MICRO FOCUS

Silk Test Workbench 21.0

.NET スクリプト入門ガイド

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

2022-10-25

目次

Silk Test Workbench スクリプト チュートリアル

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

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

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

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

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

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

スクリプトの記録：概要

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

サンプル Web アプリケーションの起動

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

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

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

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

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

サンプル Web アプリケーションに対するスクリプトの記録

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

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

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

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

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

12. **保存** をクリックします。

記録されたスクリプトの確認

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

スクリプトは、以下のサンプルのようになります。

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

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

    End Sub
End Module
```

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

記録されたテストの再生

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

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

   再生 ダイアログ ボックスが開きます。このダイアログ ボックスで、結果をどのように保存するかを指定できます。
2. **結果の説明** テキスト ボックスに Initial test results for the recorded test と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、 **実行** をクリックします。

   各結果は、一意のテスト実行番号によって識別されます。

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

結果の分析：概要

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

スクリプトの拡張：概要

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

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

検証の挿入

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

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

1. 自動見積もりのモデル タイプを定義する、以下のテキストを選択します。

   ```
   .DomElement("RX400").DomClick(MouseButton.Left, New Point(96, 11))
   ```
2. モデル (**Model**) タイプを指定する Instant Quote ウィザードのページに移動し、異なるモデル タイプに注目します。
   たとえば、**GS430** は車種 **Lexus** のモデル タイプです。
3. Silk Test Workbench でモデル タイプを変更します。
   textContents コードを

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

   から次のように変更します：

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

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

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

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

アプリケーション データの作成とローカル変数への保存

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

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

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

1. スクリプトで、電子メールの値に移動します。

   コードは以下のようになります。

   ```
   .DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
   ```
2. 電子メールの値のあとに、以下のコードを挿入します。

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

   このステップによって、新しいローカル変数 *StremailAddr* が作成されます。このローカル変数に、**Get Instant Auto Quote page** ページの電子メール アドレスのテキストを保存します。
3. 再生中に変数テキストを表示する出力を含めるには、スクリプトに Console.Write コマンドを含めます。

   以下に例を示します。

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

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

拡張したスクリプトの再生と分析

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

1. 次のいずれか 1 つのステップを行います：

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

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

   結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

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

6. **合格（1）** タブをクリックします。
   かっこ内の数は、合格した検証の合計数を示します。**テスト ステップ** ペインには検証ステップが表示され、**結果の詳細** 列には検証の合格の説明が表示されます。
7. **詳細** タブをクリックすると、各操作の結果を表示されます。
8. **出力** ウィンドウで、変数に電子メール アドレスを保存した結果を示すステップまでスクロールします。
   テキストは以下のようなものです。

   ```
   jsmith@gmail.com
   ```
9. **合格** タブをクリックすると、モデル タイプの検証結果が表示されます。
   テキストは以下のようなものです。

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

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

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

スクリプト内でのスクリプトの実行：概要

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

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

モジュール式テスト

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

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

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

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

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

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

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個の

テストをテスト「モジュール」として作成する方が有益です。 トランザクション単位ごとに別々のテストを作成し、それをテスト シナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

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

2 つめのスクリプトの記録

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

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

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

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

別のスクリプトへのスクリプトの挿入

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

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

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

1. **ファイル** > **開く** を選択します。 **アセット ブラウザ** が開きます。
2. 左側のペインで **.NET スクリプト** を選択し、スクリプトのリストを表示します。
3. リストから AutoQuote をダブルクリックして開きます。

   AutoQuote はこのチュートリアルで最初に作成したテストです。
4. **コード** ウィンドウの Public Sub Main() コードの後ろにカーソルを置き、Enter を押して、新規行を追加し、以下のように入力します。

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

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

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

再生エラーへの対応：概要

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

まず、前のレッスンで作成したモジュール式スクリプトを再生します。

モジュール式スクリプトの再生

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

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

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

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

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

結果の確認

スクリプトの結果を確認します。

1. **終了** をクリックして、再生を停止します。**再生完了** ダイアログ ボックスが開きます。
2. **結果へ移動** をクリックします。AutoQuote の結果が表示され、デフォルトで **要約** タブが表示されます。

   要約 タブにはテスト実行の全体的な詳細が表示されます。**ビジュアル テストまたは .NET スクリプト (実行回数)** フィールドには、「AutoQuote(1)」と挿入されたスクリプト「AddAccount(1)」がリストされます。
3. **詳細** タブをクリックします。
4. 青色のテキストのステップまでスクロールします。

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

   

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

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

```
smith@test.com
```

を次のように変更します：

```
psmith@test.com
```

索引

き

け

さ

す

て

ひ

へ

も