォルト ボタンかどうか。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(<code>TestObject</code> から継承されました)
<code>Click</code>	オブジェクトをクリックします。(<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	マウス ボタンをオブジェクト上でダブルクリックします。(<code>IClickable</code> から継承されました)
<code>Exists</code>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(<code>TestObject</code> から継承されました)
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<code>TestObject</code> から継承されました)
<code>Find</code>	XPath ロケーターで指定したオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>FindAll</code>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。(<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。(<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。(<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetRect</code>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<code>TestObject</code> から継承されました)
<code>HighlightObject</code>	このオブジェクトをハイライトします。(<code>TestObject</code> から継承されました)
<code>ImageClick</code>	指定したイメージ資産をクリックします。(<code>TestObject</code> から継承されました)
<code>ImageClickFile</code>	指定したイメージをクリックします。(<code>TestObject</code> から継承されました)
<code>ImageExists</code>	指定したイメージが存在するかどうかを返します。(<code>TestObject</code> から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	ボタンを選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

OracleFormsCheckbox クラス

説明

チェック ボックスを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsCheckbox

構文

```
'Declaration  
Public Class OracleFormsCheckbox _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	チェック ボックスがチェックされているかどうか。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(<code>TestObject</code> から継承されました)
<code>Check</code>	チェック ボックスをチェックします。
<code>Click</code>	オブジェクトをクリックします。(<code>IClickable</code> から継承されました)
<code>DoubleClick</code>	マウス ボタンをオブジェクト上でダブルクリックします。(<code>IClickable</code> から継承されました)
<code>Exists</code>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(<code>TestObject</code> から継承されました)
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<code>TestObject</code> から継承されました)
<code>Find</code>	XPath ロケーターで指定したオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>FindAll</code>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<code>TestObject</code> から継承されました)
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。(<code>TestObject</code> から継承されました)
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。(<code>TestObject</code> から継承されました)
<code>GetDynamicMethodList</code>	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetParent</code>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<code>TestObject</code> から継承されました)
<code>GetProperty</code>	指定したプロパティの値を返します。(<code>TestObject</code> から継承されました)
<code>GetPropertyList</code>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<code>TestObject</code> から継承されました)
<code>GetRect</code>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<code>TestObject</code> から継承されました)
<code>HighlightObject</code>	このオブジェクトをハイライトします。(<code>TestObject</code> から継承されました)
<code>ImageClick</code>	指定したイメージ資産をクリックします。(<code>TestObject</code> から継承されました)
<code>ImageClickFile</code>	指定したイメージをクリックします。(<code>TestObject</code> から継承されました)
<code>ImageExists</code>	指定したイメージが存在するかどうかを返します。(<code>TestObject</code> から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Uncheck	チェック ボックスのチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

OracleFormsChoice クラス

説明

選択肢を表します。

継承階層

- [OracleFormsContainer](#)

- OracleFormsChoice
 - [OracleFormsComboBox](#)
 - [OracleFormsListBox](#)
 - [OracleFormsPopList](#)

構文

```
'Declaration
Public Class OracleFormsChoice _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
ItemCount	選択肢の項目の数。
Items	選択肢項目のリスト。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択されている項目のインデックス。項目が現在選択されていない場合には -1 (4Test の場合は 0) を返します。
SelectedItem	選択されている項目。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetItemText	指定したインデックスの項目を返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	選択肢で指定した項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsComboBox クラス

説明

コンボ ボックスを表します。

継承階層

- [OracleFormsChoice](#)
 - OracleFormsComboBox

構文

```
'Declaration  
Public Class OracleFormsComboBox _  
Inherits OracleFormsChoice
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
DropDownOpen	ドロップダウンが現在開いているかどうか。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
ItemCount	選択肢の項目の数。(OracleFormsChoice から継承されました)
Items	選択肢項目のリスト。(OracleFormsChoice から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
SelectedIndex	選択されている項目のインデックス。項目が現在選択されていない場合には -1 (4Test の場合は 0) を返します。 (OracleFormsChoice から継承されました)
SelectedItem	選択されている項目。(OracleFormsChoice から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetItemText	指定したインデックスの項目を返します。(OracleFormsChoice から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	選択肢で指定した項目を選択します。(OracleFormsChoice から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetText	コンボ ボックスのテキストを指定したテキストに設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForProperty	<p>値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

OracleFormsContainer クラス

説明

他の Oracle Forms コンポーネントを含めることができるコンポーネントを表します。

継承階層

- [AWTContainer](#)
 - OracleFormsContainer
 - [OracleFormsApplication](#)
 - [OracleFormsButton](#)
 - [OracleFormsCheckbox](#)
 - [OracleFormsChoice](#)
 - [OracleFormsLabel](#)
 - [OracleFormsListView](#)
 - [OracleFormsMenu](#)
 - [OracleFormsMenuItem](#)
 - [OracleFormsRadioButton](#)
 - [OracleFormsScrollbar](#)
 - [OracleFormsStatusArea](#)
 - [OracleFormsStatusBar](#)
 - [OracleFormsStatusBarItem](#)
 - [OracleFormsStatusIndicator](#)
 - [OracleFormsTabBar](#)
 - [OracleFormsTabBarItem](#)
 - [OracleFormsTabPanel](#)
 - [OracleFormsTextField](#)
 - [OracleFormsTitleBar](#)
 - [OracleFormsToolBar](#)
 - [OracleFormsToolBarItem](#)
 - [OracleFormsTree](#)

構文

```
'Declaration  
Public Class OracleFormsContainer _  
Inherits AWTContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsHorizontalScrollbar クラス

説明

水平方向のスクロールバーを表します。

継承階層

- [OracleFormsScrollbar](#)
 - OracleFormsHorizontalScrollbar

構文

```
'Declaration  
Public Class OracleFormsHorizontalScrollbar _  
Inherits OracleFormsScrollbar
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーのブロック増分値。(IOracleFormsScroller から継承されました)
Position	スクロールバーの現在の位置。(IOracleFormsScroller から継承されました)
Range	スクロールバーの範囲値。(IOracleFormsScroller から継承されました)
StepSize	スクロールバーの単位増分値。(IOracleFormsScroller から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>Page</i>	指定したページ数を現在のスクロール位置に加算します。(<i>IOracleFormsScroller</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>ScrollTo</i>	指定した位置にスクロールします。(<i>IOracleFormsScroller</i> から継承されました)
<i>ScrollToMax</i>	スクロール バーの最大の位置にスクロールします。(<i>IOracleFormsScroller</i> から継承されました)

名前	説明
ScrollToMin	スクロールバーの最小の位置にスクロールします。 (IOracleFormsScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。 (IOracleFormsScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

OracleFormsLabel クラス

説明

コンテナ内に配置されるテキストのクラスを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsLabel

構文

```
'Declaration
Public Class OracleFormsLabel _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)

名前	説明
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
getParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

OracleFormsListBox クラス

説明

ListBox を表します。

継承階層

- [OracleFormsChoice](#)
 - OracleFormsListBox

構文

```
'Declaration
Public Class OracleFormsListBox _
Inherits OracleFormsChoice _
Implements IOracleFormsScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)

名前	説明
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
HorizontalScrollbarPageSize	スクロールバーのブロック増分値。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarPosition	スクロールバーの現在の位置。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarRange	スクロールバーの範囲値。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarStepSize	スクロールバーの単位増分値。(IOracleFormsScrollable から継承されました)
ItemCount	選択肢の項目の数。(OracleFormsChoice から継承されました)
Items	選択肢項目のリスト。(OracleFormsChoice から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択されている項目のインデックス。項目が現在選択されていない場合には -1 (4Test の場合は 0) を返します。(OracleFormsChoice から継承されました)
SelectedItem	選択されている項目。(OracleFormsChoice から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VerticalScrollbarPageSize	スクロールバーのブロック増分値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarPosition	スクロールバーの現在の位置。(IOracleFormsScrollable から継承されました)
VerticalScrollbarRange	スクロールバーの範囲値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStepSize	スクロールバーの単位増分値。(IOracleFormsScrollable から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。.NET

名前	説明
	では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
ClickItem	指定したリスト項目をクリックします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClickItem	リスト項目をダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalScrollbar	水平方向のスクロールバーを取得します。(IOracleFormsScrollable から継承されました)
GetItemText	指定したインデックスの項目を返します。(OracleFormsChoice から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalScrollbar	垂直方向のスクロールバーを取得します。(IOracleFormsScrollable から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
HorizontalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollTo	指定した位置にスクロールします。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	選択肢で指定した項目を選択します。(OracleFormsChoice から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
VerticalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollTo	指定した位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMax	スクロール バーの最大の位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMin	スクロール バーの最小の位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムア

名前	説明
WaitForObject	<p>ウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、<code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、<code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、<code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

OracleFormsListView クラス

説明

リストビューを表示します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsListView

構文

```
'Declaration
Public Class OracleFormsListView _
Inherits OracleFormsContainer _
Implements IOracleFormsScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
AllCellTexts	リストビューのすべてのセルのテキスト値。セルは行でグループ化され、各行にはそのセルのテキストが含まれます。
Background	コンポーネントの背景色。(AWTComponent から継承されました)
ColumnCount	リストビューの列の数。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
FocusedRow	フォーカスした行のインデックス。行がフォーカスされていない場合は -1 (4Test の場合は 0) が返されます。
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
HorizontalScrollbarPageSize	スクロールバーのブロック増分値。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarPosition	スクロールバーの現在の位置。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarRange	スクロールバーの範囲値。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarStepSize	スクロールバーの単位増分値。(IOracleFormsScrollable から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
RowCount	リストビューの行の数。
SelectedRow	選択した行のインデックス。行が選択されていない場合は -1 (4Test の場合は 0) が返されます。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VerticalScrollbarPageSize	スクロールバーのブロック増分値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarPosition	スクロールバーの現在の位置。(IOracleFormsScrollable から継承されました)

名前	説明
VerticalScrollbarRange	スクロールバーの範囲値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStepSize	スクロールバーの単位増分値。(IOracleFormsScrollable から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ClearSelection	選択範囲をクリアします。
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetCellText	セル テキストを返します。
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetColumnName	指定したインデックスの列の名前を返します。
GetColumnWidth	指定した列の幅を返します。

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalScrollbar	水平方向のスクロールバーを取得します。 (IOracleFormsScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalScrollbar	垂直方向のスクロールバーを取得します。 (IOracleFormsScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
HorizontalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollTo	指定した位置にスクロールします。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。 (IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。 (IOracleFormsScrollable から継承されました)
HorizontalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
IsRowVisible	指定した行が表示されているかどうかを返します。
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
ScrollToRow	指定した行にスクロールします。
SelectRow	指定した行を選択します。
SetColumnWidth	指定した列の幅を設定します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
VerticalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollTo	指定した位置にスクロールします。 (IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。 (IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。 (IOracleFormsScrollable から継承されました)
VerticalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

OracleFormsMenu クラス

説明

トップレベルメニューを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsMenu
 - [OracleFormsPopupMenu](#)

構文

```
'Declaration  
Public Class OracleFormsMenu _  
Inherits OracleFormsContainer _  
Implements IOracleFormsMenuBase
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	メニュー項目がチェックされているかどうか。(IOracleFormsMenuBase から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 System.Drawing.Imaging.PixelFormat.Format32bppRgb が、このために使用できます。(TestObject から継承されました)
Check	メニュー項目をチェックします
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsChecked	メニュー項目がチェックされているかどうかを示す値。
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	メニュー項目を選択します
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキストフィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	メニュー項目のチェックを外します
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

OracleFormsMenuItem クラス

説明

Menu 内の選択可能な項目を表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsMenuItem

構文

```
'Declaration
Public Class OracleFormsMenuItem _
Inherits OracleFormsContainer _
Implements IOOracleFormsMenuBase
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	メニュー項目がチェックされているかどうか。(IOOracleFormsMenuBase から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)

名前	説明
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Check</i>	メニュー項目をチェックします
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	メニュー項目を選択します
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Uncheck	メニュー項目のチェックを外します
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsPopList クラス

説明

PopList を表します。

継承階層

- [OracleFormsChoice](#)
 - OracleFormsPopList

構文

```
'Declaration
Public Class OracleFormsPopList _
Inherits OracleFormsChoice
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
DropDownOpen	ドロップダウンが現在開いているかどうか。

名前	説明
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
ItemCount	選択肢の項目の数。(OracleFormsChoice から継承されました)
Items	選択肢項目のリスト。(OracleFormsChoice から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	選択されている項目のインデックス。項目が現在選択されていない場合には -1 (4Test の場合は 0) を返します。(OracleFormsChoice から継承されました)
SelectedItem	選択されている項目。(OracleFormsChoice から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetItemText</i>	指定したインデックスの項目を返します。(<i>OracleFormsChoice</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
Select	選択肢で指定した項目を選択します。 (OracleFormsChoice から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsPopupMenu クラス

説明

コンテキスト メニューを表します。

継承階層

- [OracleFormsMenu](#)
 - OracleFormsPopupMenu

構文

```
'Declaration
Public Class OracleFormsPopupMenu _
Inherits OracleFormsMenu
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	メニュー項目がチェックされているかどうか。(IOOracleFormsMenuBase から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
Check	メニュー項目をチェックします (OracleFormsMenu から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsChecked	メニュー項目がチェックされているかどうかを示す値。 (OracleFormsMenu から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
Select	メニュー項目を選択します (OracleFormsMenu から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Uncheck	メニュー項目のチェックを外します (OracleFormsMenu から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

OracleFormsRadioButton クラス

説明

ラジオ ボタンを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsRadioButton

構文

```
'Declaration  
Public Class OracleFormsRadioButton _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Checked	ラジオ ボタンがチェックされているかどうか。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Check	ラジオ ボタンをチェックします。
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsScrollbar クラス

説明

スクロールバーを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsScrollbar
 - [OracleFormsHorizontalScrollbar](#)
 - [OracleFormsVerticalScrollbar](#)

構文

```
'Declaration
Public Class OracleFormsScrollbar _
Inherits OracleFormsContainer _
Implements IOracleFormsScroller
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PageSize	スクロールバーのブロック増分値。(IOracleFormsScroller から継承されました)

名前	説明
Position	スクロールバーの現在の位置。(IOracleFormsScroller から継承されました)
Range	スクロールバーの範囲値。(IOracleFormsScroller から継承されました)
StepSize	スクロールバーの単位増分値。(IOracleFormsScroller から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(AWTComponent から継承されました)
Page	指定したページ数を現在のスクロール位置に加算します。(IOracleFormsScroller から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollTo	指定した位置にスクロールします。(IOracleFormsScroller から継承されました)
ScrollToMax	スクロール バーの最大の位置にスクロールします。(IOracleFormsScroller から継承されました)
ScrollToMin	スクロール バーの最小の位置にスクロールします。(IOracleFormsScroller から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Step	指定したステップ数を現在のスクロール位置に加算します。(IOracleFormsScroller から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsStatusArea クラス

説明

ステータス情報を含む領域を表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsStatusArea

構文

```
'Declaration
Public Class OracleFormsStatusArea _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsStatusBar クラス

説明

ステータスバーを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsStatusBar

構文

```
'Declaration
Public Class OracleFormsStatusBar _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、 <code>ObjectNotFoundException</code> がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsStatusBarItem クラス

説明

ステータスバー内の項目を表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsStatusBarItem

構文

```
'Declaration
Public Class OracleFormsStatusBarItem _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsStatusIndicator クラス

説明

ステータス インジケータを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsStatusIndicator

構文

```
'Declaration
Public Class OracleFormsStatusIndicator _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsTabBar クラス

説明

タブバーを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsTabBar

構文

```
'Declaration
Public Class OracleFormsTabBar _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)

名前	説明
<i>Background</i>	コンポーネントの背景色。(AWTComponent から継承されました)
<i>Cursor</i>	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
<i>Enabled</i>	コンポーネントが有効かどうか。(AWTComponent から継承されました)
<i>Focusable</i>	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
<i>SelectedTab</i>	現在選択されているタブ。
<i>TabCount</i>	タブの数。
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パ

名前	説明
	ラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetTabIndex</i>	タブ インデックスを返します。
<i>GetTabTitle</i>	タブ タイトルを返します。
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>IsTabEnabled</i>	タブが現在有効かどうかを返します。
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>Select</i>	指定したタブを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsTabBarItem クラス

説明

タブバー項目を表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsTabBarItem

構文

```
'Declaration
Public Class OracleFormsTabBarItem _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。 (AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。 (AWTComponent から継承されました)
Background	コンポーネントの背景色。 (AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。 (AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>Select</i>	タブを選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsTabPanel クラス

説明

コンポーネントのコンテナー クラスを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsTabPanel

構文

```
'Declaration
Public Class OracleFormsTabPanel _
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。 (AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。 (AWTComponent から継承されました)
Background	コンポーネントの背景色。 (AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。 (AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。 (AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。 (AWTComponent から継承されました)
Font	コンポーネントのフォント。 (AWTComponent から継承されました)
Foreground	コンポーネントの前景色。 (AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。 (AWTComponent から継承されました)
Name	コンポーネントの名前。 (AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Width	現在のコンポーネントの幅。 (AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
Invoke	テストオブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsTextField クラス

説明

Oracle Forms テキスト フィールドを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsTextField

構文

```
'Declaration
Public Class OracleFormsTextField _
Inherits OracleFormsContainer _
Implements IOracleFormsScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
CaretPosition	テキスト フィールドのテキスト挿入キャレットの位置。
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Editable	テキスト フィールドのテキストをユーザーが変更できるかどうか。
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
HorizontalScrollbarPageSize	スクロール バーのブロック増分値。(IOacleFormsScrollable から継承されました)
HorizontalScrollbarPosition	スクロール バーの現在の位置。(IOacleFormsScrollable から継承されました)
HorizontalScrollbarRange	スクロール バーの範囲値。(IOacleFormsScrollable から継承されました)
HorizontalScrollbarStepSize	スクロール バーの単位増分値。(IOacleFormsScrollable から継承されました)
IsPassword	テキスト フィールドがパスワード テキスト フィールドであるかどうか。
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
PasswordChar	テキスト フィールドがパスワード テキスト フィールドの場合に、実際に入力された文字の代わりに表示する文字。
SelectedText	現在選択されているテキスト。
SelectionEnd	選択されたテキストの終了位置。
SelectionStart	選択されたテキストの開始位置。
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
VerticalScrollbarPageSize	スクロールバーのブロック増分値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarPosition	スクロールバーの現在の位置。(IOracleFormsScrollable から継承されました)
VerticalScrollbarRange	スクロールバーの範囲値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStepSize	スクロールバーの単位増分値。(IOracleFormsScrollable から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Clear	すべてのテキストをテキスト フィールドから削除します
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalScrollbar	水平方向のスクロールバーを取得します。 (IOracleFormsScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalScrollbar	垂直方向のスクロールバーを取得します。 (IOracleFormsScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
HorizontalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollTo	指定した位置にスクロールします。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。 (IOracleFormsScrollable から継承されました)
HorizontalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。 (IOracleFormsScrollable から継承されました)
HorizontalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロールペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>SelectAll</i>	テキスト フィールド内のすべてのテキストを選択します。
<i>SetCaretPosition</i>	テキスト フィールドのテキスト挿入キャレットの位置を設定します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>SetSelectionRange</i>	指定した範囲のテキストを選択します。
<i>SetText</i>	テキスト フィールドのテキストを指定したテキストに設定します。
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
VerticalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollTo	指定した位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsTitleBar クラス

説明

タイトルバーを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsTitleBar

構文

```
'Declaration  
Public Class OracleFormsTitleBar _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsToolBar クラス

説明

ツールバーを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsToolBar

構文

```
'Declaration  
Public Class OracleFormsToolBar _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsToolBarItem クラス

説明

ツールバーの要素を表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsToolBarItem

構文

```
'Declaration  
Public Class OracleFormsToolBarItem _  
Inherits OracleFormsContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)

名前	説明
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。(<i>AWTComponent</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。(<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(<i>AWTComponent</i> から継承されました)
<i>Select</i>	要素を選択します。
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。(<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。(<i>TestObject</i> から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsTree クラス

説明

ツリーを表します。

継承階層

- [OracleFormsContainer](#)
 - OracleFormsTree

構文

```
'Declaration
Public Class OracleFormsTree _
Inherits OracleFormsContainer _
Implements IOracleFormsScrollable
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
HorizontalScrollbarPageSize	スクロールバーのブロック増分値。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarPosition	スクロールバーの現在の位置。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarRange	スクロールバーの範囲値。(IOracleFormsScrollable から継承されました)
HorizontalScrollbarStepSize	スクロールバーの単位増分値。(IOracleFormsScrollable から継承されました)

名前	説明
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
SelectedItemPath	選択された項目パスのリスト。
SelectedItemText	選択された項目のテキスト。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
VerticalScrollbarPageSize	スクロール バーのブロック増分値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarPosition	スクロール バーの現在の位置。(IOracleFormsScrollable から継承されました)
VerticalScrollbarRange	スクロール バーの範囲値。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStepSize	スクロール バーの単位増分値。(IOracleFormsScrollable から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
ClickItem	項目パスによって指定されたツリー ノードをクリックします。
Collapse	指定した項目パスによって識別されるツリー ノードを折りたたみます。
CollapseAll	すべてのツリー ノードを折りたたみます。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClickItem	項目パスによって指定されたツリー ノードをダブルクリックします。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Expand</i>	指定した項目パスによって識別されるツリー ノードを展開します。
<i>ExpandAll</i>	すべてのツリー ノードを展開します。
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalScrollbar</i>	水平方向のスクロール バーを取得します。(<i>IOracleFormsScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalScrollbar</i>	垂直方向のスクロール バーを取得します。(<i>IOracleFormsScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>HorizontalScrollbarPage</i>	指定したページ数を現在のスクロール位置に加算します。(<i>IOracleFormsScrollable</i> から継承されました)
<i>HorizontalScrollbarScrollTo</i>	指定した位置にスクロールします。(<i>IOracleFormsScrollable</i> から継承されました)
<i>HorizontalScrollbarScrollToMax</i>	スクロール バーの最大の位置にスクロールします。(<i>IOracleFormsScrollable</i> から継承されました)
<i>HorizontalScrollbarScrollToMin</i>	スクロール バーの最小の位置にスクロールします。(<i>IOracleFormsScrollable</i> から継承されました)

名前	説明
HorizontalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。 (IOracleFormsScrollable から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsExpanded	指定した項目パスが展開されている場合に 'true' を返します。
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
IsVisible	指定した項目パスが表示されている場合に 'true' を返します。
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
ScrollItemIntoView	指定した項目パスを表示できる位置にスクロールします。
Select	指定した項目パスによって識別されるツリー ノードを選択します。

名前	説明
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
VerticalScrollbarPage	指定したページ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollTo	指定した位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMax	スクロールバーの最大の位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarScrollToMin	スクロールバーの最小の位置にスクロールします。(IOracleFormsScrollable から継承されました)
VerticalScrollbarStep	指定したステップ数を現在のスクロール位置に加算します。(IOracleFormsScrollable から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の

名前	説明
WaitForObject	値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

OracleFormsVerticalScrollbar クラス

説明

垂直方向のスクロールバーを表します。

継承階層

- [OracleFormsScrollbar](#)
 - OracleFormsVerticalScrollbar

構文

```
'Declaration
Public Class OracleFormsVerticalScrollbar _
Inherits OracleFormsScrollbar
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)

名前	説明
<i>Font</i>	コンポーネントのフォント。(AWTComponent から継承されました)
<i>Foreground</i>	コンポーネントの前景色。(AWTComponent から継承されました)
<i>Height</i>	現在のコンポーネントの高さ。(AWTComponent から継承されました)
<i>Name</i>	コンポーネントの名前。(AWTComponent から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
<i>PageSize</i>	スクロールバーのブロック増分値。(IOracleFormsScroller から継承されました)
<i>Position</i>	スクロールバーの現在の位置。(IOracleFormsScroller から継承されました)
<i>Range</i>	スクロールバーの範囲値。(IOracleFormsScroller から継承されました)
<i>StepSize</i>	スクロールバーの単位増分値。(IOracleFormsScroller から継承されました)
<i>Text</i>	コントロールのテキスト。(TestObject から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
<i>Width</i>	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
<i>Click</i>	オブジェクトをクリックします。(IClickable から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定

名前	説明
<i>Exists</i>	されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Find</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この <i>TestObject</i> から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>AWTComponent</i> から継承されました)
<i>Page</i>	指定したページ数を現在のスクロール位置に加算します。 (<i>IOracleFormsScroller</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。 (<i>AWTComponent</i> から継承されました)
<i>ScrollTo</i>	指定した位置にスクロールします。 (<i>IOracleFormsScroller</i> から継承されました)
<i>ScrollToMax</i>	スクロール バーの最大の位置にスクロールします。 (<i>IOracleFormsScroller</i> から継承されました)
<i>ScrollToMin</i>	スクロール バーの最小の位置にスクロールします。 (<i>IOracleFormsScroller</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>Step</i>	指定したステップ数を現在のスクロール位置に加算します。 (<i>IOracleFormsScroller</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

SplitPaneDivider クラス

説明

スプリットペインの分割線を表示します。

継承階層

- [AWTContainer](#)
 - SplitPaneDivider

構文

```
'Declaration  
Public Class SplitPaneDivider _  
Inherits AWTContainer
```

プロパティ

名前	説明
AccessibleName	コンポーネントのアクセス可能な名前。(AWTComponent から継承されました)
AccessibleRole	コンポーネントのアクセス可能なロール。(AWTComponent から継承されました)
Background	コンポーネントの背景色。(AWTComponent から継承されました)
Cursor	コンポーネントに設定されているカーソルの名前。(AWTComponent から継承されました)
Enabled	コンポーネントが有効かどうか。(AWTComponent から継承されました)
Focusable	コンポーネントをフォーカスすることができるかどうか。(AWTComponent から継承されました)
Font	コンポーネントのフォント。(AWTComponent から継承されました)
Foreground	コンポーネントの前景色。(AWTComponent から継承されました)
Height	現在のコンポーネントの高さ。(AWTComponent から継承されました)
Name	コンポーネントの名前。(AWTComponent から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Width	現在のコンポーネントの幅。(AWTComponent から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数

名前	説明
	<code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (AWTComponent から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	コンポーネントが非表示でスクロール ペインの子である場合に、ビューにスクロールします。(AWTComponent から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)

名前	説明
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Win32 クラス リファレンス

Windows API ベースのコントロールのテストに利用可能なクラスを一覧します。

AccessibleControl クラス

説明

ユーザー補助拡張機能によって識別されるコントロール クラス。

継承階層

- [BaseGuiTestObject](#)
 - AccessibleControl

構文

```
'Declaration
Public Class AccessibleControl _
Inherits BaseGuiTestObject
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Description	ユーザー補助オブジェクトの説明文。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Role	このユーザー補助オブジェクトの役割。
State	このユーザー補助オブジェクトのさまざまな状態を説明する状態文字列。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoDefaultAction	基になったユーザー補助コントロールのサポートするデフォルトの操作を実行します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

Header クラス

説明

リストヘッダー コントロールのクラスです。 Header は ListViewW32 クラスのコントロールで最も一般に使用されます。

継承階層

- [Control](#)

- Header

構文

```
'Declaration
Public Class Header _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
ColumnCount	ヘッダーの列の数。
ColumnOrder	ヘッダー項目のインデックス (ヘッダーの項目が現在並んでいる順番)。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DividerDoubleClick	指定した項目の右側にある分割線をダブルクリックします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)

名前	説明
DoubleClick	項目をダブルクリックします。
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetColumnName	指定したヘッダー項目のタイトルを返します。
GetColumnWidth	指定した列の幅を返します。
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)

名前	説明
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Resize	ヘッダーの項目のサイズを変更します。
Select	項目またはボタンを選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Listview クラス

説明

非階層的データのリストを表示するさまざまな方法をサポートする複雑なリスト コントロールのクラスです。これらのビューには、再配置可能な大きいアイコンや小さいアイコン、および詳細 (またはレポート) ビューが含まれます。

継承階層

- [Control](#)
 - ListView

構文

```
'Declaration  
Public Class ListView _  
Inherits Control
```

プロパティ

名前	説明
AllowsCheck	リストビューがチェックをサポートするかどうか。
AllowsMultiSelect	リストビューが複数項目の選択をサポートするかどうか。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
CheckedIndices	チェックされた項目のインデックスを含むリスト。
CheckedItems	チェックされた項目を含むリスト。
ColumnCount	項目ごとの列の数。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	リストビューに含まれる項目の数。
Items	リストビューのすべての項目を含むリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndices	選択された項目のインデックスを含むリスト。
SelectedItems	選択された項目を含むリスト。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Check	項目をチェックします。
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClick	項目をダブルクリックします。
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
ExtendSelect	拡張選択リスト ビューの選択範囲を拡張して項目の範囲を選択します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetContents	ListView コントロールの内容を返します。 返される項目はセミコロンで区切られます。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetItemImageIndex	アイコンを含むリスト ビューのイメージ リストのインデックスを返します。 これにより、正しいイメージが項目に対して使用されていることを検証することができます。
GetItemIndex	リスト ビュー項目 ' ' のインデックスを返します (見つからなかった場合は -1)。

名前	説明
GetItemRect	項目の矩形領域を返します。 <code>getIconRect</code> が設定されていると、項目のアイコンの矩形領域が返されます。設定されていない場合は、項目テキストの矩形領域が返されます。 <code>columnIndex</code> が指定されていると、指定した列の矩形領域が返されます。
GetItemText	項目のテキストを返します。
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この <code>TestObject</code> から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MultiSelect	現在の選択項目の集合にリスト内の項目を追加します。
MultiUnselect	リスト内の項目を選択項目の集合から削除します。

名前	説明
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetItemFocus	リスト内の項目にフォーカスを与えます。
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Uncheck	項目のチェックを外します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

MonthCalendar クラス

説明

カレンダーの日付と時間を選択するクラスです。

継承階層

- [Control](#)
 - MonthCalendar

構文

```
'Declaration
Public Class MonthCalendar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExtendSelect	日付と時間の範囲を選択します。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
OpenMonthMenu	月を指定するためにカレンダーメニューを開きます。

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	日付および時間を選択します。
SelectNextMonth	次の月を選択します。
SelectPreviousMonth	前の月を選択します。
SelectToday	今日の日付を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Pager クラス

説明

タブ付きまたは複数ページのダイアログ、およびボタンのクラスです。Windows では、このクラスは TabControl コントロールに対応します。Windows では、Silk Test は主要なタブのみをサポートしていますが、'サブページ' や '子ページ' などと呼ばれる補助的なタブはサポートしません。

継承階層

- [Control](#)
 - Pager

構文

```
'Declaration
Public Class Pager _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)

名前	説明
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Horizontal	アップダウン コントロールが水平方向に配置されているかどうか。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollDown	スクロールバーを下に移動します。
ScrollUp	スクロールバーを上移動します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、

名前	説明
	ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ProgressBar クラス

説明

進行状況バー コントロールのクラスです。進行状況バーは進行状況を表示します。

継承階層

- [Control](#)
 - ProgressBar

構文

```
'Declaration
Public Class ProgressBar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Position	現在の位置。

名前	説明
Range	進行状況バー コントロールの範囲。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)

名前	説明
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してくだ

名前	説明
WaitForObject	さい。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

StatusBar クラス

説明

ステータス バー コントロールのクラスです。ステータス バーは動的に変更するテキスト ラベルの集合のコンテナです。

継承階層

- [Control](#)
 - StatusBar

構文

```
'Declaration
Public Class StatusBar _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。た

名前	説明
	だし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

UpDown クラス

説明

アップダウン コントロールのクラスです。

継承階層

- [Control](#)
 - UpDown
 - [NumericUpDown](#)

構文

```
'Declaration
Public Class UpDown _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Horizontal	アップダウン コントロールが水平方向に配置されているかどうか。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Position	アップダウン コントロールの位置。
Range	アップダウン コントロールの範囲。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

名前	説明
WrapAround	値が最小値より小さく、最大値より大きい場合にラップ アラウンドするかどうか。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Decrement	アップダウン コントロールの位置の値を減らします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)

名前	説明
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Increment	アップダウン コントロールの位置の値を増やします。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)

名前	説明
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

Windows Forms クラス リファレンス

Windows Forms コントロールのテストに利用可能なクラスを一覧します。

CheckedListBox クラス

説明

CheckedListBox は、チェック ボックスを含むリスト ボックスを表します。

継承階層

- [ListBox](#)
 - CheckedListBox

構文

```
'Declaration
Public Class CheckedListBox _
Inherits ListBox
```

プロパティ

名前	説明
AllowsMultiSelect	コントロールが複数項目の選択をサポートするかどうか。(ListBox から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
CheckedIndices	チェックした項目のインデックス (複数)。
CheckedItems	チェックした項目の名前 (複数)。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	リスト ボックスの項目の数。(ListBox から継承されました)
Items	リスト ボックスの項目のリスト。(ListBox から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
SelectedIndex	最初の選択された項目のインデックス。(ListBox から継承されました)
SelectedIndices	選択した項目の索引。(ListBox から継承されました)
SelectedItem	最初の選択された項目の名前。(ListBox から継承されました)
SelectedItems	選択した項目の名前。(ListBox から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Check	チェック ボックス付きリスト ボックスから項目を選択します。
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
DoubleClick	リスト ボックスの項目をダブルクリックします。(ListBox から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExtendSelect	リスト ボックスの選択範囲を拡張して項目の範囲を選択します。(ListBox から継承されました)

名前	説明
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHorizontalBar</i>	このコントロールの水平スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetVerticalBar</i>	このコントロールの垂直スクロールバーを返します。(<i>IScrollable</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)

名前	説明
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>MultiSelect</i>	複数または拡張選択リスト ボックスの項目を選択します。 (<i>ListBox</i> から継承されました)
<i>MultiUnselect</i>	複数または拡張選択リスト ボックスの項目の選択を解除します。 (<i>ListBox</i> から継承されました)
<i>OpenContextMenu</i>	指定した位置でコンテキスト メニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>Select</i>	リスト ボックスから項目を選択します。 (<i>ListBox</i> から継承されました)
<i>SelectRange</i>	拡張選択リスト ボックスの項目の範囲を選択します。 (<i>ListBox</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>IFocusable</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Uncheck</i>	チェック ボックス付きリスト ボックスから項目を選択します。

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DataGrid クラス

説明

データグリッドとデータグリッドビューのクラスです。

継承階層

- [Control](#)
 - DataGrid

構文

```
'Declaration
Public Class DataGrid _
Inherits Control
```

プロパティ

名前	説明
AllowsMultiSelect	テーブルが複数項目の選択をサポートするかどうか。
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
ColumnCount	DataGrid の列の数。
ColumnItems	列のすべての項目のリスト。
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Items	列のすべての項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
RowCount	DataGrid の行の数。
RowItems	行のすべての項目のリスト。
SelectedItems	選択した項目。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
DoubleClickItem	項目をダブルクリックします。

名前	説明
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExtendSelectItem	指定した項目に選択範囲を拡張します。
Find	XPath ロケータで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
MultiSelectItem	テーブルの行を選択行の集合に追加します。
MultiUnselectItem	項目を選択項目の集合から削除します。
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	データ グリッドの左上隅に項目を移動しようとします。
SelectAllItems	データ グリッド内のすべての項目を選択します。
SelectItem	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)

名前	説明
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DataGridColumn クラス

説明

テーブルにおける列のクラスです。

継承階層

- [Item](#)
 - [DataGridColumn](#)

構文

```
'Declaration  
Public Class DataGridColumn _  
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
Width	列の幅。

メソッド

名前	説明
AutoSize	分割線をダブルクリックして列のサイズを自動的に変更します。
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケータで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケータで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケータを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
Resize	列のサイズを変更します。
ScrollIntoView	データ グリッドをスクロールさせて左側に列を移動しようとします。
Select	列を選択します。
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DataGridItem クラス

説明

データグリッドとデータグリッド ビュー行のクラスです。

継承階層

- [Item](#)
 - DataGridItem

構文

```
'Declaration
Public Class DataGridItem _
Inherits Item
```

プロパティ

名前	説明
<i>Background</i>	GUI オブジェクトの背景色。(<i>GuiTestObject</i> から継承されました)
<i>Column</i>	データグリッドの項目の列。
<i>Enabled</i>	GUI オブジェクトが有効かどうか。(<i>GuiTestObject</i> から継承されました)
<i>Font</i>	GUI オブジェクトのフォントの種類。(<i>GuiTestObject</i> から継承されました)
<i>Foreground</i>	GUI オブジェクトの前景色。(<i>GuiTestObject</i> から継承されました)
<i>NativeHandle</i>	オブジェクトのネイティブ ウィンドウ ハンドル。 (<i>INativeWindow</i> から継承されました)
<i>Row</i>	データグリッドの項目の行。
<i>Selected</i>	項目が選択されているかどうか。
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (<i>TestObject</i> から継承されました)
<i>Visible</i>	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(<i>GuiTestObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。 (<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)

名前	説明
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>HighlightObject</i>	このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)
<i>ImageClick</i>	指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)
<i>ImageClickFile</i>	指定したイメージをクリックします。(<i>TestObject</i> から継承されました)
<i>ImageExists</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageExistsFile</i>	指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangle</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。(<i>TestObject</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。(<i>IFocusable</i> から継承されました)

名前	説明
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	データ グリッドの左上隅に項目を移動しようとします。
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DataGridRow クラス

説明

テーブルにおける行のクラスです。

継承階層

- [Item](#)
 - DataRow

構文

```
'Declaration
Public Class DataRow _
Inherits Item
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)

名前	説明
Height	列の幅。
ItemCount	行の項目の数。
Items	行の項目のリスト。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)

名前	説明
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Resize	列のサイズを変更します。
ScrollIntoView	データ グリッドをスクロールさせて上側に行を移動しようとします。
Select	列を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクト

名前	説明
WaitForObject	を表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomainUpDown クラス

説明

アップダウン コントロールのクラスです。

継承階層

- [Control](#)
 - DomainUpDown

構文

```
'Declaration
Public Class DomainUpDown _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
ItemCount	リストビューに含まれる項目の数。
Items	ドメイン アップダウン コントロールの項目。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
SelectedIndex	アップダウン コントロールの位置。
SelectedItem	アップダウン コントロールの位置。

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WrapAround	値が最小値より小さく、最大値より大きい場合にラップ アラウンドするかどうか。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
Decrement	アップダウン コントロールの位置の値を減らします。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Increment	アップダウン コントロールの位置の値を増やします。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Select	項目を選択します。
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイ

名前	説明
WaitForObject	タイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、 <code>WaitForObject</code> を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して <code>WaitForObject</code> は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、 <code>WaitForObject</code> ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

ElementHost クラス

説明

WPF (Windows Presentation Foundation) 要素をホストするのに使用することができる Windows Forms コントロールです。

継承階層

- [Control](#)
 - ElementHost

構文

```
'Declaration
Public Class ElementHost _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この <i>TestObject</i> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

FormsHost クラス

説明

WPF ページ上に Windows Forms コントロールをホストできるようにする要素です。

継承階層

- [Control](#)
 - FormsHost

構文

```
'Declaration
Public Class FormsHost _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。 デフォルトでは値は常に <i>true</i> です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetDynamicMethodList	この <code>TestObject</code> に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)

名前	説明
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキスト メニューを開きます。(BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

FormsWindow クラス

説明

FormsWindow は、Windows Forms アプリケーションのトップ レベル ウィンドウを表すクラスです。

継承階層

- [Window](#)
 - FormsWindow

構文

```
'Declaration  
Public Class FormsWindow _  
Inherits Window
```

プロパティ

名前	説明
Application	このウィンドウが属するアプリケーションの名前。(Window から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	オブジェクトをクリックします。 (IClickable から継承されました)
Close	ウィンドウを閉じます。 (IMoveable から継承されました)
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。 (IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド (シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。 (IMoveable から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。 (IScrollable から継承されました)
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)

名前	説明
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsActive	ウィンドウがアクティブに設定されているかどうかを返します。(IMoveable から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
LoadAssembly	Windows Forms アプリケーションで指定したパスのアセンブリをロードします。
Maximize	ウィンドウを最大化します。(IMoveable から継承されました)
Minimize	ウィンドウを最小化します。(IMoveable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Move	ウィンドウを移動します。(IMoveable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Restore	ウィンドウを元のサイズに戻します。(IMoveable から継承されました)
SetActive	ウィンドウをアクティブにします。(IMoveable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
Size	ウィンドウのサイズを変更します。(IMoveable から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

MenuStrip クラス

説明

MenuStrip は、Windows Forms アプリケーションの新しいメニュー ストリップを作成するクラスです。

継承階層

- [Control](#)
 - MenuStrip

構文

```
'Declaration
Public Class MenuStrip _
Inherits Control
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)

名前	説明
NativeHandle	オブジェクトのネイティブ ウィンドウハンドル。 (INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)

名前	説明
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。(IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。(BaseGuiTestObject から継承されました)

名前	説明
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。(IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前に

名前	説明
WaitForProperty	<p>トランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

NumericUpDown クラス

説明

数値アップダウン コントロールのクラスです。

継承階層

- [UpDown](#)
 - NumericUpDown

構文

```
'Declaration
Public Class NumericUpDown _
Inherits UpDown
```

プロパティ

名前	説明
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
Horizontal	アップダウン コントロールが水平方向に配置されているかどうか。(UpDown から継承されました)
Maximum	アップダウン コントロールの最大値。
Minimum	アップダウン コントロールの最小値。
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)

名前	説明
<i>Position</i>	アップダウン コントロールの位置。(<i>UpDown</i> から継承されました)
<i>PositionDecimal</i>	アップダウン コントロールの位置。
<i>Range</i>	アップダウン コントロールの範囲。(<i>UpDown</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Visible</i>	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(<i>GuiTestObject</i> から継承されました)
<i>WrapAround</i>	値が最小値より小さく、最大値より大きい場合にラップ アラウンドするかどうか。(<i>UpDown</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>Decrement</i>	アップダウン コントロールの位置の値を減らします。(<i>UpDown</i> から継承されました)
<i>DecrementDecimal</i>	アップダウン コントロールの位置の値を減らします。
<i>DoubleClick</i>	マウス ボタンをオブジェクト上でダブルクリックします。(<i>IClickable</i> から継承されました)
<i>Exists</i>	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケータで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)

名前	説明
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHorizontalBar	このコントロールの水平スクロールバーを返します。(IScrollable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetVerticalBar	このコントロールの垂直スクロールバーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Increment	アップダウン コントロールの位置の値を増やします。(UpDown から継承されました)
IncrementDecimal	アップダウン コントロールの位置の値を増やします。
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
IsFocused	コントロールがフォーカスを持つかどうかを返します。 (IFocusable から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
OpenContextMenu	指定した位置でコンテキストメニューを開きます。 (BaseGuiTestObject から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (IFocusable から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

xBrowser クラス リファレンス

xBrowser コントロールのテストに利用可能なクラスを一覧します。

BrowserApplication クラス

説明

ブラウザー アプリケーションのトップ レベル ウィンドウを表します。

メモ: BrowserApplication は BaseGuiTestObject から派生されており、2 つのスクロールバー メソッド GetHorizontalBar と GetVerticalBar を継承しています。ただし、これら 2 つのメソッドは、BrowserApplication では機能しません。

継承階層

- [Window](#)
 - BrowserApplication

構文

```
'Declaration
Public Class BrowserApplication _
Inherits Window
```

プロパティ

名前	説明
Application	このウィンドウが属するアプリケーションの名前。(Window から継承されました)
Background	GUI オブジェクトの背景色。(GuiTestObject から継承されました)
Enabled	GUI オブジェクトが有効かどうか。(GuiTestObject から継承されました)
Font	GUI オブジェクトのフォントの種類。(GuiTestObject から継承されました)
Foreground	GUI オブジェクトの前景色。(GuiTestObject から継承されました)
NativeHandle	オブジェクトのネイティブ ウィンドウ ハンドル。(INativeWindow から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(GuiTestObject から継承されました)
WindowState	ウィンドウの状態。有効な値は、1=最小化、2=最大化、3=元のサイズに戻すです。(IMoveable から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
ClearCache	ブラウザのキャッシュをクリアします。
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	ウィンドウを閉じます。(IMoveable から継承されました)
CloseOtherTabs	アクティブなタブ以外のすべてのタブを閉じます。複数のタブが開いている場合には、Internet Explorer では、一部のタブと、スクロール ボタンが表示されます。このメソッドは選択されているタブ以外のすべてのタブを現在表示されていないタブも含めて閉じます。このメソッドはモバイル Web アプリケーションではサポートされていません。

名前	説明
CloseSynchron	ウィンドウを閉じ、ウィンドウが実際に閉じられるまで待ちます。 (IMoveable から継承されました)
CloseTab	指定したタブを閉じます。タブが指定されていない場合、アクティブなタブが閉じられます。複数のタブが開いている場合には、Internet Explorer では、一部のタブと、スクロール ボタンが表示されます。結果として、現在表示されているタブだけが、インデックスの影響を受けるこの方式と機能します。たとえば、現在表示されているタブだけを閉じることができます。インデックス 0 は最初の表示されているタブを識別します。このメソッドはモバイル Web アプリケーションではサポートされていません。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
ExistsTab	タブが存在する場合に true、それ以外は false を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetActiveTab	アクティブなタブを返します。このメソッドはモバイル Web アプリケーションではサポートされていません。
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCloseWindows	アプリケーションのメイン ウィンドウだけが開いている状態にするために、閉じる必要のあるウィンドウを返します。(IMoveable から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetFocus	入力フォーカスを持つオブジェクトを返します。(IMoveable から継承されました)
GetHorizontalBar	このコントロールの水平スクロール バーを返します。(IScrollable から継承されました)

名前	説明
GetNextCloseWindow	アプリケーションのメイン ウィンドウ以外のすべてのウィンドウを閉じるために、閉じる必要のある次のウィンドウを返します。 (IMoveable から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetSelectedTab	現在選択されているタブを返します。 このメソッドはモバイル Web アプリケーションではサポートされていません。
GetSelectedTabIndex	アクティブなタブのインデックスを返します。 複数のタブが開いている場合には、Internet Explorer では、一部のタブと、スクロール ボタンが表示されます。 結果として、このメソッドは現在表示されているタブに相対的なインデックスのみを返します。 インデックス 0 は最初の表示されているタブを識別します。 このメソッドはモバイル Web アプリケーションではサポートされていません。
GetSelectedTabName	アクティブなタブの名前を返します。 このメソッドはモバイル Web アプリケーションではサポートされていません。
GetTabCount	開いているタブの数を返します。 複数のタブが開いている場合には、Internet Explorer では、一部のタブと、スクロール ボタンが表示されます。 結果として、このメソッドは現在表示されているタブの数のみを返します。 このメソッドはモバイル Web アプリケーションではサポートされていません。
GetVerticalBar	このコントロールの垂直スクロール バーを返します。(IScrollable から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsActive</i>	ウィンドウがアクティブに設定されているかどうかを返します。 (<i>IMoveable</i> から継承されました)
<i>IsFocused</i>	コントロールがフォーカスを持つかどうかを返します。 (<i>IFocusable</i> から継承されました)
<i>Maximize</i>	ウィンドウを最大化します。(<i>IMoveable</i> から継承されました)
<i>Minimize</i>	ウィンドウを最小化します。(<i>IMoveable</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Move</i>	ウィンドウを移動します。(<i>IMoveable</i> から継承されました)
<i>Navigate</i>	現在アクティブなタブ/ウィンドウで指定した URL に移動します
<i>OpenContextMenu</i>	指定した位置でコンテキストメニューを開きます。 (<i>BaseGuiTestObject</i> から継承されました)
<i>OpenTab</i>	新しいタブを開き、指定した URL に移動します。URL が指定されていない場合には、タブは about:blank に移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。(<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)
<i>Restore</i>	ウィンドウを元のサイズに戻します。(<i>IMoveable</i> から継承されました)
<i>SelectTab</i>	指定したタブを選択します。複数のタブが開いている場合には、Internet Explorer では、一部のタブと、スクロール ボタンが表示されます。結果として、現在表示されているタブだけが、インデックスの影響を受けるこの方式と機能します。たとえば、現在表示されているタブだけが選択することができます。インデックス 0 は最初の表示されているタブを選択します。このメソッドはモバイル Web アプリケーションではサポートされていません。
<i>SetActive</i>	ウィンドウをアクティブにします。(<i>IMoveable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。(<i>IFocusable</i> から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。(TestObject から継承されました)
 Size 	ウィンドウのサイズを変更します。(IMoveable から継承されました)
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。(TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラ

名前	説明
	メーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

BrowserObject クラス

説明

BrowserApplication 内に含まれているすべてのオブジェクトの基本クラスです。

継承階層

- [TestObject](#)
 - BrowserObject
 - [BrowserWindow](#)
 - [DomElement](#)

構文

```
'Declaration
Public Class BrowserObject _
Inherits TestObject _
Implements IClickable, IKeyable
```

プロパティ

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。.NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)

名前	説明
Click	オブジェクトをクリックします。(IClickable から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
<i>ImageRectangleFile</i>	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Invoke</i>	テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>TypeKeys</i>	一連のキーストロークをオブジェクトに送信します。 (<i>IKeyable</i> から継承されました)
<i>TypePasswordKeys</i>	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (<i>IKeyable</i> から継承されました)
<i>Verify</i>	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (<i>TestObject</i> から継承されました)
<i>WaitForChildDisappearance</i>	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (<i>TestObject</i> から継承されました)

名前	説明
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

BrowserWindow クラス

説明

ブラウザ ウィンドウは Web ページのトップ レベル コンテナです。DomElement を使って DOM ツリーにアクセスできます。たとえば、IE7 ではタブ、SWT では埋め込みブラウザ コントロールになります。

継承階層

- [BrowserObject](#)
 - BrowserWindow

構文

```
'Declaration
Public Class BrowserWindow _
Inherits BrowserObject
```

プロパティ

名前	説明
Text	コントロールのテキスト。(TestObject から継承されました)

名前	説明
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
ViewportHeight	ブラウザー ビューポートの高さ
ViewportName	ブラウザー ウィンドウの innerWidth と innerHeight に一致するビューポートの名前
ViewportOrientation	ブラウザー ビューポートの向き。 横向きか縦向きを指定できます。
ViewportWidth	ブラウザー ビューポートの幅
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
AcceptAlert	OK を押して JavaScript 警告を承認します。
Back	履歴の "戻る" エントリに移動します。
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
CaptureFullPageBitmap	Web ページ全体のスクリーンショットをキャプチャして、指定したパスにファイルを保存します。ファイル名の拡張子でファイルの種類を指定できます。有効な拡張子は、.jpg、.jpeg、.png、.bmp です。
Click	オブジェクトをクリックします。(IClickable から継承されました)
Close	モーダル ブラウザー ダイアログ、ブラウザーのタブ、または最後のタブの場合は (または IE6 の場合) ブラウザー自身を閉じます。
DismissAlert	JavaScript 警告を解除します。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
ExecuteJavaScript	トップ レベル ドキュメントのコンテキストで JavaScript コードを評価します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
Forward	履歴の "進む" エントリに移動します。
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetAlertText	JavaScript 警告のメッセージテキストを取得します。
GetBrowserMajorVersion	ブラウザのメジャーバージョン番号を返します
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow の矩形領域を HTML ピクセルで取得します。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザ内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザ対応になり、位置計算はモバイルブラウザの拡大レベルの変更の影響を受けなくなります。
GetPageSource	Web サイトのページのソースを返します
GetParent	テストアプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetUrl	メインドキュメントの URL を返します。
GetUserAgent	ブラウザのユーザーエージェント文字列を返します。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)

名前	説明
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
IsAlertPresent	JavaScript 警告が開いている場合に true を返します。
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Navigate	メイン ドキュメントを URL に移動します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
Reload	ブラウザの再読み込みボタンの動作と同様に、メイン ドキュメントを再読み込みします。
SetFocus	コントロールにフォーカスを与えます。(BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
SetViewportName	ブラウザ ウィンドウのサイズをブラウザ サイズ リストで指定したサイズに設定します。
SetViewportSize	ブラウザ ウィンドウのサイズを指定した幅と高さに設定します。
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

DomButton クラス

説明

DomButton は次のタグのいずれかを使用して指定されるすべての DOM 要素を表します： <input type='submit'>、<input type='reset'>、<input type='button'>、または <button>。

継承階層

- [DomElement](#)

- DomButton

構文

```
'Declaration
Public Class DomButton _
Inherits DomElement
```

プロパティ

名前	説明
<i>IsFocused</i>	DOM 要素がフォーカスを持っているかどうか。(<i>DomElement</i> から継承されました)
<i>Text</i>	コントロールのテキスト。(<i>TestObject</i> から継承されました)
<i>Value</i>	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(<i>TestObject</i> から継承されました)
<i>Visible</i>	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(<i>BrowserObject</i> から継承されました)

メソッド

名前	説明
<i>CaptureBitmap</i>	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(<i>TestObject</i> から継承されました)
<i>Click</i>	オブジェクトをクリックします。(<i>IClickable</i> から継承されました)
<i>DomClick</i>	DOM API を使用してクリックを呼び出します。 DomElement やそのサブクラスの低レベルな再生で使用される Click 関数を呼び出すこともできます。(<i>DomElement</i> から継承されました)
<i>DomDoubleClick</i>	DOM API を使用してダブルクリックを呼び出します。 DomElement やそのサブクラスの低レベルな再生で使用される DoubleClick 関数を呼び出すこともできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>DomElement</i> から継承されました)
<i>DomMouseMove</i>	DOM API を使用してマウスの移動を呼び出します。 DomElement やそのサブクラスの低レベルな再生で使用される MouseMove 関数を使用することもできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>DomElement</i> から継承されました)

名前	説明
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (DomElement から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。 (DomElement から継承されました)
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。 (DomElement から継承されました)
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。 (DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。(DomElement から継承されました)
Highlight	DOM 要素をハイライトします。(DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。(DomElement から継承されました)
Select	ボタンをクリックします。

名前	説明
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。 (DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。 (DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケータに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイム

名前	説明
WaitForProperty	<p>アウト値を増やすことを検討してください。(TestObject から継承されました)</p> <p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

DomCheckBox クラス

説明

DomCheckBox は <input type='checkbox'> タグを使用して指定されるすべての DOM 要素を表します。

継承階層

- [DomElement](#)
 - DomCheckBox

構文

```
'Declaration
Public Class DomCheckBox _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。(DomElement から継承されました)
State	チェック ボックスの状態。 有効な値は、1=チェックあり、2=チェックなし、3=未設定 です。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行している

名前	説明
	<p>マシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Check	チェック ボックスをチェックします。
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DomClick	DOM API を使用してクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>Click</code> 関数を呼び出すこともできます。 (DomElement から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>DoubleClick</code> 関数を呼び出すこともできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>MouseMove</code> 関数を使用することもできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (DomElement から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうか検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。 (DomElement から継承されました)

名前	説明
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。 (DomElement から継承されました)
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む)のリストを返します。 (TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。 GetRect 関数によって提供される画面座標と異なり、 GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響を受けません。 Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。 ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。 これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。 (DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。 (DomElement から継承されました)
Highlight	DOM 要素をハイライトします。 (DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
<i>Invoke</i>	テストオブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>InvokeMethods</i>	この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)
<i>IsChecked</i>	チェックボックスがチェックされている場合は <i>true</i> を、それ以外の場合は <i>false</i> を返します。
<i>MouseMove</i>	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>PressKeys</i>	一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)
<i>PressMouse</i>	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ReleaseKeys</i>	一連のキーまたはマウス ボタンを離します。 (<i>IKeyable</i> から継承されました)
<i>ReleaseMouse</i>	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (<i>IClickable</i> から継承されました)
<i>ScrollIntoView</i>	ブラウザ ウィンドウの表示領域に DOM 要素をスクロールします。 (<i>DomElement</i> から継承されました)
<i>Select</i>	チェックボックスを選択します
<i>SetDomAttribute</i>	オブジェクトに指定された DOM 属性の値を設定します。 (<i>DomElement</i> から継承されました)
<i>SetFocus</i>	コントロールにフォーカスを与えます。 (<i>BrowserObject</i> から継承されました)
<i>SetProperty</i>	指定したプロパティの値を設定します。 (<i>TestObject</i> から継承されました)
<i>SetUndecided</i>	チェックボックスを未設定の状態に設定します。
<i>TextCapture</i>	このオブジェクトの表示領域内のテキストを返します。 (<i>TestObject</i> から継承されました)
<i>TextClick</i>	指定したテキストの中央をクリックします。 (<i>TestObject</i> から継承されました)
<i>TextExists</i>	指定したテキストが存在するかどうかを返します。 (<i>TestObject</i> から継承されました)
<i>TextRectangle</i>	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (<i>TestObject</i> から継承されました)
<i>Toggle</i>	チェックボックスの状態を切り替えます。チェックボックスがチェックされている場合は、チェックを外し、そうでない場合はチェックします。チェックボックスが未設定の状態にある場合は、未設定のままになります。

名前	説明
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Uncheck	チェック ボックスのチェックを外します。
Unhighlight	元的前景色と背景色に戻します。 (DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)

DomElement クラス

説明

DomElement は DOM ツリーの任意の要素を表します。

継承階層

- [BrowserObject](#)
 - DomElement
 - [DomButton](#)
 - [DomCheckBox](#)
 - [DomEmbeddedElement](#)
 - [DomForm](#)
 - [DomLink](#)
 - [DomListBox](#)
 - [DomRadioButton](#)
 - [DomTable](#)
 - [DomTableRow](#)
 - [DomTextField](#)

構文

```
'Declaration  
Public Class DomElement _  
Inherits BrowserObject
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。.NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
DomClick	DOM API を使用してクリックを呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される Click 関数を呼び出すこともできます。
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される DoubleClick 関数を呼び出すこともできます。このメソッドはモバイル Web アプリケーションではサポートされていません。
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される MouseMove 関数を使用することもできます。このメソッドはモバイル Web アプリケーションではサポートされていません。
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これ

名前	説明
	によって、スクリプトがクロスブラウザー対応になり、位置計算はモバイルブラウザーの拡大レベルの変更の影響を受けなくなります。
GetParent	テストアプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。
Highlight	DOM 要素をハイライトします。
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)

名前	説明
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。
SetFocus	コントロールにフォーカスを与えます。(BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomEmbeddedElement クラス

説明

<embed> タグ、または <object> タグで指定された DOM 要素。

継承階層

- [DomElement](#)
 - DomEmbeddedElement

構文

```
'Declaration
Public Class DomEmbeddedElement _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。(DomElement から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)</p>
Click	<p>オブジェクトをクリックします。 (IClickable から継承されました)</p>
DomClick	<p>DOM API を使用してクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>Click</code> 関数を呼び出すこともできます。 (DomElement から継承されました)</p>
DomDoubleClick	<p>DOM API を使用してダブルクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>DoubleClick</code> 関数を呼び出すこともできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)</p>
DomMouseMove	<p>DOM API を使用してマウスの移動を呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>MouseMove</code> 関数を使用することもできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)</p>
DoubleClick	<p>マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)</p>
ExecuteJavaScript	<p>親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (DomElement から継承されました)</p>
Exists	<p>ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (TestObject から継承されました)</p>
Exists	<p>オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (TestObject から継承されました)</p>
Find	<p>XPath ロケーターで指定したオブジェクトを検索します。 (TestObject から継承されました)</p>
FindAll	<p>XPath ロケーターで指定したすべてのオブジェクトを検索します。 (TestObject から継承されました)</p>
GenerateLocator	<p>このオブジェクトのロケーターを返します。 (TestObject から継承されました)</p>
GetChildren	<p>このオブジェクトの子オブジェクトを返します。 (TestObject から継承されました)</p>
GetCssStyle	<p>指定したスタイル名の算出 CSS スタイルを取得します。 (DomElement から継承されました)</p>
GetDomAttribute	<p>オブジェクトに指定された DOM 属性の値を取得します。 (DomElement から継承されました)</p>

名前	説明
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。(DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。(DomElement から継承されました)
Highlight	DOM 要素をハイライトします。(DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。 (DomElement から継承されました)
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。 (DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。 (DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指

名前	説明
WaitForDisappearance	<p>定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p> <p>オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>
WaitForObject	<p>指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)</p>
WaitForProperty	<p>'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)</p>

DomForm クラス

説明

DomForm は <form> タグを使用して指定されるすべての DOM 要素を表します。このクラスのすべてのメソッドとプロパティはモバイル Web アプリケーションではサポートされていません。

継承階層

- [DomElement](#)
 - DomForm

構文

```
'Declaration
Public Class DomForm _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。 (DomElement から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。 (BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DomClick	DOM API を使用してクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>Click</code> 関数を呼び出すこともできます。 (DomElement から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>DoubleClick</code> 関数を呼び出すこともできます。このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>MouseMove</code> 関数を使用することもできます。このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (DomElement から継承されました)
Exists	ロケータに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetCssStyle</i>	指定したスタイル名の算出 CSS スタイルを取得します。(<i>DomElement</i> から継承されました)
<i>GetDomAttribute</i>	オブジェクトに指定された DOM 属性の値を取得します。(<i>DomElement</i> から継承されました)
<i>GetDomAttributeList</i>	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>DomElement</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHtmlRect</i>	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。(<i>DomElement</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetRect</i>	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)
<i>GetText</i>	DOM 要素の表示されるテキストを取得します。(<i>DomElement</i> から継承されました)

名前	説明
Highlight	DOM 要素をハイライトします。 (DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。 (DomElement から継承されました)
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。 (DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
Submit	フォームを送信します。

名前	説明
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。(DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイム

名前	説明
	ムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomLink クラス

説明

DomLink は <a> タグを使用して指定されるすべての DOM 要素を表します。

継承階層

- [DomElement](#)
 - DomLink

構文

```
'Declaration
Public Class DomLink _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。(DomElement から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)

名前	説明
DomClick	DOM API を使用してクリックを呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される Click 関数を呼び出すこともできます。(DomElement から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される DoubleClick 関数を呼び出すこともできます。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される MouseMove 関数を使用することもできます。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。(DomElement から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに false が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。(DomElement から継承されました)
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。(DomElement から継承されました)
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、

名前	説明
	<p>GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。 (<i>DomElement</i> から継承されました)</p>
<i>GetParent</i>	<p>テスト アプリケーション内でこのオブジェクトの親を検索します。 (<i>TestObject</i> から継承されました)</p>
<i>GetProperty</i>	<p>指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)</p>
<i>GetPropertyList</i>	<p>指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)</p>
<i>GetRect</i>	<p>このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(<i>TestObject</i> から継承されました)</p>
<i>GetText</i>	<p>DOM 要素の表示されるテキストを取得します。(<i>DomElement</i> から継承されました)</p>
<i>Highlight</i>	<p>DOM 要素をハイライトします。(<i>DomElement</i> から継承されました)</p>
<i>HighlightObject</i>	<p>このオブジェクトをハイライトします。(<i>TestObject</i> から継承されました)</p>
<i>ImageClick</i>	<p>指定したイメージ資産をクリックします。(<i>TestObject</i> から継承されました)</p>
<i>ImageClickFile</i>	<p>指定したイメージをクリックします。(<i>TestObject</i> から継承されました)</p>
<i>ImageExists</i>	<p>指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)</p>
<i>ImageExistsFile</i>	<p>指定したイメージが存在するかどうかを返します。(<i>TestObject</i> から継承されました)</p>
<i>ImageRectangle</i>	<p>指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)</p>
<i>ImageRectangleFile</i>	<p>指定したイメージのオブジェクトに相対的な矩形領域を返します。(<i>TestObject</i> から継承されました)</p>
<i>Invoke</i>	<p>テスト オブジェクトのメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)</p>
<i>InvokeMethods</i>	<p>この TestObject から始まる一連のメソッドを動的に呼び出します。 (<i>TestObject</i> から継承されました)</p>
<i>MouseMove</i>	<p>オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>IClickable</i> から継承されました)</p>
<i>PressKeys</i>	<p>一連のキーまたはマウス ボタンを押します (離しません)。 (<i>IKeyable</i> から継承されました)</p>

名前	説明
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。(DomElement から継承されました)
Select	リンクをクリックします。
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。(DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。(BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。(DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト

名前	説明
WaitForObject	値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション <code>OPT_WAIT_RESOLVE_OBJDEF</code> によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomListBox クラス

説明

DomListBox は <select> タグを使用して指定されるすべての DOM 要素を表します。

継承階層

- [DomElement](#)
 - DomListBox

構文

```
'Declaration
Public Class DomListBox _
Inherits DomElement
```

プロパティ

名前	説明
AllowsMultiSelect	コントロールが複数選択可能かどうか。
IsFocused	DOM 要素がフォーカスを持っているかどうか。(DomElement から継承されました)
ItemCount	リスト ボックスの項目の数。
Items	リスト ボックスの項目のリスト。
SelectedIndices	選択した項目の索引。

名前	説明
SelectedItems	選択した項目の名前。
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのために、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。(TestObject から継承されました)
Click	オブジェクトをクリックします。(IClickable から継承されました)
DomClick	DOM API を使用してクリックを呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される Click 関数を呼び出すこともできます。(DomElement から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される DoubleClick 関数を呼び出すこともできます。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。DomElement やそのサブクラスの低レベルな再生で使用される MouseMove 関数を使用することもできます。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。(IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。(DomElement から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。timeout パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。(TestObject から継承されました)

名前	説明
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。(DomElement から継承されました)
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。(DomElement から継承されました)
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響を受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。(DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。(DomElement から継承されました)
Highlight	DOM 要素をハイライトします。(DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)

名前	説明
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
MultiSelect	項目のリストを選択します。
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ScrollIntoView	ブラウザ ウィンドウの表示領域に DOM 要素をスクロールします。(DomElement から継承されました)
Select	項目を選択します。
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。(DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。(BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)

名前	説明
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。(DomElement から継承されました)
Unselect	項目の選択を解除します。
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomRadioButton クラス

説明

DomRadioButton は `<input type='radio'>` タグを使用して指定されるすべての DOM 要素を表します。

継承階層

- [DomElement](#)
 - DomRadioButton

構文

```
'Declaration
Public Class DomRadioButton _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。 (DomElement から継承されました)
RadioListItemCount	ラジオ リストの項目の数。 このプロパティはモバイル Web アプリケーションではサポートされていません。
RadioListItems	ラジオ リストの項目のリスト。
RadioListSelectedIndex	選択する項目のインデックス。 Silk Test 17.5 で削除されたため、このプロパティの set アクセス操作子は機能しません。 このプロパティはモバイル Web アプリケーションではサポートされていません。
RadioListSelectedItem	ラジオ リストで選択されている項目。 Silk Test 17.5 で削除されたため、このプロパティの set アクセス操作子は機能しません。 このプロパティはモバイル Web アプリケーションではサポートされていません。
Selected	ラジオ ボタンが選択されているかどうか。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。 デフォルトでは値は常に <code>true</code> です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。 (BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。

名前	説明
	<p>キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
Click	オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)
DomClick	DOM API を使用してクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>Click</code> 関数を呼び出すこともできます。 (<code>DomElement</code> から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>DoubleClick</code> 関数を呼び出すこともできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (<code>DomElement</code> から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>MouseMove</code> 関数を使用することもできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (<code>DomElement</code> から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (<code>DomElement</code> から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。 (<code>DomElement</code> から継承されました)
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。 (<code>DomElement</code> から継承されました)

名前	説明
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。(DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。(DomElement から継承されました)
Highlight	DOM 要素をハイライトします。(DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)

名前	説明
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
RadioListSelect	ラジオ リストの項目を選択します。
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	ブラウザ ウィンドウの表示領域に DOM 要素をスクロールします。 (DomElement から継承されました)
Select	ラジオ ボタンを選択します。
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。 (DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。 (DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)

名前	説明
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomTable クラス

説明

DomTable は <table> タグを表します。このクラスのすべてのメソッドとプロパティはモバイル Web アプリケーションではサポートされていません。

継承階層

- [DomElement](#)
 - DomTable

構文

```
'Declaration
Public Class DomTable _
Inherits DomElement
```


プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。 (DomElement から継承されました)
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <code>true</code> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。 (BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (TestObject から継承されました)
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DomClick	DOM API を使用してクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>Click</code> 関数を呼び出すこともできます。 (DomElement から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>DoubleClick</code> 関数を呼び出すこともできます。このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>MouseMove</code> 関数を使用することもできます。このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (DomElement から継承されました)
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムア

名前	説明
	ウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(<i>TestObject</i> から継承されました)
<i>Exists</i>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(<i>TestObject</i> から継承されました)
<i>Find</i>	XPath ロケーターで指定したオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>FindAll</i>	XPath ロケーターで指定したすべてのオブジェクトを検索します。(<i>TestObject</i> から継承されました)
<i>GenerateLocator</i>	このオブジェクトのロケーターを返します。(<i>TestObject</i> から継承されました)
<i>GetCell</i>	指定したセルを返します。
<i>GetCellText</i>	セル要素のコンテンツを返します。
<i>GetChildren</i>	このオブジェクトの子オブジェクトを返します。(<i>TestObject</i> から継承されました)
<i>GetColumnCount</i>	テーブルの列の数を返します。
<i>GetCssStyle</i>	指定したスタイル名の算出 CSS スタイルを取得します。(<i>DomElement</i> から継承されました)
<i>GetDomAttribute</i>	オブジェクトに指定された DOM 属性の値を取得します。(<i>DomElement</i> から継承されました)
<i>GetDomAttributeList</i>	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(<i>DomElement</i> から継承されました)
<i>GetDynamicMethodList</i>	この TestObject に対して動的に呼び出すことのできるメソッド(シグネチャを含む) のリストを返します。(<i>TestObject</i> から継承されました)
<i>GetHtmlRect</i>	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響は受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。(<i>DomElement</i> から継承されました)
<i>GetParent</i>	テスト アプリケーション内でこのオブジェクトの親を検索します。(<i>TestObject</i> から継承されました)
<i>GetProperty</i>	指定したプロパティの値を返します。(<i>TestObject</i> から継承されました)
<i>GetPropertyList</i>	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(<i>TestObject</i> から継承されました)

名前	説明
GetRect	このオブジェクトの大きさと位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)
GetRow	指定した行の DOM 要素を返します。
GetRowCount	行の数を返します
GetRows	行 (DomTableRow) のリストを返します。
GetRowText	指定した行のすべてのセル要素のコンテンツを返します。
GetText	DOM 要素の表示されるテキストを取得します。(DomElement から継承されました)
Highlight	DOM 要素をハイライトします。(DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。(TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。(TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。(TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。(TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。(TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。(TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。(IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。(IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。(IClickable から継承されました)

名前	説明
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。(DomElement から継承されました)
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。(DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。(BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。(TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。(TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。(TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。(TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。(TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。(IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。(IKeyable から継承されました)
Unhighlight	元の前景色と背景色に戻します。(DomElement から継承されました)
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中

名前	説明
WaitForObject	に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomTableRow クラス

説明

DomTableRow は <tr> タグを表します。 このクラスのすべてのメソッドとプロパティはモバイル Web アプリケーションではサポートされていません。

継承階層

- [DomElement](#)
 - DomTableRow

構文

```
'Declaration
Public Class DomTableRow _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。(DomElement から継承されました)
Text	コントロールのテキスト。(TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。(TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。 表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に true です。 ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。(BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。 絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存さ

名前	説明
	<p>れます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのため、イメージの解析時には無視されます。 .NET では、定数 <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> が、このために使用できます。 (<code>TestObject</code> から継承されました)</p>
<code>Click</code>	オブジェクトをクリックします。 (<code>IClickable</code> から継承されました)
<code>DomClick</code>	DOM API を使用してクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>Click</code> 関数を呼び出すこともできます。 (<code>DomElement</code> から継承されました)
<code>DomDoubleClick</code>	DOM API を使用してダブルクリックを呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>DoubleClick</code> 関数を呼び出すこともできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (<code>DomElement</code> から継承されました)
<code>DomMouseMove</code>	DOM API を使用してマウスの移動を呼び出します。 <code>DomElement</code> やそのサブクラスの低レベルな再生で使用される <code>MouseMove</code> 関数を使用することもできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (<code>DomElement</code> から継承されました)
<code>DoubleClick</code>	マウス ボタンをオブジェクト上でダブルクリックします。 (<code>IClickable</code> から継承されました)
<code>ExecuteJavaScript</code>	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (<code>DomElement</code> から継承されました)
<code>Exists</code>	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <code>timeout</code> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。 タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <code>false</code> が返されます。 (<code>TestObject</code> から継承されました)
<code>Exists</code>	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。 (<code>TestObject</code> から継承されました)
<code>Find</code>	XPath ロケーターで指定したオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
<code>FindAll</code>	XPath ロケーターで指定したすべてのオブジェクトを検索します。 (<code>TestObject</code> から継承されました)
<code>GenerateLocator</code>	このオブジェクトのロケーターを返します。 (<code>TestObject</code> から継承されました)
<code>GetCell</code>	指定したセルの DOM 要素を返します。
<code>GetCellCount</code>	このテーブル行のセルの数を返します
<code>GetChildren</code>	このオブジェクトの子オブジェクトを返します。 (<code>TestObject</code> から継承されました)

名前	説明
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。 (DomElement から継承されました)
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。 (DomElement から継承されました)
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。 (TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。 GetRect 関数によって提供される画面座標と異なり、 GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響を受けません。 Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。 ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。 これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。 (DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。 (TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。 (TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。 (TestObject から継承されました)
GetRect	このオブジェクトの大きさと位置を返します。 デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。 (TestObject から継承されました)
GetText	DOM 要素の表示されるテキストを取得します。 (DomElement から継承されました)
Highlight	DOM 要素をハイライトします。 (DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)

名前	説明
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインタを移動します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。 (DomElement から継承されました)
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。 (DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (BrowserObject から継承されました)
SetProperty	指定したプロパティの値を設定します。 (TestObject から継承されました)
TextCapture	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
TextClick	指定したテキストの中央をクリックします。 (TestObject から継承されました)
TextExists	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
TextRectangle	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
TypeKeys	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
TypePasswordKeys	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
Unhighlight	元的前景色と背景色に戻します。 (DomElement から継承されました)

名前	説明
Verify	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。(TestObject から継承されました)
WaitForChildDisappearance	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)
WaitForObject	指定したロケーターに一致するオブジェクトを待機します。タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。通常のスクリプト実行中に不規則にタイムアウトエラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。(TestObject から継承されました)
WaitForProperty	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイムアウトに到達するまで待機します。オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。タイムアウト値に到達すると例外がスローされます。(TestObject から継承されました)

DomTextField クラス

説明

DomTextField は次のタグのいずれかを使用して指定されるすべての DOM 要素を表します： <input type='text'>、<input type='password'>、<input type='file'>、または <textarea>。

継承階層

- [DomElement](#)
 - DomTextField

構文

```
'Declaration
Public Class DomTextField _
Inherits DomElement
```

プロパティ

名前	説明
IsFocused	DOM 要素がフォーカスを持っているかどうか。 (DomElement から継承されました)
IsPassword	テキスト フィールドがパスワード テキスト フィールドであるかどうか。
Text	コントロールのテキスト。 (TestObject から継承されました)
Value	コントロールの値 (たとえば、テキスト コントロールのテキスト)。 (TestObject から継承されました)
Visible	オブジェクトが表示されるかどうかを指定します。表示可能なオブジェクトのみ検索できます。デフォルトでは値は常に <i>true</i> です。ただし、テスト対象アプリケーションですでに検索したオブジェクトが再生中に非表示になった場合に、このプロパティが必要になる場合があります。 (BrowserObject から継承されました)

メソッド

名前	説明
CaptureBitmap	このオブジェクトのビットマップ イメージをファイルに保存します。絶対ファイル名で指定しない場合、エージェントを実行しているマシンのデフォルトのビットマップ保存場所にビットマップが保存されます (%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps)。キャプチャされたビットマップのピクセルは 32 ビット RGB 順で保存されます。それぞれのピクセルの 4 バイト目 (アルファ チャンネル) はパディングのためで、イメージの解析時には無視されます。 .NET では、定数 <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> が、このために使用できます。 (TestObject から継承されました)
ClearText	テキスト フィールドのテキストをクリアします
Click	オブジェクトをクリックします。 (IClickable から継承されました)
DomClick	DOM API を使用してクリックを呼び出します。 DomElement やそのサブクラスの低レベルな再生で使用される Click 関数を呼び出すこともできます。 (DomElement から継承されました)
DomDoubleClick	DOM API を使用してダブルクリックを呼び出します。 DomElement やそのサブクラスの低レベルな再生で使用される DoubleClick 関数を呼び出すこともできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DomMouseMove	DOM API を使用してマウスの移動を呼び出します。 DomElement やそのサブクラスの低レベルな再生で使用される MouseMove 関数を使用することもできます。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (DomElement から継承されました)
DoubleClick	マウス ボタンをオブジェクト上でダブルクリックします。 (IClickable から継承されました)
ExecuteJavaScript	親ドキュメントのコンテキストで指定した JavaScript コードを評価します。 (DomElement から継承されました)

名前	説明
Exists	ロケーターに一致する子オブジェクトがテスト対象アプリケーションのオブジェクトに対して存在するかどうかを検査します。 <i>timeout</i> パラメーターが指定された場合は、エージェントは指定されたタイムアウト値を経過するまで繰り返し検索します。タイムアウト値が指定されていない場合は、最初の検索でオブジェクトが見つからないとすぐに <i>false</i> が返されます。(TestObject から継承されました)
Exists	オブジェクトがテスト対象アプリケーション内に存在するかどうかを確認します。(TestObject から継承されました)
Find	XPath ロケーターで指定したオブジェクトを検索します。(TestObject から継承されました)
FindAll	XPath ロケーターで指定したすべてのオブジェクトを検索します。(TestObject から継承されました)
GenerateLocator	このオブジェクトのロケーターを返します。(TestObject から継承されました)
GetChildren	このオブジェクトの子オブジェクトを返します。(TestObject から継承されました)
GetCssStyle	指定したスタイル名の算出 CSS スタイルを取得します。(DomElement から継承されました)
GetDomAttribute	オブジェクトに指定された DOM 属性の値を取得します。(DomElement から継承されました)
GetDomAttributeList	この要素に対して利用可能なすべての DOM 属性の名前を返します。このメソッドはモバイル Web アプリケーションではサポートされていません。(DomElement から継承されました)
GetDynamicMethodList	この TestObject に対して動的に呼び出すことのできるメソッド(シングネチャを含む) のリストを返します。(TestObject から継承されました)
GetHtmlRect	BrowserWindow に相対的な DomElement の矩形領域を取得します。GetRect 関数によって提供される画面座標と異なり、GetHtmlRect によって提供される座標は、モバイル ブラウザーの拡大レベルの影響を受けません。Click 関数もすべての DomElements に対する HTML ピクセル座標を使用します。ブラウザー内のすべての位置計算には GetHtmlRect を使用することをお勧めします。これによって、スクリプトがクロスブラウザー対応になり、位置計算はモバイル ブラウザーの拡大レベルの変更の影響を受けなくなります。(DomElement から継承されました)
GetParent	テスト アプリケーション内でこのオブジェクトの親を検索します。(TestObject から継承されました)
GetProperty	指定したプロパティの値を返します。(TestObject から継承されました)
GetPropertyList	指定したオブジェクトに対して取得できるプロパティ名のリストを返します。(TestObject から継承されました)
GetRect	このオブジェクトの大きさや位置を返します。デフォルトでは、返される座標は、このオブジェクトを含む親ウィンドウに対する相対値となります。(TestObject から継承されました)

名前	説明
GetText	DOM 要素の表示されるテキストを取得します。 (DomElement から継承されました)
Highlight	DOM 要素をハイライトします。 (DomElement から継承されました)
HighlightObject	このオブジェクトをハイライトします。 (TestObject から継承されました)
ImageClick	指定したイメージ資産をクリックします。 (TestObject から継承されました)
ImageClickFile	指定したイメージをクリックします。 (TestObject から継承されました)
ImageExists	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageExistsFile	指定したイメージが存在するかどうかを返します。 (TestObject から継承されました)
ImageRectangle	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
ImageRectangleFile	指定したイメージのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
Invoke	テスト オブジェクトのメソッドを動的に呼び出します。 (TestObject から継承されました)
InvokeMethods	この TestObject から始まる一連のメソッドを動的に呼び出します。 (TestObject から継承されました)
MouseMove	オブジェクト上の指定した場所にポインターを移動します。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
PressKeys	一連のキーまたはマウス ボタンを押します (離しません)。 (IKeyable から継承されました)
PressMouse	オブジェクト上でマウス ボタンを押します (離しません)。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ReleaseKeys	一連のキーまたはマウス ボタンを離します。 (IKeyable から継承されました)
ReleaseMouse	マウス ボタンを離します。 このメソッドはモバイル Web アプリケーションではサポートされていません。 (IClickable から継承されました)
ScrollIntoView	ブラウザー ウィンドウの表示領域に DOM 要素をスクロールします。 (DomElement から継承されました)
SetDomAttribute	オブジェクトに指定された DOM 属性の値を設定します。 (DomElement から継承されました)
SetFocus	コントロールにフォーカスを与えます。 (BrowserObject から継承されました)

名前	説明
 SetProperty 	指定したプロパティの値を設定します。 (TestObject から継承されました)
 SetText 	テキストを設定します。
 TextCapture 	このオブジェクトの表示領域内のテキストを返します。 (TestObject から継承されました)
 TextClick 	指定したテキストの中央をクリックします。 (TestObject から継承されました)
 TextExists 	指定したテキストが存在するかどうかを返します。 (TestObject から継承されました)
 TextRectangle 	指定したテキストのオブジェクトに相対的な矩形領域を返します。 (TestObject から継承されました)
 TypeKeys 	一連のキーストロークをオブジェクトに送信します。 (IKeyable から継承されました)
 TypePasswordKeys 	暗号化されたパスワードをテキスト フィールドなどのオブジェクトに入力します。 (IKeyable から継承されました)
 Unhighlight 	元的前景色と背景色に戻します。 (DomElement から継承されました)
 Verify 	この UI オブジェクトのコンテキストで、指定した資産に対する検証を実行します。 (TestObject から継承されました)
 WaitForChildDisappearance 	'locator' パラメーターで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
 WaitForDisappearance 	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)
 WaitForObject 	指定したロケーターに一致するオブジェクトを待機します。 タイムアウト内に一致するオブジェクトが見つからない場合は、ObjectNotFoundException がスローされます。 デフォルトのタイムアウトは 5 秒で、オプション OPT_WAIT_RESOLVE_OBJDEF の値を設定してタイムアウト値を変更できます。 結果を表示する前にトランザクションが処理される場合など、AUT が特定のオブジェクトを表示するのに長時間かかる場合は、WaitForObject を使用してください。 デフォルトでは組み込みの同期が機能するため、UI の操作に対して WaitForObject は必要ありません。 通常のスクリプト実行中に不規則にタイムアウト エラーが発生する場合、WaitForObject ステートメントをスクリプトに追加するのではなく、デフォルトのタイムアウト値を増やすことを検討してください。 (TestObject から継承されました)
 WaitForProperty 	'propertyName' パラメーターで指定したプロパティが 'expectedValue' パラメーターで指定した値を返すまで、またはタイ

名前	説明
	<p>ムアウトに到達するまで待機します。 オプションの 'timeout' パラメーターを使用してタイムアウト値を明示的に指定しない場合は、オプション OPT_WAIT_RESOLVE_OBJDEF によって指定されたタイムアウト値が使用されます。 タイムアウト値に到達すると例外がスローされます。 (TestObject から継承されました)</p>

索引

記号

.NET スクリプト

- バージョン、比較 240
- 比較 240
- コア クラス 239
- 引数のエクスポート、eCATT スクリプト 470
- 引数を eCATT スクリプトからインポートする 469

.NET のサポート

- Silverlight 449
- Windows Forms の概要 436
- Windows Presentation Foundation (WPF) の概要 441
- 概要 435

Apple Safari のテスト

- Information Service、インストール 411, 486

外部データ

- テスト、使用 137

キーワード駆動テスト

- アクティブ データの使用 137
- Silk Central キーワードの実装 532
- アプリケーション構成 521
- インポートとエクスポート 309
- 概要 517
- キーワード レコメンド、アルゴリズム 527
- キーワード、検索 535
- キーワードのアップロード、Silk Central 533
- キーワードの削除 524
- キーワードの実装 521
- キーワードの追加 524
- 基本状態 521
- 記録 519
- クラス リファレンス 1784
- 作成 519
- 作成、アセット ブラウザ 304
- テスト メソッドの指定 523
- 名前の変更 305
- 並べ替え、アセット ブラウザ 307
- パラメータ、例 528
- ビジュアル テストの指定 522
- 開く 304
- フィルタリング、アセット ブラウザ 306
- 編集 524
- 変数を指定した実行 530
- 利点 517

既存のスクリプトの実行

- iOS 9.3 416

機能

- iOS 337, 415

サイズの指定

- ブラウザー ウィンドウ 499

詳細設定

- オプション 581

スクリプトの実行

- 停止 276

制限事項

- モバイル Web アプリケーション 430

Apple Safari 487

Google Chrome 491

Google Chrome、macOS 493

Microsoft Edge 497

Mozilla Firefox 495, 496

Windows 10 508

ネイティブ モバイル アプリケーション 431

セキュアな接続

Information Service 337, 415

前提条件

Android、ネイティブ モバイル アプリケーション 398

Android、モバイル Web アプリケーション 398

Apple Safari 485

Google Chrome 490

iOS、ネイティブ モバイル アプリケーション 405

iOS、モバイル Web アプリケーション 405

属性除外リスト

MSUIA、設定 565, 576

UI オートメーション、設定 565, 576

設定 566

属性のタイプ

UI オートメーション 249, 511

停止

スクリプトの実行 276

データの追加

テスト 137

テキスト解決

概要 252

テキスト クリックの記録

概要 252

デバイスが接続されていません

モバイル 424

変更の表示

.NET スクリプト 240

.NET スクリプト、バージョン 240

資産 304

資産、バージョン 304

オブジェクト マップ 347

オブジェクト マップ、バージョン 347

ビジュアル テスト 217

ビジュアル テスト、バージョン 217

ポートの構成

Information Service、クライアント 336

Open Agent 335

メソッドの動的呼び出し

Apache Flex 366

Apache Flex カスタム コントロール 370

Java AWT 392, 396

Java Swing 392, 396

Java SWT 392, 396

SAP 457

SAP コントロール 459

Silverlight 451

UI オートメーション 247

Windows Forms 437

Windows Presentation Foundation (WPF) 444

モバイル Web アプリケーション

Android、前提条件 398
iOS、前提条件 405
制限事項 430
呼び出し
Android 433
iOS 433
UI オートメーション 247

A

AbstractButton クラス 2232
Access
データベース、コンパクト化 602
データベースのレコードのロック解除 601
データベースバージョンの更新 600
データベースを開く 600
AccessibleControl クラス 2658
ActiveData クラス 798
ActiveDataRow クラス 803
AddColumn メソッド
ActiveData 798
AddRow メソッド
ActiveData 799
Adobe Flex
Adobe AIR のサポート 376
automationName プロパティ 384
FlexDataGrid コントロール 377
Select メソッド、概要 376
Select メソッド、設定 386
アプリケーションの作成 382
構成情報の追加 380
コンテナ 387
コンテナのコーディング 387
実行時のパラメーター渡し 381
実行時の読み込み 379
実行時読み込み 380
実行前のパラメーター渡し 381
セキュリティ設定 390
テストの再生 389
パラメータを渡す 381
複数ビュー コンテナ 388
メソッドの呼び出し 366
Agent クラス
コア クラス 808
AJAX アプリケーション
記録オプション、設定 566
スクリプトのハング 504
Android
メソッドの呼び出し 433
モバイル Web アプリケーション、前提条件 398
モバイル ネイティブ アプリケーション、前提条件 398
USB デバッグの有効化 400
USB ドライバのインストール 400
インストール済みアプリ、テスト 417
エミュレータを設定する 401
推奨設定 401
テスト 398
デバイスの開放 423
デバイスの開放、記録 423
デバイスの開放、再生 423

トラブルシューティング 424
ハイブリッド アプリケーション 399
並列テスト、テスト済みの構成 402
Android エミュレータ
設定 401
Apache Flex
automationIndex プロパティ 383
automationName プロパティ 383
Component Explorer 365
Flash Player 設定 364
Select メソッド、設定 386
アプリケーションの事前コンパイル 378
アプリケーションの有効化 377
オートメーション パッケージのリンク 378
概要 364
カスタム コントロール 257, 365
カスタム コントロール、実装 371
カスタム コントロール、定義 367, 374, 383
カスタム コントロールのメソッドの呼び出し 370
記録、アプリケーション 388
クラス定義ファイル 374, 383
コントロールが認識されない 391
初期化、アプリケーション 388
スクリプトのカスタマイズ 375
スタイル 389
属性 390, 509
テスト 365
複数のアプリケーションのテスト 375
メソッドの呼び出し 366
ワークフロー 388
Apache Flex アプリケーション
カスタム属性 321, 383
API 再生
比較、ネイティブ再生 479
Apple Safari
制限事項 487
Information Service、インストール 411, 416, 486, 489
サポート 473
準備 486
接続文字列 482
前提条件 485
テスト 485
複数のテストの実行 489
Applet クラス 2236
AppletContainer クラス 2240
ArgumentAttribute クラス
キーワード駆動テスト クラス 1784
Attach メソッド
Agent 809
AWTButton クラス 2244
AWTCanvas クラス 2248
AWTCheckbox クラス 2252
AWTCheckboxMenuItem クラス 2256
AWTChoice クラス 2260
AWTComponent クラス 2264
AWTContainer クラス 2268
AWTDialog クラス 2272
AWTFrame クラス 2277
AWTHorizontalScrollbar クラス 2282
AWTLabel クラス 2286

AWTList クラス 2290
AWTMenu クラス 2295
AWTMenuItem クラス 2301
AWTRadioButton クラス 2305
AWTScrollbar クラス 2309
AWTScrollPane クラス 2313
AWTTextArea クラス 2318
AWTTextComponent クラス 2323
AWTTextField クラス 2327
AWTVerticalScrollbar クラス 2332
AWTWindow クラス 2336

B

BaseGuiTestObject クラス 650
BaseState クラス
 コア クラス 814, 818
BasicArrowButton クラス 2341
BLOB
 eCATT 464
BLOB でデータを転送する
 スタンドアロン Silk Test Workbench を構成する 461
BrowserApplication クラス 2735
BrowserObject クラス 2741
BrowserWindow クラス 2744

C

CBanner クラス 1680
CEF
 テスト 472
CharSet 列挙 654
CheckBox クラス 654
CheckBoxToolItem クラス 658
CheckedListBox クラス 2691
Chrome
 拡張、テスト 491
 既知の問題 642
 クロス ブラウザー スクリプト 505
 構成設定 480
 前提条件 490
 テスト 489
 ユーザー データ ディレクトリ、テスト 491
 ロケーター 505
Chrome for Android
 サポート 473
 ブラウザの種類、設定 430
Chromium Embedded Framework
 テスト 472
ClickType
 データ型 836
CloseAll メソッド
 ConsoleWindow 820
Close メソッド
 ConsoleWindow 819
Color
 データ型 836
ColumnCount プロパティ
 ActiveData 802
ComboBox クラス 662
CommandLineArguments プロパティ
 BaseState 815

CommandLinePattern プロパティ
 BaseState 816
Common プロジェクト
 資産、接頭部の追加 559
 説明 36, 604
Component Explorer
 Apache Flex 365
Configuration Assistant
 自動サイン 412
Connect メソッド
 Agent 810
ConsoleWindow クラス
 コア クラス 819
Control クラス 666
CoolBar クラス 1684
CoolItem クラス 1688
Count プロパティ
 ActiveData 803, 807
CTabFolder クラス 1692
CTabItem クラス 1696
Customer Care 78, 79

D

DataGrid クラス 2695
DataGridColumn クラス 2699
DataGridItem クラス 2703
DataGridRow クラス 2707
Decrypt メソッド
 Agent 811
Desktop クラス 671
Desktop プロパティ
 Agent 814
DetachAll メソッド
 Agent 811
Dialog クラス 671
Dispose メソッド
 ActiveData 800
 ActiveDataRow 804
dll
 Visual Basic からの呼び出し 240
 Visual Basic スクリプトからの呼び出し 241
 関数の宣言構文 241
 関数への引数の受け渡し 242
 関数への文字列引数の受け渡し 244
 規則の変更 244
 名前のエイリアス設定 244
 変更可能な引数の関数への受け渡し 243
 呼び出しの例 241
DllCall クラス
 コア クラス 823
DLL の呼び出し
 Visual Basic 240
 Visual Basic スクリプト 241
 例 241
DLL ファイル
 スクリプトでの参照 238
DomainUpDown クラス 2711
DomButton クラス 2748
DomCheckBox クラス 2753
DomElement クラス 2757
DomEmbeddedElement クラス 2762

DomForm クラス 2766
DomLink クラス 2771
DomListBox クラス 2775
DomRadioButton クラス 2780
DomTable クラス 2784
DomTableRow クラス 2789
DomTextField クラス 2793
DropDownToolItem クラス 676
DSN

Access 594
Oracle 596
SQL Server 595
概要 594

DynamicInvoke
Android 433
Apache Flex 366
iOS 433
Java AWT 392, 396
Java Swing 392, 396
SAP 457
Silverlight 451
UI オートメーション 247
Windows Forms 437
Windows Presentation Foundation (WPF) 444

E

eCATT

BLOB 464
BLOB 追跡情報の入力 464
BLOB の内容の入力または変更 465
eCATT UI からスクリプトを記録する 463
eCATT UI からスクリプトをテストする 463
eCATT UI からスクリプトを記録/再生する 463
eCATT スクリプトから .NET スクリプトへの引数のインポート 469
eCATT スクリプトからビジュアル テストへの引数のインポート 468
eCATT スクリプト引数の作成 466
eCATT スクリプト引数のサポート 466
eCATT でのテスト スクリプト属性の指定 463
RFC 接続パラメータの設定または変更 462
SAP/R3 ログイン パラメータの設定 461
Silk Test Workbench テスト結果 471
Silk Test Workbench テストを実行する 470
Silk Test Workbench と統合する 460
Silk Test Workbench を登録する 460
概要 459
スクリプト引数の編集 467
スタンドアロン Silk Test Workbench を使用した eCATT スクリプトの作成 463
テスト スクリプトの定義 463
統合オプション 557
統合のセットアップ 460
引数コンテナからの引数の削除 467
引数のエクスポート、.NET スクリプトから eCATT スクリプト 470
ビジュアル テストでの eCATT スクリプトへの引数のエクスポート 469
ユーザー アカウントを作成する 461
eCATT と統合する

概要 460
ECCUST_ET
Silk Test Workbench を登録する 460

Eclipse
トラブルシューティング 397

Eclipse RCP
サポート 394

Edge
既知の問題 644
制限事項 497
接続文字列 482
テスト 497
リモートテスト 482
ロケーター 505

Elapsed プロパティ
Timer 827
ElapsedMilliseconds プロパティ
Timer 828

ElementHost クラス 2715

Encrypt メソッド
Agent 811

Executable プロパティ
BaseState 816
ExecutablePattern プロパティ
BaseState 816

Execute メソッド
BaseState 824

ExecuteBaseState メソッド
Agent 812

ExecutionMode 列挙 793
ExecutionResult クラス 793

ExpandBar クラス 1700
ExpandItem クラス 1704
ExternalRowNumber プロパティ
ActiveDataRow 808

F

FAQ

オブジェクト マップ 356
クロス ブラウザ テスト 502
コードの追加、AUT 257
動的呼び出し 255

FileHandle クラス 794

FileInfo クラス 679

FileOpenMode 列挙 794

FilePointerMode 列挙 794

FileShareMode 列挙 795

Firefox

制限事項 495, 496
拡張機能、テスト 495
クロス ブラウザー スクリプト 505
構成設定 480
テスト 493
プロファイル、テスト 494
ロケーター 505

Flash Player
アプリケーションを開く 364
セキュリティ設定 390

Flex
Adobe AIR のサポート 376

automationIndex プロパティ 383
automationName プロパティ 383, 384
Component Explorer 365
Flash Player 設定 364
FlexDataGrid コントロール 377
Select メソッド、概要 376
Select メソッド、設定 386
アプリケーションの作成 382
アプリケーションの事前コンパイル 378
アプリケーションの有効化 377
オートメーション パッケージのリンク 378
概要 364
カスタム コントロール 257, 365
カスタム コントロール、実装 371
カスタム コントロール、定義 367, 374, 383
カスタム コントロールのメソッドの呼び出し 370
記録、アプリケーション 388
クラス定義ファイル 374, 383
構成情報の追加 380
コンテナ 387
実行時のパラメーター渡し 381
実行時の読み込み 379
実行時読み込み 380
実行前のパラメーター渡し 381
初期化、アプリケーション 388
スクリプトのカスタマイズ 375
スタイル 389
セキュリティ設定 390
属性 390, 509
テスト 365
テストの再生 389
パラメータを渡す 381
複数のアプリケーションのテスト 375
複数ビュー コンテナ 388
メソッドの呼び出し 366
ワークフロー 388
Flex クラス リファレンス 842
FlexAccordion クラス 842
FlexAdvancedDataGrid クラス 849
FlexAlert クラス 858
FlexApplication クラス 865
FlexAreaChart クラス 872
FlexAreaSeries クラス 879
FlexAxisRenderer クラス 885
FlexBarChart クラス 891
FlexBarSeries クラス 898
FlexBox クラス 904
FlexBubbleChart クラス 910
FlexBubbleSeries クラス 917
FlexButton クラス 923
FlexButtonBar クラス 929
FlexCandlestickChart クラス 936
FlexCandlestickSeries クラス 943
FlexCanvas クラス 949
FlexCartesianChart クラス 956
FlexChart クラス 963
FlexChartLegend クラス 970
FlexChartSeries クラス 975
FlexCheckBox クラス 981
FlexColorPicker クラス 988
FlexColumnChart クラス 994
FlexColumnSeries クラス 1001
FlexComboBase クラス 1007
FlexComboBox クラス 1014
FlexContainer クラス 1021
FlexContainerMovieClip クラス 1027
FlexDataGrid クラス 1032
FlexDateChooser クラス 1041
FlexDateField クラス 1047
FlexDisplayObject クラス 1054
FlexDividedBox クラス 1058
FlexForm クラス 1065
FlexFormItem クラス 1072
FlexHLOCChart クラス 1078
FlexHLOCSeries クラス 1085
FlexHLOCSeriesBase クラス 1091
FlexImage クラス 1097
FlexLabel クラス 1103
FlexLineChart クラス 1108
FlexLineSeries クラス 1115
FlexLinkBar クラス 1121
FlexList クラス 1128
FlexListBase クラス 1136
FlexListLabel クラス 1143
FlexLoader クラス 1147
FlexMenu クラス 1152
FlexMenuBar クラス 1158
FlexNavigationBar クラス 1164
FlexNumericStepper クラス 1171
FlexObject クラス 1177
FlexOLAPDataGrid クラス 1182
FlexPanel クラス 1192
FlexPieChart クラス 1198
FlexPieSeries クラス 1205
FlexPlotChart クラス 1211
FlexPlotSeries クラス 1218
FlexPopUpButton クラス 1224
FlexProgressBar クラス 1231
FlexRadioButton クラス 1237
FlexRepeater クラス 1243
FlexRule クラス 1247
FlexScrollBar クラス 1252
FlexScrollBase クラス 1257
FlexSlider クラス 1263
FlexStandalonePlayer クラス 1269
FlexTabNavigator クラス 1273
FlexTextArea クラス 1280
FlexTitleWindow クラス 1286
FlexToggleButtonBar クラス 1293
FlexTree クラス 1300
FlexUIMovieClip クラス 1308
FlexVideoDisplay クラス 1313
FlexViewStack クラス 1318
FlexWindow クラス 1325
FlexWindowedApplication クラス 1332
FormatDate 関数
 式デザイン 184
FormatTime 関数
 式デザイン 186
FormsHost クラス 2719
FormsWindow クラス 2723

G

GetContents メソッド
 ConsoleWindow 821
GetDouble メソッド
 ActiveDataRow 804
GetEnumerator メソッド
 ActiveData 800
GetLong メソッド
 ActiveDataRow 804
GetLongLong メソッド
 ActiveDataRow 805
GetOption メソッド
 Agent 812
GetString メソッド
 ActiveDataRow 805
Google Chrome
 オプション、設定 483
 ケイパビリティ、設定 483
 制限事項 491
 制限事項、macOS 493
 macOS 483
 拡張、テスト 491
 既知の問題 642
 構成設定 480
 サポート 473
 全画面表示 502
 前提条件 490
 追加のバージョン、テスト 501
 テスト 489
 ユーザー データ ディレクトリ、テスト 491
 リモート テスト 482
Group クラス 680
GUI
 概要 40
 メイン画面 40
GuiTestObject クラス 684
GWT
 コントロールの検索 320

H

Header クラス 2662
HorizontalSash クラス 1707
HorizontalScrollBar クラス 688
HTTPS
 証明書、置換 338
 Information Service 337, 415

I

IAWTScrollable インターフェイス 2345
IAWTScroller インターフェイス 2345
IBaseScrollable インターフェイス 2346
IBaseState
 インターフェイス 824
IClickable インターフェイス 692
IFocusable インターフェイス 693
IKeyable インターフェイス 693
image assets
 creating 358
IME
 UI オートメーション 251

IMobileClickable インターフェイス 1786
IMobileGestures インターフェイス 1786
IMobileKeyable インターフェイス 1787
IMoveable インターフェイス 693
INativeWindow インターフェイス 694
Information Service
 証明書、置換 338
 編集 337, 415
 ポート、設定 334
 ポートの構成、クライアント 336
 HTTPS 337, 415
 Mac、インストール 411, 416, 486, 489
innerHTML
 xBrowser 504
innerText
 xBrowser 504
IntelliSense
 使用法 114
Internet Explorer
 link.select のフォーカスの問題 504
 既知の問題 643
 クロス ブラウザー スクリプト 505
 構成設定 480
 サポート 473
 スクリプト例の起動 120
 全画面表示 502
 例、新しいブラウザ ウィンドウ 118
 ロケーター 505
Internet Explorer 10
 予期しない Click 動作 506
invoke
 Java AWT 392, 396
 Java SWT 392, 396
 SAP 457
 Silverlight 451
 Swing 392, 396
 Windows Forms 437
 Windows Presentation Foundation (WPF) 444
InvokeMethods
 Android 433
 Apache Flex 366
 iOS 433
 Java AWT 392, 396
 Java Swing 392, 396
 SAP 457
 Silverlight 451
 UI オートメーション 247
 Windows Forms 437
 Windows Presentation Foundation (WPF) 444
Invoke メソッド
 呼び出し可能なメソッド 255
IOracleFormsMenuBase インターフェイス 2346
IOracleFormsScrollable インターフェイス 2347
IOracleFormsScroller インターフェイス 2348
iOS
 デバイス、準備 410
 メソッドの呼び出し 433
 モバイル Web アプリケーション、前提条件 405
 モバイル ネイティブ アプリケーション、前提条件 405

Information Service、インストール 411, 416, 486, 489
Mac、準備 412
Web アプリ、シミュレータ 408
Web アプリ、テスト 408
アプリ、テストの準備 410
インストール済みアプリ、テスト 417
推奨設定 416
テスト 404
テスト、デベロッパ アカウントなし 413
デバイスの開放 423
デバイスの開放、記録 423
デバイスの開放、再生 423
ネイティブ アプリ、シミュレータ 407
ネイティブ アプリ、テスト 406
ハイブリッド アプリケーション 409
iOS 9.3
既存のスク립ト、実行 416
ISapContextMenuable インターフェイス 1832
IScrollable インターフェイス 694
IsRunning プロパティ
Timer 828
Item クラス 695
ItemIdentifier
データ型 837
ItemPath
データ型 838
Item メソッド
ActiveData 800

J

Java AWT
概要 391
属性 392, 509
属性の種類 392, 509
メソッドの呼び出し 392, 396
Java AWT および Swing クラス リファレンス 2232
Java FX
サポート 246
Java Network Launching Protocol (JNLP)
スク립トの記録 117
ビジュアル テスト 85
Java Swing
概要 391
属性 392, 509
メソッドの呼び出し 392, 396
Java SWT
サポート 394
属性の種類 395, 509
トラブルシューティング 397
メソッドの呼び出し 392, 396
Java SWT クラス リファレンス 395, 1680
Java AWT/Swing
priorlabel 393
JButton クラス 2348
JCheckBox クラス 2352
JCheckBoxMenuItem クラス 2357
JColorChooser クラス 2361
JComboBox クラス 2365
JComponent クラス 2369
JDesktopPane クラス 2374

JDialog クラス 2378
JEditorPane クラス 2382
JFrame クラス 2387
JHorizontalScrollBar クラス 2392
JLabel クラス 2396
JLayeredPane クラス 2400
JList クラス 2404
JMenu クラス 2409
JMenuBar クラス 2413
JMenuItem クラス 2417
JNLP
スク립トの記録 117
ビジュアル テスト 85
JPanel クラス 2421
JPasswordField クラス 2425
JPopupMenu クラス 2430
JProgressBar クラス 2434
JRadioButton クラス 2438
JRadioButtonMenuItem クラス 2442
JRootPane クラス 2446
JScrollBar クラス 2451
JScrollPane クラス 2455
JSlider クラス 2459
JSpinner クラス 2463
JSplitPane クラス 2468
JTabbedPane クラス 2472
JTable クラス 2476
JTableHeader クラス 2481
JTextArea クラス 2485
JTextComponent クラス 2490
JTextField クラス 2494
JTextPane クラス 2499
JToggleButton クラス 2503
JToolBar クラス 2508
JTree クラス 2512
JVerticalScrollBar クラス 2516
JViewport クラス 2521
JWindow クラス 2525

K

KeywordAttribute クラス
キーワード駆動テスト クラス 1785
KeywordGroupAttribute クラス
キーワード駆動テスト クラス 1785

L

Label クラス 699
Link クラス 703
ListBox クラス 706
ListView クラス 2666
LoadActiveData メソッド
Workbench 829
LoadAssembly
アセンブリをコピーできない 257
Locator プロパティ
BaseState 817

M

Mac

- Apple Safari、前提条件 485
- Apple Safari、テスト 485
- Information Service、インストール 411, 416, 486, 489

Menu クラス 711

MenuItem クラス 714

MenuStrip クラス 2727

MFC

- サポート 473

Microsoft Edge

- 既知の問題 644
- サポートするバージョン、新しい 36
- 制限事項 497
- サポート 473
- 接続文字列 482
- 追加のバージョン、テスト 501
- テスト 497
- リモート テスト 482

Microsoft Foundation Class

- サポート 473

Microsoft ユーザー補助

- オブジェクト解決の向上 245

Mobile Center

- 有効化 418

Mobile クラス リファレンス 1786

MobileButton クラス 1787

MobileDevice クラス 1791

MobileObject クラス 1796

MobileTextField クラス 1800

MobileWindow クラス 1804

ModifierKeys

- データ型 838

MonthCalendar クラス 2671

MouseButton

- データ型 839

Mozilla Firefox

- オプション、設定 483
- ケイパビリティ、設定 483
- サポートするバージョン、新しい 35
- 制限事項 495, 496
- macOS 483
- 拡張機能、テスト 495
- 既知の問題 645
- 構成設定 480
- サポート 473
- 全画面表示 502
- 追加のバージョン、テスト 501
- テスト 493
- プロファイル、テスト 494
- リモート テスト 482

MSUIA

- オブジェクト解決、改善 246
- オプション、設定 565, 576
- メソッドの呼び出し 247
- 有効化 565, 576

N

Name プロパティ

- Timer 828

NumericUpDown クラス 2731

O

Open Agent

- 接続ポート、構成 335
- ポートの構成、Information Service 336
- ポートの設定、リモート エージェント 334
- ポート番号 334
- オプション 325
- 概要 325
- 起動 325
- スクリプトから開始 325
- スクリプトから停止 325
- テスト、リモート 339

Open Agent の開始

- スクリプト 325

Open Agent の停止

- スクリプト 325

Open メソッド

- ConsoleWindow 821

OPT_ENABLE_ACCESSIBILITY

- オプション 581

OPT_ENABLE_EMBEDDED_CHROME_SUPPORT

- オプション 472

OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT

- オプション 581

OPT_IMAGE_ASSET_DEFAULT_ACCURACY

- オプション 581

OPT_IMAGE_VERIFICATION_DEFAULT_ACCURACY

- オプション 581

OPT_LOCATOR_ATTRIBUTES_CASE_SENSITIVE

- オプション 581

OPT_REMOVE_FOCUS_ON_CAPTURE_TEXT

- オプション 581

Oracle

- ORA_DBA グループの作成 590
- Oracle 認証の設定 589
- OS 認証 588
- 概要 587
- クライアント接続用のデータベースの準備 590
- クライアントの設定 590
- データベース、コピー 601
- データベースの準備 593
- データベースの新規作成 589
- データベースのレコードのロック解除 601
- データベース バージョンの更新 600
- データベースを開く 599
- ドメインなしでの設定 593, 594
- 認証方式 587
- パスワード認証スキーマ 592
- ユーザー、設定 588
- ユーザーに対するサーバー アクセスの作成 591
- ユーザーの設定 588, 591

Oracle Forms

- 既知の問題 647
- サポートされる Java バージョン 647
- サポートするバージョン 393
- 前提条件 394
- 属性 394
- について 393

OracleFormsApplication クラス 2529

OracleFormsButton クラス 2533
OracleFormsCheckbox クラス 2537
OracleFormsChoice クラス 2541
OracleFormsComboBox クラス 2546
OracleFormsContainer クラス 2550
OracleFormsHorizontalScrollbar クラス 2554
OracleFormsLabel クラス 2559
OracleFormsListBox クラス 2563
OracleFormsListView クラス 2568
OracleFormsMenu クラス 2574
OracleFormsMenuItem クラス 2578
OracleFormsPopList クラス 2582
OracleFormsPopupMenu クラス 2586
OracleFormsRadioButton クラス 2590
OracleFormsScrollbar クラス 2595
OracleFormsStatusArea クラス 2599
OracleFormsStatusBarItem クラス 2603
OracleFormsStatusIndicator クラス 2611
OracleFormsTabBar クラス 2615
OracleFormsTabBarItem クラス 2619
OracleFormsTabPanel クラス 2623
OracleFormsTextField クラス 2627
OracleFormsTitleBar クラス 2633
OracleFormsToolBar クラス 2637
OracleFormsToolBarItem クラス 2641
OracleFormsTree クラス 2645
OracleFormsVerticalScrollbar クラス 2650

P

Pager クラス 2675
Pause メソッド
 Timer 826
Point
 データ型 839
priorlabel
 Java AWT/Swing テクノロジ ドメイン 393
priorLabel
 Win32 テクノロジ ドメイン 472
Private 変数
 スクリプト 220
ProgressBar クラス 2679
Public 変数
 スクリプト 220
PushButton クラス 718
PushToolItem クラス 722

Q

QT
 サポート 246

R

RadioList クラス 725
RadioListToolItem クラス 729
Range
 データ型 839
Rectangle

 データ型 840
RegistryCategory 列挙 733
RegistryView 列挙 734
RemoveColumn メソッド
 ActiveData 801
RemoveRow メソッド
 ActiveData 801
RenameColumn メソッド
 ActiveData 801
Reset メソッド
 ActiveData 802
ResetOptions メソッド
 Agent 812
ResultComment メソッド
 Workbench 835
Resume メソッド
 Timer 826
RowNumber プロパティ
 ActiveDataRow 808
Rumba
 RumbaKey クラス 1808
 Unix ディスプレイ 456
 画面検証の使用 456
 キー ニーモニック 1808
 サポートの有効化と無効化 455
 について 454
 ロケーター属性 455, 512
Rumba クラス リファレンス 1807
Rumba ロケーター属性
 コントロールの識別 455, 512
RumbaCharacterAttribute
 Rumba データ型 1810
RumbaField クラス 1811
RumbaLabel クラス 1815
RumbaObject クラス 1819
RumbaScreen クラス 1823
RumbaTextField クラス 1827
Rumba データ型
 RumbaCharacterAttribute 1810
RunScript メソッド
 Workbench 832

S

Safari
 制限事項 487
 準備 486
 接続文字列 482
 前提条件 485
 テスト 485
 複数のテストの実行 489
SAP
 概要 456
 既知の問題 645
 クラス リファレンス 457
 セキュリティ設定 459
 属性の種類 457, 510
 メソッドの呼び出し 457
SAP クラス リファレンス 1832
SapBarChart クラス 1832
SapBox クラス 1836
SapButton クラス 1840

SapCalendar クラス 1844
SapChart クラス 1848
SapCheckBox クラス 1852
SapColorSelector クラス 1856
SapComboBox クラス 1860
SapComponent クラス 1864
SapContainer クラス 1868
SapContainerShell クラス 1872
SapContextMenu クラス 1876
SapCustomControl クラス 1879
SapDockShell クラス 1883
SapGridView クラス 1887
SapHorizontalScrollBar クラス 1894
SapHTMLViewer クラス 1897
SapLabel クラス 1901
SapMenu クラス 1905
SapMenuBar クラス 1909
SapNetPlan クラス 1913
SapOfficeIntegration クラス 1917
SapOkCodeField クラス 1921
SapPicture クラス 1925
SapRadioButton クラス 1929
SapScrollbar クラス 1933
SapScrollContainer クラス 1936
SapShell クラス 1940
SapSimpleContainer クラス 1944
SapSplitterContainer クラス 1947
SapStatusBar クラス 1951
SapTab クラス 1955
SapTable クラス 1959
SapTabStrip クラス 1963
SapTextEdit クラス 1967
SapTextField クラス 1971
SapTitlebar クラス 1975
SapToolBar クラス 1979
SapToolBarControl クラス 1983
SapTree クラス 1987
SapUserArea クラス 1994
SapVerticalScrollBar クラス 1998
SapWindow クラス 2001
SAP コントロール
 メソッドの動的呼び出し 459
Sash クラス 1711
SashForm クラス 1715
Sauce Labs
 有効化 419
Save メソッド
 ActiveData 802
Scale クラス 734
ScrollableControl クラス 1719
ScrollBar クラス 738
ScrolledComposite クラス 1723
Select メソッド
 Apache Flex、設定 386
SeparatorItem クラス 742
SetDouble メソッド
 ActiveDataRow 806
SetLong メソッド
 ActiveDataRow 806
SetLongLong メソッド
 ActiveDataRow 807
SetOption メソッド
 Agent 813
SetString メソッド
 ActiveDataRow 807
Shell クラス 1727
Shutdown メソッド
 Agent 813
Silk Central
 Mobile Center、有効化 418
 Sauce Labs、有効化 419
 キーワードのアップロード 533
 場所の設定 531
 パラメータ 206
Silk Central キーワード
 実装 532
Silk Performer
 実行時間の計測 262
Silk Test Workbench
 機能 30
 基本機能 27
 詳細 26
Silverlight
 概要 449
 クラス リファレンス 450
 サポート 449
 スクロール 453
 属性の種類 450, 510
 トラブルシューティング 454
 メソッドの呼び出し 451
 ロケータ属性 450, 510
Silverlight クラス リファレンス 2006
SLApplication クラス 2006
SLAutoCompleteBox クラス 2009
SLBase クラス 2014
SLButton クラス 2018
SLCalendar クラス 2022
SLCalendarButton クラス 2026
SLCalendarDayButton クラス 2030
SLCheckBox クラス 2034
SLComboBox クラス 2038
SLComboBoxItem クラス 2042
SLDataGrid クラス 2046
SLDataGridCell クラス 2050
SLDataGridDetails クラス 2054
SLDataGridRow クラス 2058
SLDataPager クラス 2062
SLDatePicker クラス 2065
SLDescriptionViewer クラス 2069
SLFrame クラス 2073
SLGridSplitter クラス 2077
SLGroup クラス 2080
SLHeader クラス 2084
SLHeaderItem クラス 2088
SLHorizontalScrollBar クラス 2091
SLHyperlinkButton クラス 2095
SLImage クラス 2099
SLListBox クラス 2103
SLListItem クラス 2107
SLMediaElement クラス 2111
SLMenu クラス 2115
SLMenuBar クラス 2118

SLMenuItem クラス 2122
SLMultiScaleImage クラス 2126
SLPane クラス 2129
SLPasswordBox クラス 2133
SLPopup クラス 2137
SLProgressBar クラス 2141
SLRadioButton クラス 2145
SLRepeatButton クラス 2148
SLRichTextBox クラス 2152
SLSeparator クラス 2156
SLSlider クラス 2160
SLSpinner クラス 2164
SLSplitButton クラス 2167
SLStatusBar クラス 2171
SLTabControl クラス 2175
SLTabItem クラス 2179
SLTable クラス 2182
SLTextBlock クラス 2186
SLTextBox クラス 2190
SLThumb クラス 2194
SLTitleBar クラス 2197
SLToggleButton クラス 2201
SLToolBar クラス 2205
SLToolTip クラス 2209
SLTreeView クラス 2212
SLTreeViewItem クラス 2217
SLValidationSummary クラス 2221
SLVerticalScrollBar クラス 2224
SLWindow クラス 2228
SparkAirHTML クラス 1340
SparkApplication クラス 1345
SparkBorderContainer クラス 1351
SparkButton クラス 1357
SparkButtonBar クラス 1363
SparkButtonBarButton クラス 1369
SparkButtonBase クラス 1375
SparkCheckBox クラス 1380
SparkComboBox クラス 1386
SparkComplexDisplay クラス 1392
SparkDataGrid クラス 1399
SparkDataGridLabel クラス 1405
SparkDataGroup クラス 1410
SparkDataRenderer クラス 1417
SparkDropDownList クラス 1424
SparkDropDownListBase クラス 1430
SparkForm クラス 1437
SparkFormItem クラス 1443
SparkGroup クラス 1449
SparkGroupBase クラス 1456
SparkImage クラス 1462
SparkLabel クラス 1468
SparkList クラス 1473
SparkListBase クラス 1480
SparkListLabel クラス 1486
SparkMuteButton クラス 1493
SparkNavigatorContent クラス 1498
SparkNumericStepper クラス 1504
SparkObject クラス 1510
SparkPanel クラス 1515
SparkPopUpAnchor クラス 1521
SparkRadioButton クラス 1527
SparkRange クラス 1532
SparkRichEditableText クラス 1538
SparkRichText クラス 1544
SparkScrollBar クラス 1550
SparkSkinnableContainer クラス 1555
SparkSkinnableContainerBase クラス 1561
SparkSkinnableDataContainer クラス 1567
SparkSkinnablePopUpContainer クラス 1572
SparkSkinnableTextBase クラス 1578
SparkSlider クラス 1584
SparkSpinner クラス 1590
SparkTabBar クラス 1595
SparkTextArea クラス 1601
SparkTextBase クラス 1608
SparkTextInput クラス 1613
SparkTileGroup クラス 1620
SparkTitleWindow クラス 1627
SparkToggleButton クラス 1633
SparkToggleButtonBase クラス 1638
SparkTrackBase クラス 1644
SparkVideoDisplay クラス 1649
SparkVideoPlayer クラス 1655
SparkVolumeBar クラス 1660
SparkWindow クラス 1666
SparkWindowedApplication クラス 1673
Spinner クラス 1731
SplitPaneDivider クラス 2654
SQL Server
 SQL Server データベースの新規作成 585
 管理ユーザー、作成 585
 構成する 584
 データベース、コピー 601
 データベースの準備 587
 データベースのレコードのロック解除 601
 データベース バージョンの更新 600
 データベースを開く 599
 ドメインなしでの設定 593
 ユーザーの設定 586
Start メソッド
 Timer 826
StartNew メソッド
 Timer 827
StatusBar クラス 2683
Stop メソッド
 Timer 827
STW.exe
 ログ 555
STW.EXE
 前提条件、Silk Central 547
 パラメータ 537
 例 547
STW.EXE、前提条件 547
StyledText クラス 1735
SupportLine 78, 79
Swing
 概要 391
 属性 392, 509
 メソッドの呼び出し 392, 396
SWTBrowser クラス 1740
SWTDateTime クラス 1743
SWTTabControl クラス 1747

SWTTabItem クラス 1751
SWTTable クラス 1755
SWTTableColumn クラス 1760
SWTTableRow クラス 1763
SWTTree クラス 1767
SWTTreeColumn クラス 1772
SystemFunctions クラス 795
SystemFunctions プロパティ
Agent 814

T

TabControl クラス 746
Table クラス 750
TableColumn クラス 754
TableRow クラス 758
TechDomains プロパティ
BaseState 817
TestObject クラス 762
textContents
xBrowser 504
TextField クラス 764
TextPosition
データ型 840
TextRange
データ型 841
Timer クラス
コア クラス 825
ToggleButton クラス 768
ToolBar クラス 772
ToolItem クラス 776
Tree クラス 779
TreeContent
データ型 841
TreeNode
データ型 842
TypeKeys メソッド
ConsoleWindow 822

U

UAC
概要 127
UI
概要 40
UI オートメーション
オブジェクト解決、改善 246
オプション、設定 565, 576
制限事項 251
属性のタイプ 249, 511
メソッドの呼び出し 247
有効化 565, 576
ロケータ属性 249, 511
スクロール 250
トラブルシューティング 251
Unicode コンテンツ
サポート 251
Unicode 文字のマッピング 600
Unix ディスプレイ
Rumba 456
UnloadAllDllsFromAgent メソッド

DllCall 823
UnloadDllFromAgent メソッド
DllCall 823
UpDown クラス 2687
URL プロパティ
BrowserBaseState 819
USB ドライバのインストール
Android 400

V

Verify メソッド
Workbench 833
VerifyAsset メソッド
Workbench クラス 835
VerticalSash クラス 1776
VerticalScrollBar クラス 784
ViewForm クラス 1780
Visual Basic
DLL の呼び出し 240
スクリプトからの DLL の呼び出し 241
Visual COBOL
サポートするバージョン 454
について 454

W

Web ページ
キャプチャ、全画面 294
WebSync 78, 79
Web アプリケーション
xBrowser テスト オブジェクト 476
カスタム属性 322, 507
カスタム属性、設定 566
既知の問題 642
記録オプション、設定 566
サポートされている属性 507, 513
設定 574
Win32
priorLabel 472
Win32 クラス リファレンス 471, 2658
Window クラス 788
Windows
属性の種類 441, 513
Windows 10
制限事項 508
Windows API
サポート 471
Windows Forms
概要 436
クラス リファレンス 436
属性の種類 436, 513
メソッドの呼び出し 437
Windows Forms クラス リファレンス 2691
Windows Presentation Foundation (WPF)
WPFItemsControl クラス 443
概要 441
カスタム コントロール 443
クラスの公開 448, 568
クラス リファレンス 441
メソッドの呼び出し 444

ロケータ属性 441, 513
Windows Forms アプリケーション
カスタム属性 323, 436
WinForms アプリケーション
カスタム属性 323, 436
Workbench クラス 829
WorkingDirectory プロパティ
BaseState 817
Works Order 番号 78, 79
WPF

WPFItemsControl クラス 443
カスタム コントロール 443
クラスの公開 448, 568
クラス リファレンス 441
メソッドの呼び出し 444
ロケータ属性 441, 513
WPF アプリケーション
カスタム属性 323, 443
WPF ロケータ属性
コントロールの識別 441, 513

X

xBrowser
Internet Explorer で四角形の位置が正しくない 504
機能の公開 504
認識されないダイアログ 502
xBrowser
Apple Safari 485
Chrome for Android、設定 430
DomClick が Click のように動作しない 502
FAQ 502
FieldInputField.DomClick でダイアログが開かない
502
Google Chrome 489
innerHTML 504
innerText 504
innerText がロケータで使用されない 503
link.select のフォーカスの問題 504
Microsoft Edge 497
Mozilla Firefox 493
textContent 504
新しいページへの移動 504
オブジェクト解決 477
オブジェクト マップ、使用 343
概要 473
カスタム属性、設定 566
クラス リファレンス 508
クロス ブラウザー スクリプト 505
現在のブラウザの種類、表示 503
再生、API とネイティブの比較 479
スクロール 503
設定 574
属性の種類 507, 513
正しくないタイムスタンプ、ログ 504
正しくないロケータの記録 504
テスト オブジェクト 476
フォント タイプの検証 503
ブラウザ構成設定 480
ページ同期 478
マウス移動の記録 504

マウス移動の詳細設定、設定 480, 563
ロケータにないクラスとスタイル 506
ロケータの記録 505
xBrowser クラス リファレンス 2735
xBrowser テスト
Apple Safari、制限事項 487
Microsoft Edge、制限事項 497
現在のブラウザの種類、表示 503
ブラウザの記録オプション、設定 566
XPath
トラブルシューティング 323

あ

アイコン
テスト ステップ 629
表示、メッセージ ボックス 100
アイコンの表示
メッセージ ボックス 100
アクセス
データベース、コピー 601
アクティブ オブジェクト
記録中のハイライト 134
アクティブ プロジェクト
説明 36, 604
アクティブ データ
概要 137
キーワード駆動テスト 137
資産 138
資産の作成 138, 151
利点 138
インポートとエクスポート 309
オプション、詳細設定 582
行数、ビジュアル テスト 150
繰り返しロジック、作成 62
繰り返しロジック、ステップの定義 62
繰り返しロジック、データのマップ 63
繰り返しロジックの作成、ビジュアル テスト 145
検索する 307
更新されたデータの保存、ビジュアル テスト 149
再生、ビジュアル テスト 64
削除 306
シートの選択、VB .NET スクリプト 154
シートの選択、ビジュアル テスト 150
資産の作成、チュートリアル 61
使用、ビジュアル テスト 144
使用するデータの指定 142
スクリプト概要 151
スクリプトへの組み込み 152
チュートリアル 60
データの更新、式の使用 148
データの更新、自動保存の回避 149
データの更新、ステップ テキスト 147
データの更新、ビジュアル テスト 148
データの準備 139
データのマップ、ステップ テキスト 147
データのマップ、ビジュアル テスト 146
データ ファイル、確認 61
データ ファイル、作成 140
データ ファイル、準備 139
名前の変更 305

- 並べ替え 307
- 表示、複数バージョン 308
- 開く 304
- ファイルの編集 142
- フィルタリング 306
- プロパティ 100
- 分析結果、ビジュアル テスト 64
- 読み取りオプションの設定、データ ファイル 141
- 例、詳細 93
- 例、テストシナリオ 93
- 例、ビジュアル テスト 93
- 列数、ビジュアル テスト 150
- アクティブ データ資産
 - 関連付け、ビジュアル テスト 144
 - 作成、アセット ブラウザ 304
- アクティブ データの更新
 - 自動保存の回避 149
- アクティブ データの保存
 - 自動保存の回避 149
- アセット ブラウザ
 - 概要 635
 - 検索する 307
 - 資産の作成 304
 - 資産の並べ替え 307
 - 資産を開く 304
 - ビジュアル テストを開く 91
 - フィルタリング 306
- 値の選択
 - 条件 175
- 新しいブラウザ ウィンドウ
 - ビジュアル テスト 86
- アップロードする
 - キーワード ライブラリ 533
 - ライブラリ 533
- アドオン
 - Google Chrome 491
 - Mozilla Firefox 495
- アプリケーション構成
 - Java Network Launching Protocol (JNLP)、スクリプト 117
 - Java Network Launching Protocol (JNLP)、ビジュアル テスト 85
 - エラー 128
 - 環境変数 129
 - キーワード駆動テスト 521
 - 削除 125
 - 定義 123
 - テクノロジー ドメインの無効化 127
 - トラブルシューティング 128
 - 変更する 125
 - 編集、プロジェクト 124
 - 無応答のアプリケーションのキャンセル 575
- アプリケーション準備完了タイムアウト
 - 設定 575
- アプリケーションの選択
 - ダイアログ ボックス 126
- アプリのアップロード
 - Mac 398
- 安定した識別子
 - について 318
- 安定したロケータ

- 作成 319
- 安定したロケータを作成する
 - 概要 319

い

- 一時停止
 - タイマー 214
 - ビジュアル テスト 213
- 移動
 - 資産、プロジェクト間 305
 - 指定したラベル、ビジュアル テスト 211
 - テスト ステップ 211
 - ビジュアル テストでのラベル 208
 - ラベル テスト ステップの作成、ビジュアル テスト 210
- イメージ クリック
 - イメージ資産変数 360
 - 記録 357
- イメージ解決
 - 概要 357
 - 記録設定の設定 562
 - メソッド 357
 - 有効化 357
- イメージ クリックの記録
 - 概要 357
- イメージ検証
 - イメージ資産変数 362
 - 概要 360
 - 記録中に追加する 362
 - 作成 360
 - 失敗時の相違点の表示 362
 - [相違点] ビュー 363
 - ビジュアル テストへの追加 361
- イメージ資産
 - インポートとエクスポート 309
 - 概要 358
 - 作成、アセット ブラウザ 304
 - 詳細設定の設定 562
 - フィルタリング、アセット ブラウザ 306
 - 複数のイメージ、追加 359
- イメージのクリック
 - 記録の設定 562
- イメージのチェック
 - 概要 360
- 印刷
 - スクリプト 272
 - スクリプトの結果 292
 - ビジュアル テスト 267
 - ビジュアル テストの結果 291
 - ヘルプ トピック 78
- インストール
 - Information Service、Mac 411、486
- インストール済みアプリ
 - Android、テスト 417
 - iOS、テスト 417
- インストールする
 - Information Service、Mac 416、489
 - 必要な権限 584
- インポート
 - 資産 309

インポートの権限
資産 312

う

[ウィンドウ] メニュー 622
埋め込み Chrome
テスト 472
上書きする
オプション、再生 279

え

エイリアス
ロケータの使用 562
エージェント
ポートの構成、Information Service 336
ポート番号 334
オプション 325
概要 325
起動 325
クラス リファレンス 798
エージェント オプション
Open Agent 325
エクスポートする
資産 311
エクスポートの権限
資産 312
エミュレータ
定義、再生 417
テスト 398
エラー
概要 57
確認 60
自動処理の設定、プロパティ ペイン 266
出力ウィンドウに表示 639
処理、再生 265
調べる 57
データベース 603
デバッグ 58
エラー時の移動先プロパティ
プロパティ 105
エラー処理
ロジック 169
エラー処理ロジック
ビジュアルテスト、追加 169
エラーのトレース
スクリプト 290
ビジュアルテスト 290
演算子
式デザイン 180

お

オブジェクト
.NET スクリプトを識別する 119
オブジェクト識別ダイアログ、ビジュアルテスト 88
解決、タイムアウトの設定 575
検索する 313
識別、画面レビュー 88
識別、テスト対象アプリケーション 87

存在確認 317
プロパティ 105
オブジェクト タイプ
ロケータ 313
オブジェクト マップ エントリ
名前の変更 343
オブジェクト解決
改善、UI オートメーション 246
Exists メソッド 317
FindAll メソッド 318
Find メソッドの使用 317
安定したロケータを作成する 318
概要 313
カスタム属性 321
属性の使用 314
複数のオブジェクトの識別 318
ユーザー補助を使用して向上する 245
オブジェクト解決タイムアウト
設定 575
オブジェクト解決の向上
ユーザー補助 245
オブジェクトの解決
xBrowser 477
オブジェクトの識別
概要 313
ダイアログ ボックス 636
[オブジェクトの識別] ダイアログ
ビジュアルテスト 88
オブジェクトの待機
時間の指定 559
ビジュアルテスト 212
オブジェクトの同期化
再生遅延の設定 213
ビジュアルテスト 212
オブジェクトマップ
バージョン、比較 347
比較 347
FAQ 356
Web アプリケーション 343
xBrowser 343
インポートとエクスポート 309
オフに切り替え 341
オンに切り替え 341
概要 340
記録 341, 562
検索する 307
項目のグループ化 355
項目のコピー 349
項目の削除 354
項目の追加 350
項目の名前変更 343
コピー 305
削除 306
削除、スクリプト 356
作成、アセット ブラウザ 304
手動作成 356
詳細設定の設定 562
スクリプトから開く 352
スクリプトの構文 112
スコープ 341, 562
操作の記録中のマージ 342

- 名前の変更 305
- 並べ替え 307
- ビジュアル テストでのロケータからオブジェクト
マップへの移動 84
- 表示、複数バージョン 308
- 開く 304
- フィルタリング 306
- ベストプラクティス 355
- 変更 345
- マージ 356
- 利点 341
- ロック 562
- オブジェクト マップ項目
 - エラーの検出 353
 - グループ化 355
 - コピー 349
 - 削除 354
 - 識別 346, 352
 - 追加 350
 - テスト アプリケーションからの更新 348
 - テスト アプリケーションでの検索 352
 - ハイライト 352
 - ロケータの変更 346
- オブジェクト マップの記録
 - 有効化/無効化 562
- オプション
 - .NET スクリプトの再生結果 579
 - MSUIA、設定 565, 576
 - UI オートメーション、設定 565, 576
 - 詳細設定 581
 - eCATT の統合 557
 - OPT_ENABLE_EMBEDDED_CHROME_SUPPORT
472
 - アクティブ データ、詳細設定 582
 - 開始画面 560
 - 概要 557
 - 記録 561, 568, 571
 - 記録、保存の適用 572
 - 記録の出力、ビジュアル テスト 570
 - グローバル 557
 - 再生 572
 - 再生、値の取得 279
 - 再生、上書き 278, 279
 - 再生、保存の適用 580
 - 再生オプションの上書き 279
 - 再生結果、キーワード駆動テスト 579
 - 再生結果、設定 577
 - 再生ステータス ダイアログ ボックス、設定 297
 - 再生のタイミング、設定 575
 - スクリプト作成 581
 - 全般 558
 - 全般再生、設定 573
 - 統合 557
 - ビジュアル テストの再生 278
 - ビジュアル テストの再生結果 578
 - ブラウザの記録、設定 566
 - 保存 571, 580
- オプション プロファイル
 - 記録、適用 572
 - 再生用の作成 580
 - 適用 580

- オフに切り替え
 - 基本状態 129
- 親スクリプト
 - ドライバ スクリプト 113
- オンに切り替え
 - 基本状態 129

か

- 開始画面
 - オプション 560
 - 概要 41, 613
 - ビジュアル テスト
記録、開始画面 82
 - ビジュアル テスト、記録 82
 - フラグ 560
- 開始行
 - アクティブ データ 142
- 開始タイマー
 - ビジュアル テスト 214
- 概要
 - Oracle データベースの構成 587
 - 自動テスト 32
 - テスト戦略 32
 - 利点 27
- 拡張
 - Google Chrome 491
 - VB .NET スクリプト 217
 - ビジュアル テスト、概要 51
- 拡張機能
 - Mozilla Firefox 495
- カスタマイズ
 - [ツール] メニュー 621
 - ビジュアル ナビゲータのレイアウト 634
- カスタム コントロール
 - Apache Flex の動的呼び出し 370
 - Apache Flex、実装 371
 - Apache Flex、定義 374, 383
 - AUT にコードを追加する 255
 - AUT へのコードの追加、FAQ 257
 - Flex、定義 367
 - 概要 254
 - カスタム クラスの作成 259
 - 管理 258
 - サポート 259
 - 挿入したコードが AUT で使用されない 257
 - ダイアログ ボックス 261
 - テストする (Apache Flex) 257, 365
 - 動的呼び出し、FAQ 255
 - 呼び出しで予期しない文字列が返される 255
- カスタム コントロールのテスト
 - AUT にコードを追加する 255
- カスタム属性
 - UI オートメーション、設定 565, 576
 - Apache Flex アプリケーション 321, 383
 - Web アプリケーション 322, 507
 - Windows Forms アプリケーション 323, 436
 - WPF アプリケーション 323, 443
 - 記録オプション、設定 566
 - コントロール 321
 - 設定 567

- カスタム プロパティ
 - コントロール 321
- 画面キャプチャ
 - .NET スクリプトの再生オプション 579
 - 画面プレビュー 189
 - 記録オプション 568
 - 記録オプション、設定 570
 - 再生オプション、キーワード駆動テスト 579
 - ビジュアル テストの再生オプション 578
- 画面のキャプチャ
 - 手動 89
- 画面の最小化
 - 検索中 559
 - 再生中 559
- 画面のフラッシュ
 - 記録中 559
- 画面プレビュー
 - キャプチャされたイメージ、表示 631
 - コントロール、識別 88
 - コントロールの識別、コントロールの自動ステップ 187
 - ステップの削除 215
 - 設定、構成 189
 - ビジュアル テストの更新 51
 - 表示 632
- 環境変数
 - アプリケーション構成 129
- 監視
 - 再生中の変数 193
- 関数
 - 式デザイナー 182
- カンマ区切りファイル
 - アクティブ データ、準備 139
- 管理
 - 概要 584
 - キーワード 524
 - グループ 611
 - グローバル変数 197
 - 結果 288
 - 資産 299
 - スクリプト 235
 - テスト フロー 208
 - テスト フロー、移動の作成 211
 - テスト フロー、ラベル テスト ステップの作成 210
 - プロジェクト 36, 604
 - プロジェクト、削除 608
 - プロジェクト、追加 604
 - プロジェクト、名前の変更 607, 608
 - ユーザー 609
 - ユーザー、削除 610
 - ユーザー、追加 609
 - ユーザー、編集 610
 - ユーザー アクセス 609
- 管理者権限
 - インストールする 584
 - 実行 584
- 関連付け
 - アクティブ データ、ビジュアル テスト 146
 - アクティブ データ資産、ビジュアル テスト 144

き

- キー モニック
 - Rumba 1808
- キーボード
 - ショートカット 37
- キーボード イベント遅延
 - 設定 575
- キーワード
 - プロジェクトでの検索 535
 - Silk Central にアップロードする 533
 - 管理 524
 - 記録 522
 - グループ化 535
 - 結合 524, 530
 - 削除 524
 - シーケンス 524
 - 実装 521, 524
 - 置換 524
 - 追加 524
 - データの受け渡し 523
 - テスト メソッドの指定 523
 - について 518
 - ネスト 524
 - パラメータ 524, 527
 - パラメータ、例 528
 - ビジュアル テスト、指定 522
 - 開く 524
 - 開く、Silk Central 531
 - フィルタリング 535
- キーワード シーケンス
 - インポートとエクスポート 309
 - 作成 530
 - 作成、アセット ブラウザ 304
 - 名前の変更 305
 - 並べ替え、アセット ブラウザ 307
 - パラメータ 527
 - 開く 304
 - フィルタリング、アセット ブラウザ 306
- キーワード ライブラリ
 - アップロードする 533
- キーワード駆動
 - テスト 517
- キーワード駆動テスト エディター
 - 推薦するキーワード 527
- キーワード駆動テストの実行
 - 変数 530
- キーワードの削除
 - キーワード駆動テスト 524
- キーワードの追加
 - キーワード駆動テスト 524
- 既知の問題
 - Google Chrome 642
 - Internet Explorer 643
 - Microsoft Edge 644
 - Mozilla Firefox 645
 - Oracle Forms 647
 - SAP 645
 - Silk Test Workbench 647
 - Web アプリケーション 642
 - 一般的な問題 640

- について 640
- モバイル Web アプリケーション 642
- 基本機能
 - Silk Test Workbench 30
 - 自動テスト 32
 - ソフトウェア コンポーネント 32
 - テスト戦略 32
 - 利点 27
- 基本状態
 - 実行 130
 - オフに切り替え 129
 - オンに切り替え 129
 - キーワード駆動テスト 521
 - 詳細 129
 - 変更、スクリプト 132
 - 変更、ビジュアル テスト 132
 - 変更、ユーザー インターフェイス 130
- キャプチャ
 - Web ページ、全画面 294
 - アプリケーション画面、.NET スクリプト 579
 - アプリケーション画面、キーワード駆動テスト 579
 - アプリケーション画面、ビジュアル テスト 578
 - アプリケーションの画面、設定 570
- キャプチャ画面
 - 更新、再生 191
 - 更新、テスト アプリケーション 191
- キャプチャ画面の更新
 - 再生、ビジュアル テスト 191
 - すべて 191
 - ビジュアル テスト、テスト アプリケーション 191
- キャプチャされたイメージの表示
 - 画面プレビュー 631
- 行数
 - アクティブ データ、ビジュアル テスト 150
- 行の実行の制御
 - スクリプトのデバッグ 270
- 記録
 - 画像が表示されない 424
 - Apache Flex アプリケーション 388
 - イメージ検証を追加する 362
 - オブジェクト マップ 341
 - オブジェクトのハイライト 134
 - オプション 561, 571
 - オプション プロファイル 571
 - オプション プロファイル、適用 572
 - 概要 134
 - キーワード 522
 - キーワード駆動テスト 519
 - 既存のテストへの操作 122
 - 事前読み込みの設定 449
 - 出力オプション 568
 - 出力オプション、ビジュアル テスト 570
 - スクリプト、概要 69
 - データ入力 220-223
 - データベース 598
 - デバイスの開放 423
 - ビジュアル テスト 80, 90
 - ビジュアル テスト、開始画面 82
 - ビジュアル テスト、複数のアプリケーション 83
 - ビジュアル テストを使用する利点 92
 - モバイル アプリケーション 417

利用可能なアクション 135



- 組み込んだスクリプト
 - パラメータの使用、ビジュアル テスト 226
 - 変更する 209
- 組み込んだビジュアル テスト
 - 実行 274
 - パラメータ、値の割り当て 204
 - パラメータ、再利用 208
- クラス
 - 無視 564
- クラス リファレンス
 - エージェント 798
 - キーワード駆動テスト 1784
 - テスト オブジェクト 798
- 繰り返しロジック
 - アクティブ データ、ステップの定義 62
 - アクティブ データ、リテラル データへのマップ 63
 - アクティブ データ テスト 145
 - コンボ ボックスに基づく 162
 - 作成、アクティブ データ 62
 - ビジュアル テスト、追加 162
 - リストに基づく 162
- 繰り返しロジックの定義
 - コンボ ボックスに基づく 162
 - リストに基づく 162
- クリック
 - モバイル Web 434
- グループ
 - 管理 611
 - 権限の編集 612
 - 削除 612
 - 追加 611
- グループ化
 - オブジェクト マップ項目 355
 - キーワード 535
- クローズ オプション
 - 設定 577
- グローバル プロジェクト
 - 定義 605
- グローバル参照
 - 定義 605
- グローバル変数
 - 管理 197
 - 作成、ビジュアル テスト 197
 - 取得、ビジュアル テスト 198
 - ビジュアル テスト 196
- クロス ブラウザ テスト
 - Apple Safari、制限事項 487
 - Microsoft Edge、制限事項 497
 - Apple Safari 485
 - FAQ 502
 - Google Chrome 489
 - Microsoft Edge 497
 - Mozilla Firefox 493
 - オブジェクト解決 477
 - オブジェクト マップ、使用 343
 - 概要 473
 - 現在のブラウザの種類、表示 503

- スクロール 503
- 接続文字列 482
- 正しくないタイムスタンプ、ログ 504
- テストオブジェクト 476
- ブラウザの記録オプション、設定 566
- マウス移動の詳細設定、設定 480, 563
- リモートロケーション、追加 126
- ロケータの記録 505

け

継続的インテグレーション

- キーワードライブラリのアップロード 533

結果

- アセットブラウザから開く 290
- インポートとエクスポート 309
- エラーの確認 60, 76
- 概要 48
- カスタマイズ 289
- カスタマイズする 559
- 管理 288
- [結果] ウィンドウ 283, 634
- 結果の記録のオン/オフの切り替え 289
- 検索する 307
- コピー 305
- コメント 291
- 最大数、設定 307
- 削除 306
- 作成 288
- 作成、アセットブラウザ 304
- 実行ステータス 286
- 自動的に開く 289
- 手動で開く 289
- スクリプトでの検証 218
- スクリプトの印刷 292
- ステップのステータス 286
- タブ、ウィンドウ 284
- ツールバー 283
- データベース整合性チェック 603
- 名前の変更 305
- 並べ替え 307
- について 282
- ビジュアルテストへの切り替え 290
- ビジュアルテストの印刷 291
- ビジュアルナビゲータ 287
- 表示、複数バージョン 308
- 開く 304
- ファイルの保存 293
- フィルタ 294
- フィルタ、作成 294
- フィルタ、適用 296
- フィルタの削除 297
- フィルタリング 306
- プロパティ 109
- 分析 273, 282
- [要約] タブ 284

[結果] ウィンドウ

- 概要 283, 634
- 画面プレビュー 50
- タブ 49
- ツールバー 49

- ビジュアルナビゲータ 287
- [プロパティ] ペイン 49

結果の作成

- スクリプト 288
- ビジュアルテスト 288

結果の分析

- スクリプト 282
- ビジュアルテスト 282

結果フィルタ

- 適用 296
- 複製 296
- 編集 295

結合

- キーワード 524

言語リファレンス

- 概要 650
- キーワード駆動テストクラス 1784
- コアクラス 798

現在の時刻の取得

- FormatTime 関数 186

現在の日付を取得

- FormatDate 関数 184

現在のブラウザの種類

- 表示 503

検索する

- キーワード、キーワード駆動テスト 535
- 資産 307

検索範囲

- ロケータ 313

検証

- インポートとエクスポート 309
- 作成、アセットブラウザ 304
- スクリプトへの追加 72, 218
- 追加 52
- データベース 603
- フィルタリング、アセットブラウザ 306

検証結果の説明

- プロパティ 110

検証ロジック

- 記録 80
- 記録中のスクリプトの追加 218
- コーディングによるスクリプトへの追加 218
- 追加する 172
- ビジュアルテスト、追加 166

こ

コアクラス

- ConsoleWindow クラス 819

更新

- アクティブデータファイル、式の使用 148
- アプリケーションの画面 191
- データ、アクティブデータファイル 148
- データの保存、アクティブデータファイル 149
- ビジュアルテスト 89

構成する

- SQL Server 584
- 再生、ビジュアルテスト 278
- ビジュアルテストの再生、挿入されたテスト 280

項目識別子

- ビジュアルテスト 192

- 項目の指定
 - ビジュアル テスト 192
- コード
 - 検索、VB .NET スクリプト 235
 - [コード] ウィンドウ 638
 - 置換、VB .NET スクリプト 235
 - ブックマークを使用したスクリプトでの移動 235
- [コード] ウィンドウ
 - について 638
- 子スクリプト
 - ドライバ スクリプト 113
- コピー
 - 結果フィルタ 296
 - スクリプト 305
 - データベース 601
 - テスト ステップ 215
- コマンド
 - コマンド プロパティ 102
 - ショートカットキー 37
 - [ツール] メニューに追加 621
- コマンド ライン
 - silktest.exe コマンド ライン 537
 - STW.EXE パラメータ 537
 - STW.EXE の例 547
 - STW エラー 543
 - STW コマンド ラインの概要 537
 - STW 出力ファイル 543
 - 資産インポートの構成ファイル 552
 - 資産エクスポートの構成ファイル 554
 - 資産のインポートとエクスポート 549
 - 資産のインポート パラメータ 549
 - 資産のエクスポート パラメータ 552
 - 入力ファイル 542
 - ログ 555
- コメント
 - 結果 291
 - ビジュアル テストへの追加 154
 - メッセージ ボックスでの表示 210
- コントロール
 - .NET スクリプトを識別する 119
 - オブジェクト識別ダイアログ、ビジュアル テスト 88
 - 識別、画面プレビュー 88
 - 識別、テスト対象アプリケーション 87
 - ビジュアル テストでのプロパティ値 189
- コントロールの検索
 - GWT の例 320
 - 同列要素の例 319
- コントロールの検証
 - 判断ロジックの利用、例 97
- コントロールの識別
 - コントロールの自動化、画面プレビュー 187
 - コントロールの自動化、テスト アプリケーション 187
 - コントロールの自動化、非表示コントロール 188
 - 動的ロケータ属性 515
- コントロールの自動ステップ
 - 作成、画面プレビュー 187
 - 作成、テスト アプリケーションのコントロール 187
 - 作成、非表示コントロール 188
- コンパクト化
 - データベース、Access 602

さ

- 再開タイマー
 - ビジュアル テスト 214
- サイズ変更
 - 画面イメージ、画面プレビュー 632
- 再生
 - .NET スクリプトの結果オプション 579
 - エラー処理、ビジュアル テスト 265
 - オブジェクトの待機 212
 - オプション 572
 - オプション、値の取得 279
 - オプション プロファイル 580
 - 画面の最小化 559
 - キャプチャ画面、更新 191
 - 記録されたスクリプト 71
 - クローズ オプション 577
 - 結果オプション、キーワード駆動テスト 579
 - 結果オプション、設定 577
 - 再生ステータス ダイアログ ボックス、オプションの設定 297
 - 再生ステップのプロパティ 108
 - 事前読み込みの設定 449
 - 情報の表示 210
 - スクリプト 276
 - ステップ実行の一時停止 213
 - すべてのファイルの保存 559
 - 設定 106
 - 設定、挿入されたスクリプト 280
 - 設定、挿入されたビジュアル テスト 280
 - 設定、ビジュアル テスト 278
 - 全般オプション、設定 573
 - タイミング 214
 - タイミング、オプションの設定 575
 - テスト 273
 - テスト ロジック デザインを使用して再生エラーに自動的に対処するようにビジュアル テストを設定する 266
 - デバイスの開放 423
 - デバイスの選択 417
 - ビジュアル テスト 48
 - ビジュアル テスト、組み込み 274
 - ビジュアル テストでのスクリプトの実行 209, 226
 - ビジュアル テストの結果オプション 578
 - ブラウザーの選択 474
 - 保存したオプション、適用 580
- 再生
 - 認識されないダイアログ 502
- 再生エラー
 - 概要 57
 - 確認 60
 - 調べる 57
 - ダイアログ ボックス 298
 - デバッグ 58
 - 変数の追跡 59
- 再生オプション
 - 上書きする 278
- 再生結果
 - について 282
- 再生結果オプション
 - .NET スクリプト 579

- キーワード駆動テスト 579
- ビジュアルテスト 578
- 再生設定の取得
 - プロパティ 104
- 再生設定の設定
 - プロパティ 109
- 再生の完了
 - ダイアログ ボックス 298
- 再生の停止
 - スクリプト、選択したポイント 271
 - ビジュアルテスト、選択したポイント 263
- 再生ホットキー
 - 変更 580
- 再生モード
 - 設定 573
- 最大化
 - ブラウザー 502
- 削除
 - アプリケーション構成 125
 - キーワード 524
 - グループ 612
 - 結果フィルタ 297
 - 資産 306
 - 資産、古いバージョン 300
 - データベース レコード、Access 602
 - テスト ステップ、画面プレビュー 215
 - テスト ステップ、ストーリーボード 216
 - テスト ステップ、テスト ステップ ウィンドウ 216
 - プロジェクト 608
 - ユーザー 610
- 削除する
 - データベース レコード、Access 602
 - プロジェクト 608
 - ユーザー 610
- 作成
 - アクティブ データ資産 138, 151
 - DSN 594
 - アクティブ データ テスト 142
 - 移動、ビジュアル テスト 211
 - キーワード駆動テスト 519
 - グローバル変数、ビジュアル テスト 197
 - 結果フィルタ 294
 - パラメータ 203
 - ビジュアル テスト 80
 - ビジュアル テスト、複数のアプリケーション 83
 - プロジェクト 604
 - ラベル テスト ステップ 210
 - ラベル ステップ 208
- サポートされる Java バージョン
 - Oracle Forms 647
- 参照
 - スクリプトへの追加 238
 - プロジェクト 605
 - プロジェクト、グローバル 605
 - ライブラリからの削除 239
- サンプル アプリケーション
 - 起動 44, 69
 - 記録、ビジュアル テスト 45
 - スクリプトの記録 70

し

- シートの選択
 - アクティブ データ、VB .NET スクリプト 154
 - アクティブ データ、ビジュアル テスト 150
- 式
 - アクティブ データの更新 148
 - 変更 179
- 式デザイナー
 - Format Date 関数 184
 - FormatTime 関数 186
 - 演算子 180
 - 概要 177
 - 式の作成 177
 - 式の編集 179
- 式デザイナー
 - 関数 182
- 識別
 - オブジェクト、[オブジェクトの識別] ダイアログ ボックス 636
 - コントロール 187
 - コントロール、画面プレビュー 88
 - コントロール、テスト対象アプリケーション 87
- 識別子
 - 安定 318
- 識別する
 - コントロール、.NET スクリプト 119
- 時刻
 - FormatTime 関数 186
- 資産
 - インクルード元 308
 - 作成、アクティブ データ 138, 151
 - バージョン、比較 304
 - 比較 304
 - アクティブ データ 138
 - 移動 305
 - インクルード 308
 - インポート 309
 - インポートとエクスポート 309
 - インポートの権限 312
 - エクスポートする 311
 - エクスポートの権限 312
 - 概要 299
 - 管理 299
 - 結果、作成 288
 - 削除 306
 - 作成、アセット ブラウザ 304
 - 作成、命名規則 309
 - サポートされる種類 299
 - 資産のバージョンをページ 300
 - スクリプトから開く 304, 360
 - 接頭部の追加 559
 - データ ファイル、作成 140
 - デフォルトの保存動作、設定 307
 - 名前の変更 305
 - 並べ替え 307
 - ページするバージョンについて 300
 - ビジュアル テスト、作成 80
 - 表示、複数バージョン 308
 - フィルタリング 306
 - 複製 305

- 命名規則 309
- 資産エクスポート ウィザード
 - 使用法 311
- 資産管理
 - 資産のバージョンの最大数 557
 - デフォルトの保存動作 557
- 資産の格納
 - 最大バージョン数、設定 307
 - バージョンの最大数 557
- 資産の作成
 - アセット ブラウザ 304
- 資産の種類
 - 利用可能 299
- 資産のバージョン
 - 最大数 557
 - 最大数、設定 307
- 資産のバージョンをパーズ
 - について 300
 - パラメータ、コマンド ライン 301
 - 方法 300
- 資産の表示
 - 複数バージョン 308
- 資産の保存
 - デフォルトの保存動作、設定 307
- 資産のメンテナンス
 - ユーザー、ログアウト 610
- 事前読み込み
 - 記録/再生時の設定 449
- 実行
 - スクリプト 276
 - スクリプト、概要 276
 - テスト 273
 - ビジュアル テスト 273
 - ビジュアル テスト、組み込み 274
 - ビジュアル テストでのターゲット アプリケーション 211
- 実行ステータス
 - 結果 286
- 実行の遅延
 - テスト 262
- 実装
 - キーワード 524
- 自動
 - コントロール プロパティの使用 189
 - 操作の表示、再生 277
 - 操作の表示、設定 573
 - テスト ステップ 214
 - テスト ステップ、記録せずに作成 187
- 自動ステップ
 - 画面プレビュー、ビジュアル テスト 189
 - ステップの同期、ビジュアル テスト 189
- 自動保存の回避
 - アクティブ データ、ビジュアル テスト 149
- シミュレータ
 - 定義、再生 417
 - テスト 406
 - ネイティブ アプリ、テスト 407
 - モバイル Web アプリケーション、テスト 408
- 周辺機器が無い
 - テスト マシン 28
- 終了行

- アクティブ データ 142
- 出カウィンドウ 639
- 出カパラメータ
 - スクリプト 237
 - スクリプトからの削除 238
 - 定義 67
 - 編集 237
- 手動作成
 - オブジェクト マップ 356
- 循環参照
 - 回避、プロジェクト 606
- 順序
 - アセット ブラウザの列 307
- 準備
 - Silk Test Workbench 26
- 条件
 - 値の選択 175
 - ダイアログ ボックス 297
- 条件デザイン
 - 値の選択 175
 - 条件の編集 175
- 条件デザイナー
 - 概要 172
 - 条件の定義 174
- 条件の選択
 - ダイアログ ボックス 175
- 条件の定義
 - プロパティに基づく 158, 162
- 詳細設定プロパティ
 - 概要 100
- 使用状況データの収集
 - 無効化 78
 - 有効化 78
- ショートカット
 - [デバッグ] メニュー 619
- ショートカット キー
 - リスト 37
- 除外される文字
 - 記録 135
 - 再生 135
- 初期化する
 - Apache Flex アプリケーション 388
- シリアル番号 78, 79

す

- 推薦するキーワード
 - キーワード駆動テスト エディター 527
- 数値パラメータ
 - 例 233
- スクリーンキャスト
 - 機能しない 424
- スクリプト
 - .NET スクリプトでの参照 239
 - 再生 71, 73, 276
 - 実行中の停止 276
 - バージョン、比較 240
 - 比較 240
 - ロケーターの使用 316
 - DLL ファイルの参照 238

- Java Network Launching Protocol (JNLP) の構成 117
- アクティブ データの使用 151, 152
- 新しいブラウザ ウィンドウ 118
- アプリケーションの起動 120
- 印刷 272
- インポート 309
- インポートとエクスポート 309
- エラー処理の例 270
- エラーのトレース 290
- オートメーション呼び出しの表示 573
- オートメーション呼び出しの表示、再生 277
- オブジェクト マッピング 340
- 拡張 217
- 確認 71
- 管理 235
- 記録 115
- 記録、概要 69
- 記録中の検証の追加 218
- 組み込んだ、変更 209
- 結果 282
- 結果、作成 288
- 結果の印刷 292
- 検索する 307
- 構文 112
- コーディングによる検証の追加 218
- コードの検索 235
- コードの置換 235
- コメントの挿入 271
- 再生、概要 276
- 再生エラー 76
- 再生の停止、選択したポイント 271
- 再利用 217
- 削除 306
- 作成 110
- 作成、アセット ブラウザ 304
- 作成、ベスト プラクティス 111
- サンプル アプリケーション 44, 69
- サンプル アプリケーションに対する記録 70
- 出力パラメータ 67
- 出力パラメーターの削除 238
- 出力パラメーターの追加 237
- 出力パラメーターの編集 237
- 手動作成 114
- 信頼性 269
- スクリプトでの参照 239
- スクリプトでのパラメーターの作成と受け渡し 227
- スクリプト内からのスクリプトの参照 217
- 挿入 75
- 段階的実行、選択したポイントから 270
- 段階的実行、デバッグ モード 269
- チュートリアル、2 つめのスクリプトの記録 75
- テストをキーワードとして指定 523
- デバッグ 268
- デバッグ、行の実行の制御 270
- ドライバ 113
- ドライバーの作成 114
- 名前の変更 305
- 並べ替え 307
- 入力パラメータ 67, 235
- 入力パラメーターの削除 236
- 入力パラメーターの編集 236
- パラメータ、受け渡し 227
- パラメータ、作成 227
- パラメータ、定義 224
- パラメータの受け渡し、スクリプト内 229
- ビジュアル テストでの実行 209, 226
- ビジュアル テストでの使用 67, 68
- ビジュアル テストにデータを渡す 224
- 表示、複数バージョン 308
- 開く 122, 304
- ファイルの削除 238
- ファイルの追加 238
- ファイル名の変更 238
- フィルタリング 306
- 複数のアプリケーションの記録 117
- 複製 305
- [プロパティ] ペインの表示 638
- 編集 122
- 変数 220-223
- 変数の宣言 221
- 保存 123
- モジュール式 74
- 乱数の生成 65
- 利点 111
- ロケーターの記録 121
- スクリプト パラメータ
 - ビジュアル テスト、使用 226
- スクリプト作成
 - オプション 581
- スクリプト参照ライブラリ
 - 参照の削除 239
 - スクリプトへの追加 239
- スクリプト内からのスクリプトの参照 217
- スクリプトの拡張
 - 概要 72
 - 検証の追加 72
 - 再生 73
 - 変数の追加 73
- スクリプトの記録
 - サンプル アプリケーション 70
 - チュートリアル、2 つめの記録 75
- スクリプトの再生
 - 拡張したスクリプト 73
 - モジュール式 76
- スクリプトの削除
 - オブジェクト マップ エントリ 356
- スクロール
 - クロス ブラウザ テスト 503
- スタイル
 - Flex アプリケーション 389
- ステップ
 - アクティブ データの更新 147
 - 実行の制御、再生 264
 - ビジュアル テスト 214
- ステップ実行の制御
 - デバッグ 264
- ステップの使用
 - 挿入 89
- ステップの選択
 - ダイアログ ボックス 171
- ステップの同期

- ロジック ツールボックス 171
- ストーリーボード
 - 概要 633
 - サムネイルの移動 634
 - テストステップ、削除 216
- スプレッドシート
 - アクティブ データ、準備 139
- スリープ
 - テストへの追加 262

せ

- 製品サポート 78, 79
- 製品スイート
 - コンポーネント 29
- セキュリティ設定
 - SAP 459
- 接続
 - データベース 596
- 接続の設定
 - データベース 597
- 接続文字列
 - デスクトップ ブラウザー、リモート 482
 - デスクトップ ブラウザー、ローカル 483
 - モバイル デバイス 419
- 接続ユーザー
 - データベース、表示 609
 - ログアウト 610
- 設定
 - アクティブ データ、詳細設定 582
 - 記録 571
 - 記録、保存の適用 572
 - グループ 611
 - 再生、全般 573
 - 再生、保存の適用 580
 - 再生結果、.NET スクリプト 579
 - 再生結果、オプション 577
 - 再生結果、キーワード駆動テスト 579
 - 再生結果、ビジュアル テスト 578
 - 再生ステータス ダイアログ ボックス 297
 - 再生のクローズ オプションの設定 577
 - 再生のタイミング 575
 - 全般、変更 559
 - ビジュアル テスト、新しいブラウザ ウィンドウ 86
 - 変数のコントロール プロパティ値 189
 - 保存 571, 580
 - マウス移動の詳細設定、クロス ブラウザ テスト 480, 563
 - ユーザー 609
 - ユーザー、追加 609
- 設定を構成する
 - 画面プレビュー 189
- 全画面表示
 - ブラウザ 502
- 宣言
 - スクリプトにおける変数 221
- 選択
 - テスト ステップ、複数 214
- 全般
 - オプション 558
- 全般オプション

- 変更する 559
- 全般プロパティ
 - 概要 103

そ

- 相違点の表示
 - イメージ検証 362
- [相違点] ビュー
 - イメージ検証 363
- 操作
 - 更新、テスト ステップ 191
 - テスト ステップ、設定 629
- 操作の記録
 - オブジェクト マップ エントリのマージ 342
- 操作の選択
 - ビジュアル テスト 154
- 操作の変更
 - ビジュアル テスト 154
- [操作] メニュー
 - コマンド 616
- 操作を記録する
 - 既存のテスト 122
 - ビジュアル テスト 89
- 挿入
 - 結果コメント 291
 - コメント 154
 - メニュー 618
- 挿入されたスクリプト
 - 再生オプションの上書き 280
- 挿入されたビジュアル テスト
 - 再生オプションの上書き 280
- 属性の種類
 - Apache Flex 390, 509
 - Java AWT 392, 509
 - Java Swing 392, 509
 - Java SWT 395, 509
 - Oracle Forms 394
 - SAP 457, 510
 - Silverlight 450, 510
 - Web アプリケーション 507, 513
 - Windows 441, 513
 - Windows Forms 436, 513
 - xBrowser 507, 513
 - 概要 508
- その他
 - プロパティ 104
- ソフトウェア コンポーネント
 - 概要 32

た

- ターゲット アプリケーション
 - ビジュアル テストでの起動 211
- ダイアログ
 - 認識しない 502
- ダイアログ ボックス
 - 再生エラー 298
 - 再生の完了 298
 - 条件 297
 - 条件の選択 175

- ステップの選択 171
- テストスクリプトの参照 239
- ビジュアルテスト変数 275
- 待機
 - プロパティ 110, 213
- タイマー
 - プロパティ 109
- タイミング
 - アプリケーションとの同期 212, 213
 - ステップ 214
- タイムスタンプ
 - 正しくない、クロス ブラウザ テスト 504
- ダウンロード 78, 79
- 正しくないタイムスタンプ
 - ログ、クロス ブラウザ テスト 504
- 段階的実行
 - スクリプト、選択したポイント 270
 - スクリプト、デバッグ モード 269
 - ビジュアルテスト、デバッグ モード 265
 - ビジュアルテスト、選択したポイント 263

ち

- 遅延
 - キーボード イベント 575
 - ステップの実行 213
 - マウス イベント 575
- 遅延プロパティ
 - 概要 103
- 置換
 - キーワード 524
- チュートリアル
 - アクティブ データ 60
 - 再生 68
 - 乱数の生成 65

つ

- 追加
 - キーワード 524
 - グループ 611
 - 資産、命名規則 309
 - [ツール] メニューの項目 621
 - テスト ステップ 208
 - プロジェクト 604
 - ユーザー 609
 - ローカル変数 194
- ツールバー
 - 操作 622
- [ツール] メニュー
 - コマンド 620

て

- 定義
 - 開く 123
- 停止タイマー
 - ビジュアル テスト 214
- データ
 - アクティブ データ 142
 - 外部、追加 140

- 共有、ビジュアル テスト 203
- スクリプトでの再利用 220
- ビジュアル テスト、再利用 192
- データ型
 - ClickType 836
 - Color 836
 - ItemIdentifier 837
 - ItemPath 838
 - ModifierKeys 838
 - MouseButton 839
 - Point 839
 - Range 839
 - Rectangle 840
 - TextPosition 840
 - TextRange 841
 - TreeContent 841
 - TreeNode 842
 - について 836
 - 変数、ビジュアル テスト 192
- データ ソース名 (DSN)
 - 作成 594
- データ ソース名 (DSN) の作成
 - Access 594
- データ ソース名 (DSN) の作成
 - Oracle 596
 - SQL Server 595
- データの共有
 - ビジュアル テスト 203
- データの再利用
 - グローバル変数 196
 - グローバル変数、作成 197
 - グローバル変数、取得 198
 - スクリプト パラメータ、ビジュアル テストでの使用 226
 - パラメータ、定義 224
 - パラメータ、ビジュアル テスト 208
 - ビジュアル テスト 192
 - ビジュアル テスト、変数 192
 - 予約変数 198
 - 予約変数、戻り値型 199
 - 予約変数、リスト 198
 - ローカル変数 194
 - ローカル変数、編集 195
- データの種類
 - 例 222
- データの準備
 - アクティブ データ ファイル 139
- データベース
 - Access を開く 600
 - Oracle を開く 599
 - Oracle、作成 589
 - SQL を開く 599
 - 構成 584
 - コピー 601
 - コンパクト化、Access 602
 - 整合性チェック 603
 - 接続 596
 - 接続、設定 597
 - 接続ユーザー、表示 609
 - バージョン、更新 602
 - バージョンの更新 600

- 開く 599
- 別の場所 33
- 膨張の抑制 603
- メンテナンス 598
- メンテナンス、接続ユーザー 603
- メンテナンス設定、構成 602
- レコードのロック解除 601
- データベース メンテナンス
 - メッセージ、受信 603
- データベース接続
 - 構成 597
 - 設定 597
- データベース接続の構成 597
- データベースのメンテナンス
 - 設定、構成 602
- データベースのレコードのロック解除 601
- データベースレコード
 - 削除、Access 602
- テキスト
 - 表示 210
- テキストの表示
 - メッセージ ボックス 103
- テキスト ファイル
 - アクティブ データ、準備 139
- 適用
 - 記録オプション 572
 - 再生オプション 580
- テクノロジドメイン
 - 無効化 127
- デスクトップ警告
 - 表示 559
- テスト
 - 拡張 137
 - 再生 273
 - 作成 80
 - 実行の遅延 262
 - 操作を記録する 122
 - モジュール式 55, 74
 - 利点 27, 32
- テストステップ ウィンドウ
 - ステップの削除 216
- テストマシン
 - 周辺機器が無い 28
- テストメソッド
 - キーワードとして指定 523
- テストロジック
 - 概要 157
 - 繰り返しロジック 161
 - 繰り返しロジック テスト ステップの編集 165
 - 検証ロジック 165
 - 検証ロジック テスト ステップの編集 169
 - ステップのコピーおよび貼り付け 215
 - テストロジック デザイナ 170
 - テストロジック デザイナ ウィザード 172
 - 判断ロジック 157
 - 判断ロジック テスト ステップの編集 161
 - ロジック ツールボックス 171
- テストロジック デザイナ
 - [エラー処理の構築] ページ 169
 - 概要 170
 - 繰り返し、コンボ ボックスに基づく 162
 - 繰り返し、リストに基づく 162
 - [繰り返しの構築] ページ 162
 - [検証資産の構築] ページ 166
 - [使用するアクティブ データ資産の定義] ページ 162
 - プロパティに基づく条件、定義 158, 162
 - [ようこそ] ページ 170
 - ようこそページ、表示 559
 - 要約ページ、エラー処理ロジック 169
 - 要約ページ、繰り返しロジック 162
 - 要約ページ、検証ロジック 166
 - 要約ページ、判断ロジック 158
 - 要約ページ、表示 559
 - ロジック ステップ ペイン、表示 559
 - [ロジックの種類を選択 (エラー処理ロジック)] ページ 169
 - [ロジックの種類を選択 (繰り返しロジック)] ページ 162
 - [ロジックの種類を選択 (検証ロジック)] ページ 166
 - [ロジックの種類を選択 (判断ロジック)] ページ 158
- テスト ロジック デザイナ ウィザード
 - [繰り返し (While) の構築] ページ 162
 - [繰り返し回数の定義] ページ 162
 - [検証の構築] ページ 166
 - 挿入 172
 - 挿入する 172
 - [タイミングに基づく条件の定義] ページ 166
 - [判断の構築] ページ 158
- テスト アプリケーション
 - オブジェクト識別ダイアログ、ビジュアル テスト 88
 - 画面イメージ、表示 632
 - キャプチャ画面、更新 191
 - コントロールの識別 87, 187
 - コントロールの識別、コントロールの自動ステップ 187
 - コントロールを識別する、.NET スクリプト 119
 - 非表示コントロールの識別、コントロールの自動ステップ 188
- テスト オブジェクト
 - クラス リファレンス 798
- テスト結果
 - 概要 48
 - 画面プレビュー 50
 - タブ 49
 - ツールバー 49
 - [プロパティ] ペイン 49
- テスト自動化
 - 障壁 28
 - 同期 280
 - ビジュアル テスト 91
- テスト スクリプト
 - 新しいブラウザ ウィンドウ 118
 - アプリケーションの起動 120
 - 記録 115
 - 結果 282
 - 結果、作成 288
 - 構文 112
 - 再生、概要 276
 - 作成 110
 - 作成、ベスト プラクティス 111
 - 手動作成 114
 - ドライバ 113

- ドライバーの作成 114
- 開く 122
- 複数のアプリケーションの記録 117
- 編集 122
- 保存 123
- 利点 111
- ロケータの記録 121
- テスト スクリプトの参照
 - ダイアログ ボックス 239
- テスト ステップ
 - アイコン 629
 - アクティブ データの更新 147
 - アプリケーションの実行 211
 - 移動 208
 - 移動、指定したラベル 211
 - 画面の更新 191
 - 記録 89
 - 組み込んだスクリプト、変更 209
 - グローバル変数 196
 - グローバル変数、管理 197
 - グローバル変数、作成 197
 - グローバル変数、取得 198
 - コピー 215
 - 削除、画面プレビュー 215
 - 削除、ストーリーボード 216
 - 削除、テスト ステップ ウィンドウ 216
 - 実行の制御、再生 264
 - 自動、手動で作成 187
 - 種類、ビジュアル テスト 629
 - パラメータ、削除 206
 - パラメータ、作成 203
 - 貼り付け 215
 - 判断ロジック、例 97
 - 判断ロジック、例のシナリオ 97
 - 判断ロジック、例の詳細 97
 - ビジュアル テスト 214
 - ビジュアル ナビゲータ 627
 - 表示 87
 - 複数、選択 214
 - プロパティ、ビジュアル テスト 632
 - プロパティ、表示 632
 - 変数のコントロール プロパティ 値 189
 - メッセージ ボックス 210
 - 予約変数 198
 - 予約変数、戻り値型 199
 - 予約変数、リスト 198
 - ラベル 208
 - ラベル、作成 210
 - ローカル変数 194
 - ローカル変数、追加 194
 - ローカル変数、編集 195
 - ローカル変数の削除 196
- テスト ステップの作成
 - ラベル 210
- テスト ステップの種類
 - ビジュアル テスト 629
- テストの再生
 - リモート マシン 339
- テスト フロー
 - 管理 208
- テストを実行する

- eCATT 470
- デバイスの開放
 - 記録 423
 - 再生 423
 - モバイル テスト 423
- デバッグ
 - エラー処理、ビジュアル テスト 265
 - 行の実行の制御、スクリプト 270
 - スクリプト 268
 - スクリプト、選択したポイントからの段階的実行 270
 - スクリプト、選択したポイントでの再生の停止 271
 - ステップ実行の制御、再生 264
 - 段階的実行、スクリプト 269
 - 段階的実行、ビジュアル テスト 265
 - テスト 263
 - ビジュアル テスト 263
 - ビジュアル テスト、選択したポイントからの段階的実行 263
 - ビジュアル テスト、選択したポイントでの再生の停止 263
 - ビジュアル テスト変数 193
- [デバッグ] メニュー
 - コマンド 619
 - ショートカット 619

と

- 同期
 - xBrowser 478
 - 設定の変更 280
 - 正しくないタイムスタンプ 504
 - について 280
- 同期オプション
 - xBrowser 574
- 統合
 - Silk Central の場所の設定 531
 - オプション 557
- 動的呼び出し
 - Android 433
 - AUT へのコードの追加、FAQ 257
 - FAQ 255
 - iOS 433
 - 概要 254
 - スクリプトの単純化 255
 - 入力引数の型が一致しない 257
 - モバイル ネイティブ 433
 - 予期しない戻り値 255
- 動的ロケータ属性
 - 詳細 515
- 同列要素
 - 検索する 319
- ドキュメント
 - 配布物 34
 - 表記規則 77
- ドメインなしでの設定
 - Oracle の設定 594
 - SQL Server 593
- ドメインなしでのデータベースの設定
 - ユーザーの作成 593
- ドライバ スクリプト
 - 概要 113

- 作成 114
- トラブルシューティング
 - Eclipse 397
 - Java SWT 397
 - Silverlight 454
 - UI オートメーション 251
 - XPath 323
 - アプリケーション構成 128
 - ハンドル無効エラー 506
 - モバイル 424
- 取り消し
 - 自動保存、アクティブ データ 149

な

- 名前の変更
 - 資産 305
 - ビジュアル テスト 305
 - プロジェクト 607, 608
- 並べ替え
 - 資産 307

に

- 入力パラメータ
 - 削除 236
 - スクリプトへの追加 235
 - 定義 67
 - 編集 236
- 入力引数の型が一致しない
 - 動的呼び出し 257
- 入力フィールド
 - 検索 507

ね

- ネイティブ モバイル
 - メソッドの呼び出し 433
- ネイティブ モバイル アプリケーション
 - Android、前提条件 398
 - iOS、前提条件 405
 - 制限事項 431
- ネイティブ再生
 - 比較、API 再生 479
- ネイティブなユーザー入力
 - 記録オプション、設定 566
 - 利点 479
- ネスト
 - キーワード 524

は

- バージョン
 - ページ 300
- ハイブリッド アプリケーション
 - Android 399
 - iOS 409
- ハイライト
 - 再生中のオブジェクト 573
- 配列パラメータ

- 例 234
- パスワードを変更する
 - ログオン 26, 597
- パラメータ
 - Silk Central 206
 - 値、割り当て 204
 - 受け渡し、スクリプト内 229
 - キーワード間でのデータの受け渡し 523
 - キーワード駆動テスト クラス 1784
 - 組み込んだビジュアル テスト、値の割り当て 204
 - 再利用、ビジュアル テスト 208
 - 削除 206
 - 作成 203
 - 処理、キーワード 527
 - 数値の例 233
 - スクリプト、受け渡し 227
 - スクリプト、作成 227
 - スクリプト、ビジュアル テストでの使用 226
 - スクリプトでの作成と受け渡し 227
 - スクリプトとビジュアル テスト間での受け渡し 224
 - 定義 224
 - 配列の例 234
 - ビジュアル テスト間での受け渡し 206
 - ビジュアル テストで定義 207
 - ビジュアル テスト、使用 203
 - ビジュアル テスト、追加 203
 - ビジュアル テストの編集 205
 - ブール値の例 234
 - リストの例 232
 - 例 232
 - 列挙の例 232
 - パラメータの受け渡し
 - スクリプト間 227
 - スクリプト内 229
 - パラメータの作成
 - スクリプト 227
 - パラメータの使用
 - ビジュアル テスト 203
 - パラメータの定義
 - データ、再利用 224
 - 貼り付け
 - テスト ステップ 215
 - 判断ロジック
 - If、Else If、Else テスト ステップ 160
 - 追加、テスト ステップ 160
 - テスト ステップの編集、例のシナリオ 97
 - テスト ステップの編集、例の詳細 97
 - ビジュアル テスト、追加 158
 - 判断ロジックの編集
 - テスト ステップ、例 97
 - ハンドル無効エラー
 - トラブルシューティング 506

ひ

- ビジュアル テスト
 - 再生 48, 273
 - 実行 273
 - バージョン、比較 217
 - 比較 217
 - 2 つめのテストの記録 56

- Java Network Launching Protocol (JNLP) 85
- アクティブ データ、マップ 146
- アクティブ データ、使用 144
- アクティブ データの更新 142, 148
- アクティブ データの更新、式の使用 148
- アクティブ データの更新、自動保存の回避 149
- アクティブ データのサンプル、詳細 93
- 新しいブラウザ ウィンドウ、起動 86
- アプリケーションの起動 211
- イメージ検証を追加する 361
- 印刷 267
- インポート 309
- インポートとエクスポート 309
- エラー処理、再生 265
- エラー処理ロジック、追加 169
- エラーのトレース 290
- オブジェクト マッピング 340
- 概要 91
- 拡張 154
- 拡張、概要 51
- 拡張したテストの再生 54
- 確認 46
- 画面の更新 191
- 画面の挿入 89
- 画面プレビュー 631
- キーワードとして指定 522
- キャプチャ画面の更新、再生 191
- キャプチャ画面の更新、テスト アプリケーション 191
- 記録 44, 80
- 記録、サンプル アプリケーション 45
- 記録、出力オプション 570
- 記録、複数のアプリケーション 83
- 組み込み、実行 55
- 組み込んだスクリプト、変更 209
- 繰り返しロジック、追加 162
- 繰り返しロジックの作成、アクティブ データ 145
- グローバル変数 196
- グローバル変数、管理 197
- グローバル変数、作成 197
- グローバル変数、取得 198
- 結果 282, 291
- 結果、作成 288
- 結果コメント 291
- 結果への切り替え 290
- 現在の時刻の取得 186
- 現在の日付を取得 184
- 検索する 307
- 検証の追加 52
- 検証ロジック、追加 166
- 更新 51
- 更新されたアクティブ データの保存 149
- コメントの追加 154
- コントロール プロパティ データ 189
- 再生の設定 108
- 再生の停止、選択したポイント 263
- 削除 306
- 作成 80
- 作成、アセット ブラウザ 304
- 作成、複数のアプリケーション 83
- サンプル 93
- サンプル、アクティブ データ 93
- 自動エラー処理の設定、プロパティ ペイン 266
- 自動ステップ、手動で作成 187
- 手動による記録 90
- スクリプトでの使用 67, 68
- スクリプトにデータを渡す 224
- スクリプトの実行 209, 226
- スクリプトパラメータ、使用 226
- ステップ実行の制御、再生 264
- ステップのコピー 215
- ステップの削除、画面プレビュー 215
- ステップの削除、ストーリーボード 216
- ステップの削除、テスト ステップ ウィンドウ 216
- ステップの貼り付け 215
- ステップの表示 87
- ストーリーボード 633
- 操作の変更 154
- 操作を記録する 89
- 挿入 57
- 段階的実行、選択したポイントから 263
- 段階的実行、デバッグ モード 265
- データ、共有 203
- データ、再利用 192
- データの受け渡し 206
- テスト ステップ 214
- テスト ステップのアイコン 629
- テスト ステップの種類 629
- テスト プロパティ 627
- テスト ロジック デザイナを使用してエラーに自動的に対処するように設定する 266
- デバッグ 263
- デバッグ中の編集 264
- 名前の変更 305
- 並べ替え 307
- パラメータ、値の割り当て 204
- パラメータ、削除 206
- パラメータ、追加 203
- パラメータ、定義 224
- パラメータの編集 205
- パラメータの再利用 208
- パラメータの定義 207
- 判断ロジック、追加 158
- 判断ロジック、例 97
- 判断ロジック、例のシナリオ 97
- 判断ロジック、例の詳細 97
- 引数を eCATT スクリプトからインポートする 468
- 引数を eCATT スクリプトにエクスポートする 469
- 表示、複数バージョン 308
- 開く 91, 304
- フィルタリング 306
- 複数のステップ、選択 214
- プロパティ 632
- プロパティ ペイン、表示 632
- 変数 192
- 保存 46, 91
- 命名 46
- メッセージ ボックス 210
- モジュール式の概要 55, 74
- 予約変数 198
- 予約変数、戻り値型 199
- 予約変数、リスト 198
- 利点 92

- ローカル変数 194
- ローカル変数、作成 53
- ローカル変数、データの格納 54
- ローカル変数、編集 195
- ロケータからオブジェクト マップへの移動 84
- ロジック 157
- ビジュアルナビゲータ
 - 概要 42, 625
 - 画面プレビュー 631
 - 結果 287
 - ストーリーボード 633
 - ストーリーボードでのサムネイルの移動 634
 - テストステップのアイコン 629
 - テストステップの種類 629
 - [テストステップ] ペイン 627
 - プロパティ 632
 - レイアウトのカスタマイズ 634
- ビジュアルブレイクポイント
 - 検出 500
- ビジュアルテストの拡張
 - 画面プレビューからの更新 51
 - 検証の追加 52
 - 再生 54
 - 変数、作成 53
 - ローカル変数、データの格納 54
- ビジュアルテストの記録
 - 概要 44
 - 確認 46
 - サンプル アプリケーション 45
 - 保存 46
- ビジュアルテストの実行
 - 値の取得、設定 279
 - 組み込み 274
 - 挿入されたスクリプト、設定 280
 - 挿入されたビジュアルテスト、設定 280
- ビジュアルテストパラメータ
 - 再利用する 208
- ビジュアルテスト変数
 - ダイアログ ボックス 275
- 日付
 - FormatDate 関数 184
- 必要な権限
 - Silk Test の実行 584
 - Silk Test のインストール 584
- ビデオ
 - 表示されない 424
- 非表示
 - 入力フィールド 507
- 表示
 - プロパティ、ビジュアルテストステップ 632
 - 画面イメージ、ビジュアルテスト 632
 - 結果コメント 291
 - コメント 154
 - 資産 635
 - 資産、複数バージョン 308
 - データベース エラー 603
 - データベースへの接続ユーザー 603
 - テストステップの説明 87
 - テストステップの選択 87
 - ビジュアルテスト 633
 - ビジュアルテスト、画面プレビュー 189

- プロジェクト参照 605
- [表示] メニュー 616
- 開く
 - Access データベース 600
 - Oracle データベース 599
 - SQL Server データベース 599
 - キーワード 524
 - 資産 304
 - データベース 599
 - ビジュアルテスト 91

ふ

- ファイアウォール
 - 競合の解決 334
 - ポート番号 336
- ファイル
 - スクリプトからの削除 238
 - スクリプトでの名前の変更 238
 - スクリプトへの追加 238
- [ファイル] メニュー
 - コマンド 614
- フィルタ
 - 複製 296
- フィルタ処理
 - 適用されたフィルタの削除 296
 - テストステップの表示 87
- フィルタの適用
 - 結果 296
- フィルタリング
 - キーワード 535
 - キーワード シーケンス、アセットブラウザ 306
 - キーワード駆動テスト、アセットブラウザ 306
 - 結果 294
 - 結果、フィルタの作成 294
 - 結果、フィルタの適用 296
 - 結果、フィルタの編集 295
 - 資産 306
- ブール値パラメータ
 - 例 234
- 複数のアプリケーションのテスト
 - テスト スクリプトの記録 117
- 複数のイメージ
 - 追加、イメージ資産 359
- 複数のイメージを追加する
 - イメージ資産 359
- 複数のテストの実行
 - Apple Safari 489
- 複製
 - 資産 305
- ブックマーク
 - スクリプトでのコードの移動 235
- ブラウザ
 - 起動、スクリプト 506
- ブラウザ
 - 最大化 502
 - 定義、再生 474
- ブラウザ ウィンドウ
 - サイズの指定 499
- ブラウザ テスト
 - 再生、並列 277

- ブラウザーの設定
 - 再生 474
- ブラウザ構成設定
 - xBrowser 480
- ブラウザの起動
 - 再生 506
- ブラウザの種類
 - Chrome for Android、設定 430
 - 現在の表示 503
 - 現在の表示、GetProperty 503
 - 使用法 503
- フラグ
 - 概要 155
 - 結果への挿入 156
 - 削除する 157
 - [テスト ステップ] ペインへの挿入 155
 - 表示 156
 - 表示、開始画面 156
 - 編集 156
- フラグの設定
 - プロパティ 103
- 古い資産のバージョン
 - 削除 300
- プロジェクト
 - アクティブの変更 37
 - アプリケーション構成、編集 124
 - インポート 309
 - インポートとエクスポート 309
 - 概要 36, 604
 - 管理 36, 604
 - 削除 608
 - 作成 604
 - 参照 605
 - 参照、グローバル 605
 - 参照、表示 605
 - 循環参照、回避 606
 - 名前の変更 607, 608
 - 表示 37
 - ユーザー アクセス権、変更 610
- プロパティ
 - アクティブ データ 100
 - エラー時の移動先プロパティ 105
 - オブジェクト 105
 - 結果 109
 - 検証結果の説明 110
 - コマンド 102
 - 再生 106, 108
 - 再生設定の取得 104
 - 再生設定の設定 109
 - 再生の設定、挿入されたスクリプト 280
 - 再生の設定、挿入されたビジュアル テスト 280
 - 詳細設定 100
 - 全般 103
 - その他 104
 - 待機 110, 213
 - タイマー 109
 - 遅延 103
 - ビジュアル テスト 632
 - ビジュアル テストでのコントロール値 189
 - ビジュアル テストの再生 278
 - フラグの設定 103

- 割り当て 101
- プロパティに基づく条件
 - 定義 158, 162
- [プロパティ] ペイン
 - スクリプト 638
 - ビジュアル テスト、表示 632
- プロファイル
 - Mozilla Firefox 494
- プロファイルを編集する
 - ユーザー 610

へ

- 並列実行
 - ブラウザ 277
 - モバイル テスト 277
- 並列テスト
 - テスト済みの構成、Android 402
- ページ同期
 - xBrowser 478
- ベスト プラクティス
 - スクリプト、作成 111
- ヘッダー
 - アクティブ データ ファイル 145
- 別の場所
 - データベース 33
- ヘルプ
 - 関連資料 34
 - キーボード 37
 - トピックの印刷 78
 - 表記規則 77
 - ヘルプの表示方法 77
- [ヘルプ] メニュー 622
- 変更
 - 記録の出力オプション 568
 - グループ 612
 - 再生ホット キー 580
- 変更する
 - オプション、記録の出力 570
 - オプション、詳細設定アクティブ データ 582
- 編集
 - アクティブ データ ファイル 142
 - アプリケーション構成 125
 - アプリケーション構成、プロジェクト 124
 - 基本状態、ユーザー インターフェイス 130
 - 組み込んだスクリプト 209
 - グループ 612
 - 結果フィルタ 295
 - デバッグ中のビジュアル テスト 264
 - メニュー 615
 - ユーザー プロファイル 610
 - リモート ロケーション 126
 - ローカル変数、ビジュアル テスト 195
- 変数
 - イメージ資産名、イメージ クリック 360
 - イメージ資産名、イメージ検証 362
 - キーワード間でのデータの受け渡し 523
 - キーワード駆動テストの実行 530
 - グローバル、取得 198
 - グローバル定義、管理 197
 - グローバル定義、作成 197

- グローバル定義、取得 198
- グローバル定義、ビジュアルテスト 196
- コントロールプロパティ値 189
- 再生中の監視 193
- 削除 196
- スクリプト 220
- スクリプトでの Public と Private 220
- スクリプトでの使用 220-223
- 追加 73
- データの格納 54
- デバッグ 59
- ビジュアルテスト 192
- 予約 198
- 予約、戻り値型 199
- 予約、リスト 198
- ローカル、作成 53
- ローカル、追加 194
- ローカル定義、ビジュアルテスト 194

ほ

- ポート
 - 構成、Information Service 336
 - Open Agent 334
- 保存
 - 記録オプション 571
 - 更新データ、アクティブデータファイル 149
 - 再生オプション 580
 - ビジュアルテスト 91
- ホットキー
 - オプション 571
 - 設定 564

ま

- マージ
 - オブジェクト マップ 356
- マウス移動の詳細設定
 - 設定、クロスブラウザテスト 480, 563
- マウス イベント遅延
 - 設定 575
- マッピング
 - アクティブデータ、ステップテキスト 147
 - アクティブデータ、ビジュアルテスト 146

む

- 無応答のアプリケーションのタイムアウト
 - 設定 575
- 無効化
 - テクノロジードメイン 127
- 無視するクラス 564

め

- 命名規則
 - 資産 309
- メイン画面
 - GUI 40
- メッセージの受信

- データベース メンテナンス 603
- メッセージボックス
 - アイコン、表示 100
 - 式、例 103
 - 挿入 210
 - テキスト、表示 103
- メニュー
 - ウィンドウ 622
 - 概要 614
 - カスタマイズ 621
 - 操作 616
 - 挿入 618
 - ツール 620
 - デバッグ 619
 - 表示 616
 - ファイル 614
 - ヘルプ 622
 - 編集 615
- メンテナンス
 - データベース、接続ユーザー 603

も

- モジュール式スクリプト
 - エラー 76
 - 概要 74
 - 挿入 75
- モジュール式テスト
 - 2 つめのテストの記録 56
 - 概要 55, 74
 - 実行 55
 - 挿入 57
- モバイル
 - トラブルシューティング 424
- モバイル Web
 - iOS 408
 - 既存のテスト 435
 - 既知の問題 642
 - クリック 434
- モバイル アプリケーション
 - 記録 417
 - テスト 397
- モバイルテストデバイス
 - ネイティブ アプリ、iOS 406
- モバイルデバイス
 - Silk Central
 - モバイルデバイス 424
 - 操作する 423
 - 定義、再生 417
 - に対して操作を実行する 423
- モバイル ネイティブ アプリケーション
 - 制限事項 431
- モバイルブラウザ
 - 制限事項 430
- モバイルテスト
 - Android 398
 - iOS 404
 - Web アプリ、iOS 408
 - Web アプリ、iOS シミュレータ 408
 - 概要 397
 - 再生、並列 277

- 接続文字列 419
- デバイスの開放 423
- ネイティブ アプリ、iOS シミュレータ 407
- リモート ロケーション、追加 126

モバイル デバイスの設定

- 再生 417

モバイルの記録

- について 417

ゆ

ユーザー

- 管理 609
- 削除 610
- 追加 609
- プロジェクト アクセス権、変更 610
- プロファイル、編集 610

ユーザー インターフェイス

- 概要 31
- 関連トピック 613

ユーザー データ ディレクトリ

- Google Chrome 491

ユーザー アカウント制御

- 概要 127

ユーザー アカウントを作成する

- eCATT 461

ユーザーの設定

- Oracle 588

ユーザーの編集

- プロファイル 610

ユーザー補助

- 有効化 581
- オブジェクト解決の向上 245
- 使用法 245

よ

[要約] タブ

- 結果 284

予期しない Click 動作

- Internet Explorer 506

よくある質問

- オブジェクト マップ 356
- クロス ブラウザ テスト 502
- コードの追加、AUT 257
- 動的呼び出し 255

読み取りオプション

- アクティブ データ、設定 141

予約変数

- 概要 198
- 戻り値 199
- リスト 198

ら

ライセンス

- 利用可能なライセンスの種類 25

ライブラリ

- アップロードする 533

ランダム数

アクティブ データ 142

り

リスト パラメーター

- 例 232

リテラル値の作成

- 条件 175

リテラル プロパティ

- 値の作成 175

リモート テスト

- Google Chrome 482
- Microsoft Edge 482
- Mozilla Firefox 482
- Open Agent 339

リモート ブラウザー テスト

- 接続文字列 482

リモート ロケーション

- 追加 126
- 編集 126

リモート エージェント

- について 339

れ

例

- アクティブ データ、テスト シナリオ 93
- ビジュアル テスト 93
- ビジュアル テスト、アクティブ データ 93
- ビジュアル テスト、アクティブ データの詳細 93
- ビジュアル テスト、判断ロジック 97
- メッセージ ボックスの式 210

例のシナリオ

- ビジュアル テスト、判断ロジック 97

例の詳細

- ビジュアル テスト、判断ロジック 97

レコメンド

- アルゴリズム 527

レスポンス Web デザイン

- ブラウザ ウィンドウ、サイズの指定 499
- ビジュアル ブレークポイント、検出 500

FileSizeUnit 列挙 680

列挙パラメータ

- 例 232

列数

- アクティブ データ、ビジュアル テスト 150

連絡先情報 78, 79

ろ

ローカル変数

- 削除 196
- 作成 53
- 追加 194
- データの格納 54
- ビジュアル テスト 194
- ビジュアル テストの編集 195

ログ

- コマンド ライン 555
- 設定 555

- ログイン
 - Silk Test Workbench 26, 597
 - グオン
 - Silk Test Workbench 26, 597
 - グオン パスワード
 - 変更する 26, 597
 - ケーター
 - サポートされているサブセット 316
 - スクリプトでのサポート 316
 - xBrowser 505
 - xBrowser 内で不正 504
 - エイリアスの使用 562
 - オブジェクト タイプ 313
 - [オブジェクトの識別] ダイアログ ボックス 636
 - オブジェクト マップでの変更 346
 - カスタマイズする 318
 - 基本概念 313
 - 記録 90
 - 検索範囲 313
 - 構文 314
 - サポートしない構成子 314
 - サポートする構成子 314
 - スクリプトの構文 112
 - スクリプトのための記録 121
 - 設定 564
 - 属性の使用 314
 - デフォルト モードの設定 564
 - ビジュアル テストでのオブジェクト マップ エントリへの移動 84
 - マッピング 340
 - ケーター属性
 - Rumba コントロール 455, 512
 - UI オートメーション コントロール 249, 511
 - WPF コントロール 441, 513
 - 記録オプション、設定 566
 - Silverlight コントロール 450, 510
 - 除外される文字 135
 - 動的 515
 - ジック
 - エラー処理 169
 - 概要 157
 - ステップのコピーおよび貼り付け 215
 - ジック ツールボックス
 - 概要 171
- ## わ
- 割り当てプロパティ
 - 概要 101