



# Silk Performance Explorer 21.0

[ヘルプ](#)

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

© Copyright 1992-2020 Micro Focus またはその関連会社。

MICRO FOCUS、Micro Focus のロゴおよび Silk Performer は、Micro Focus またはその関連会社の商標または登録商標です。

その他、記載の各名称は、各所有社の知的所有財産です。

2020-10-21

# 目次

<b>Performance Explorer21.0</b> .....	<b>4</b>
<b>入門</b> .....	<b>4</b>
Performance Explorer の概要 .....	4
用語 .....	4
時系列データ ファイル .....	5
Performance Explorer を起動する .....	7
UI のツアー .....	8
<b>結果の分析</b> .....	<b>9</b>
結果を追加する .....	10
結果を可視化する .....	11
結果を分析する .....	11
結果を処理する .....	14
結果を表示する .....	17
<b>レポートを作成する</b> .....	<b>18</b>
Word 概要レポート .....	18
HTML 概要レポート .....	19
比較レポート .....	20
<b>リアルタイム監視</b> .....	<b>22</b>
監視技術 .....	22
データ ソースを監視する .....	48
負荷テストを監視する .....	71
監視グラフと監視出力 .....	72
ホストデータ .....	73
CSV から TSD にデータを変換する .....	74
<b>組み込みの測定値</b> .....	<b>75</b>
要約測定値 .....	75
トランザクションの測定値 .....	80
Web フォーム測定値 .....	80
Web ページ測定値 .....	81
ブラウザ駆動型の測定値 .....	81
CORBA 固有の測定値 .....	82
SQL データベース固有の測定値 .....	83
ミドルウェア固有の測定値 .....	83
Citrix 固有の測定値 .....	84
カスタム タイマ測定値 .....	84
カスタム カウンタ測定値 .....	84
ストリーミングの測定値 .....	84
<b>コマンド ラインパラメータ</b> .....	<b>85</b>
使用できるコマンド .....	85
コマンド ファイル .....	87
<b>表示をカスタマイズする</b> .....	<b>88</b>
デフォルト オプション .....	88
グラフ表示をカスタマイズする .....	91
グラフの線表示をカスタマイズする .....	92
測定値の限界を表示する .....	92
パフォーマンス レベルの条件を表示する .....	93

# Performance Explorer21.0

Performance Explorer 21.0 へようこそ。

Performance Explorer では、リアルタイム監視を通じて取得した測定値を参照したり、完了した負荷テストの結果を分析できます。さまざまなツールと機能によって、キャプチャしたすべてのデータに対して包括的な分析、レポート、処理を行うことができます。

## 入門

### Performance Explorer の概要

Performance Explorer には 2 種類の主要な機能があります。

- 結果の分析
- リアルタイム監視

#### 結果の分析

Performance Explorer の結果の分析機能を使用すると、完了した負荷テストの結果をグラフ、表、レポートを利用して分析できます。グラフを使用すると、負荷テスト中に収集したデータを可視化でき、レポートや表は、重要なデータをまとめたり見つけたりするのに役立ちます。すべてのグラフ、表、レポートの基礎をなすのが .tsd ファイルのデータです。各負荷テスト中に、Silk Performer は大量のデータをキャプチャし、いくつかの時系列データ (.tsd) ファイルに保存します。負荷テストが完了すると、Performance Explorer でデータ (.tsd ファイル) を読み込み、要求に従って編集できます。すべてのグラフが完全にカスタマイズ可能で、必要な数の測定値を含めることができます。1 つまたは複数のテストからの測定値を使用して、複数のグラフを開き、類似性や差異を表示できます。Performance Explorer は、グラフ、表、レポートのさまざまなテンプレートを提供します。さらに、クライアントとサーバー パフォーマンスに関する情報を 1 つのグラフに置くことができるので、サーバー パフォーマンスがクライアントの動作に与える影響を直接見ることができます。変更をテンプレートとして保存すれば、個々の設定を再利用できます。

#### リアルタイム監視

Performance Explorer のリアルタイム監視機能を使用すると、システムのパフォーマンスをリアルタイムに表示するグラフを作成、設定することにより、幅広いシステムを監視できます。同時に複数のグラフを開き、複数のシステム (Web サーバーのパフォーマンスとオペレーティング システムのパフォーマンスなど) のパフォーマンスを同時に監視することができます。グラフに測定値を追加することは、Performance Explorer では直感的で簡単です。追加する測定値 (複数でも可) を、ツリーからグラフにドラックするだけです。

## 用語

以下の用語は、Performance Explorer および Performance Explorer ヘルプ全体で使用されます。誤解を避け、説明を明確かつ分かりやすくするため、これらの用語をここで定義します。

用語	説明
グラフ	グラフは測定値をグラフで視覚的に表現する描画領域です。グラフは、すべての表示されるグラフの線をまとめたもの示すこともあります。

用語	説明
グラフの線	グラフの線は測定値をグラフィカルに表現したものです。グラフに測定値をドラッグすると、特定の色、スタイル、幅を持つグラフの線として表示されます。
時系列	時系列は、ある測定値の一連の値です。例：測定値 アクティブ ユーザー数 は、2、4、6、8 の値を持ちます。このような一連の値が時系列です。
測定値	測定値は負荷テスト中、または特定の測定ポイントに対する監視セッション中に収集される値をグループ化した最小のエンティティです。測定値は名前と型を持ちます。測定値に関連して測定された一連の値が時系列データ (.tsd) ファイルに格納されます。例：送信されたリクエストやトランザクション (ビジー) OK [秒] は測定値です。
レポート	レポートには、負荷テストの特定の結果や重要な所見をまとめることができます。レポートは、説明テキストを持つ表とグラフで構成されます。Performance Explorer では、HTML 概要レポート、Word 概要レポート、比較レポート (地域、ユーザー タイプ、カスタム比較レポート) の各種レポートを作成できます。
表	表は、負荷テスト中や監視セッション中に収集したデータを表形式で表示します。
監視グラフ	監視グラフにはリアルタイムで更新されるグラフを表示します。動的な監視グラフは監視セッションで使用され、静的なグラフは完了した負荷テストの結果分析で使用されます。
監視出力	監視出力は、監視セッションのテキストを含んだグラフィカルな要約です。
システム	システムは、監視可能なマシン、テクノロジー、アプリケーションです。Performance Explorer は、監視対象として幅広い種類のシステムをサポートします。

## 時系列データ ファイル

TSD ファイル	ファイル名のパターン	説明
ユーザー ファイル	u@<...>.tsd	<b>ユーザー ファイル</b> には、 <b>1 人の仮想ユーザー</b> から得られた時系列データが含まれています。負荷テスト中に、ユーザー ファイルはエージェント上で生成されます。
エージェント ファイル	a@<...>.tsd	<b>エージェント ファイル</b> には、1 つのエージェントから得られた時系列データが含まれています。負荷テスト中に、エージェント ファイルはエージェント上で生成されます。
中間ファイル	i@<...>.tsd	<b>中間ファイル</b> には、 <b>1 つのエージェント</b> から得られたすべての時系列データ (そのエージェントで動作している <b>すべての仮想ユーザー</b> から得られた時系列データを含む) が含まれています。負荷テストが完了すると、エージェント上で、ユーザー ファイルとエージェント ファイルがマージされて中間ファイルが生成されます。マージされると、中間ファイルはコントローラに転送されます。

TSD ファイル	ファイル名のパターン	説明
マージされたファイル	m@<...>.tsd	<b>マージされたファイル</b> には、 <b>すべてのエージェント</b> から得られた時系列データが含まれています。コントローラ上で、中間ファイルがマージされて生成されます。マージされたファイルは、マスター ファイルであり、負荷テストのすべての時系列データが含まれています。
ユーザー タイプごとのエージェントファイル	t@<...>.tsd	<b>ユーザー タイプごとのエージェント ファイル</b> には、 <b>1つのエージェントの1つのユーザー タイプ</b> から得られた時系列データが含まれています。ユーザー ファイルと混同しないようにしてください。いくつかの仮想ユーザーを1つのユーザー タイプに割り当てることができます。つまり、ユーザー タイプごとのエージェント ファイルには、通常、いくつかのユーザー ファイルから得られた時系列データが含まれています。負荷テストが完了すると、エージェント上で、同じユーザー タイプのユーザー ファイルがマージされてユーザー タイプごとのエージェント ファイルが生成されます。マージされると、ユーザー タイプごとのエージェント ファイルはコントローラに転送されます。
ユーザー タイプごとのマージされたファイル	k@<...>.tsd	<b>ユーザー タイプごとのマージされたファイル</b> には、 <b>すべてのエージェントの1つのユーザー タイプ</b> から得られた時系列データが含まれています。コントローラ上で、同じユーザー タイプのエージェント ファイルがマージされて生成されます。
地域ファイル	n@<...>.tsd	<b>地域ファイル</b> には、 <b>1つのクラウド地域</b> から得られた時系列データが含まれています。コントローラ上で、同じ地域のクラウド エージェントの中間ファイルがマージされて生成されます。
リアルタイムファイル	r@<...>.tsd	<b>リアルタイム ファイル</b> には、サーバー側監視中に収集され計算された時系列データが含まれています。



User File



Agent File



Intermediate File



Merged File



Agent File per User Type



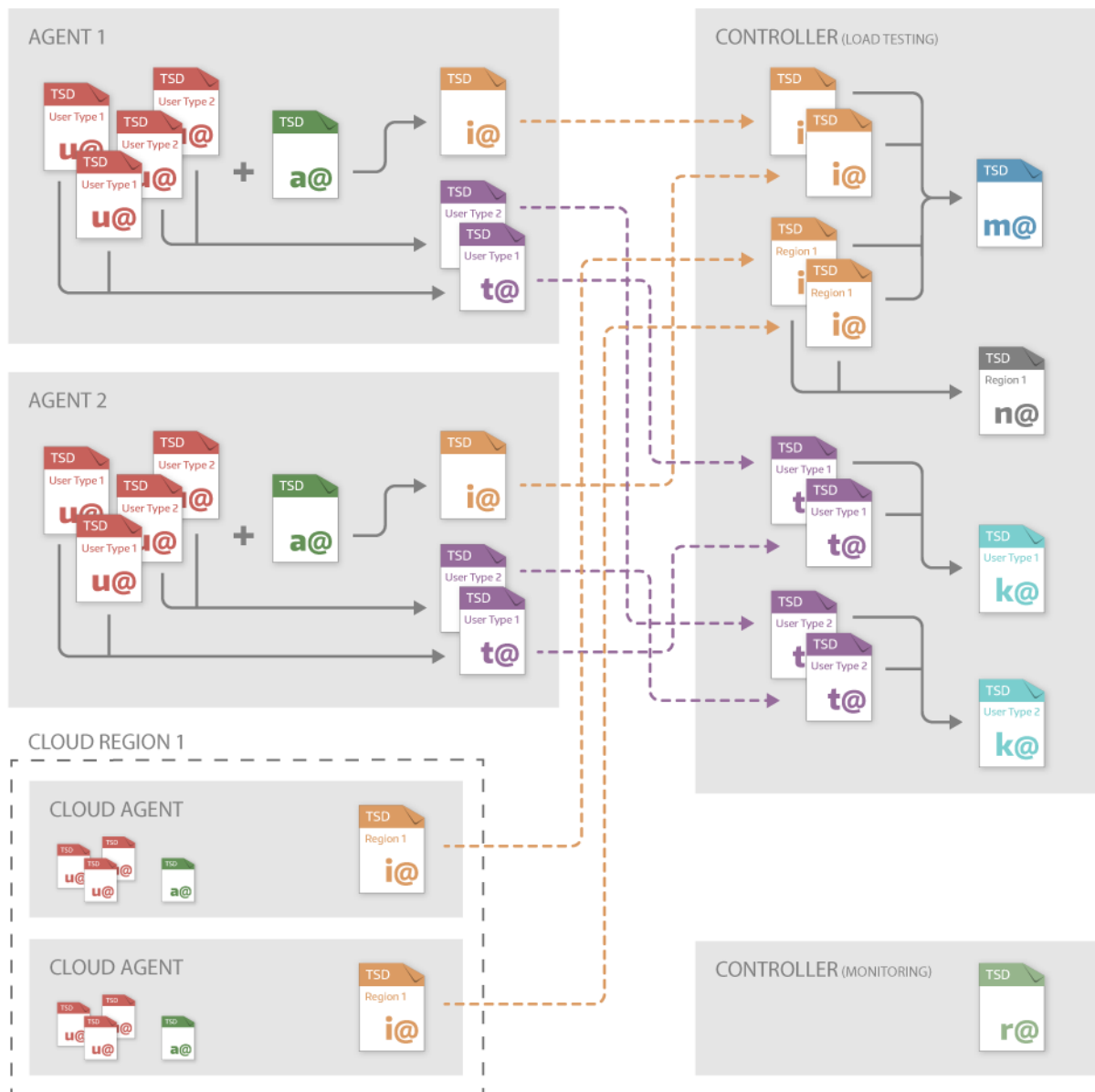
Merged File per User Type



Region File



Real-Time File



## Performance Explorer を起動する

Silk Performer での作業を始める前に、アプリケーションがどのようにセットアップされ、配置されているかを知っておく必要があります。たとえば、どのマシンでデータベース サーバーやアプリケーション サーバーを実行するかを確認します。

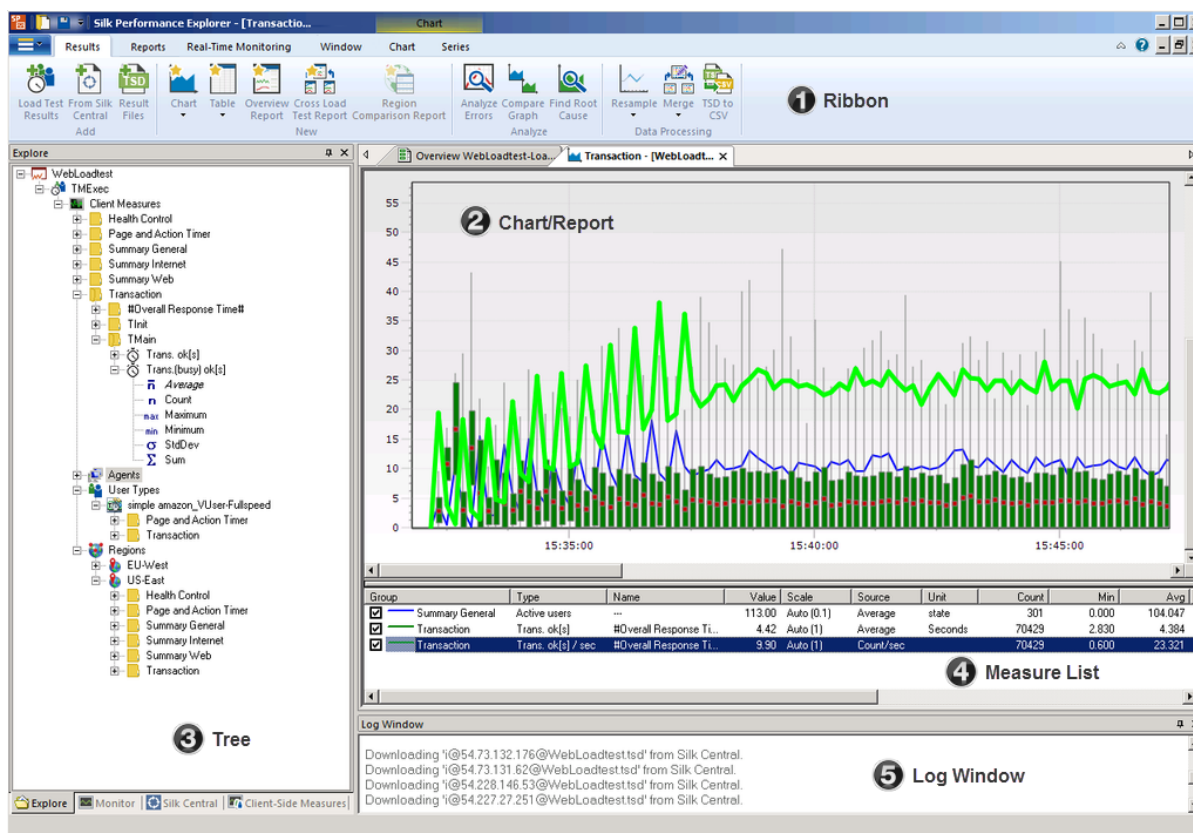
Performance Explorer を起動する方法は、いくつかあります。

- Windows スタートメニューから：**スタート > すべてのプログラム > Silk > Silk Performer <バージョン> > 分析ツール > Silk Performance Explorer** をクリックします。
- Silk Performer から：**結果 > サーバーの監視** をクリックします。
- Silk Performer から：**結果 > 時系列の検討** をクリックします。
- Silk Performer から：**ツール > Silk Performance Explorer** をクリックします。
- コマンドライン ツールから。

新しいテンプレートを編集するか、Silk Performer の結果メニューから Performance Explorer を起動すると、プログラムでは定義済みの監視グラフをいくつかと、定義済みのデータ ソースを 1 つ開きます (ローカル マシン)。

## UI のツアー

Performance Explorer ユーザー インターフェイスの主要なセクションを以下に示します。



### (1) リボン

Performance Explorer のリボンには、4 つの固定タブと、2 つのコンテキスト タブがあります。固定タブは常に表示されますが、コンテキスト タブはグラフやレポートが開かれ、アクティブな場合にのみ表示されます。固定タブは次の通りです。

- **結果** : このタブには、結果の表示と分析に関するすべてのコマンドがあります。
- **レポート** : このタブには、レポートの作成と作業に関するすべてのコマンドがあります。
- **リアルタイム監視** : このタブには、Performance Explorer のリアルタイム監視機能に関するすべてのコマンドがあります。
- **ウィンドウ** : このタブには、ウィンドウの表示方法 (並べて表示、重ねて表示) やレイアウトのリセットなどの、いくつかのウィンドウ コマンドがあります。

コンテキスト タブは次の通りです。

- **グラフ** : このタブは、グラフがアクティブな場合に表示されます。ここでは、グラフやグラフの線の表示を制御するすべてのコマンドがあります。
- **時系列** : このタブは、グラフがアクティブな場合に表示されます。ここでは、時系列の表示を制御するすべてのコマンドがあります。



## (2) グラフ/レポート

この領域には、グラフやレポートが表示されます。ツリーからグラフに測定値を簡単にドラッグできます。コンテキスト タブのコマンドをしようして、グラフ、時系列、レポートの表示を調整できます。例：グラフの色や線の幅、スタイルを変更したり、倍率の有効化、マーカーや NULL 値や測定値の限界の表示などを行います。

## (3) ツリー

ツリー領域は、次の 4 つのツリーで構成されます (ツリー タブは、ツリー領域の下部に表示されます)。**検討** ツリーには、負荷テストまたは単一の .tsd ファイルの測定値が表示されます。**監視** ツリーには、監視されているシステムの測定値が表示されます。**Silk Central** ツリーには、Silk Central から呼び出され管理されている負荷テストの測定値が表示されます。**クライアント側測定値** ツリーには、クライアント側測定値が表示されます。

ツリーからグラフに測定値をドラッグできます。ツリーの余白をダブルクリックして負荷テストやシステムを追加することができます。

## (4) 測定値リスト

測定値リストは、グラフが開かれ、アクティブな場合に表示されます。グラフに表示されているすべての測定値を一覧できます。チェック ボックスをクリックして、測定値のグラフ上での表示を切り替えることができます。このリストでは複数選択もサポートされているため、複数またはすべての測定値の表示を切り替えることも可能です。

## (5) ログ ウィンドウ

ログ ウィンドウには、エラーやクエリ状態など、パフォーマンス監視中に発生するイベントがすべて一覧表示されます。**ウィンドウ** タブでログ ウィンドウの表示を切り替えることができ、また、ログ ウィンドウを右クリックして、ログのコピーやクリアを行うことができます。

# 結果の分析

Performance Explorer の結果の分析機能を使用すると、完了した負荷テストの結果をグラフ、表、レポートを利用して分析できます。グラフを使用すると、負荷テスト中に収集したデータを可視化でき、レポートや表は、重要なデータをまとめたり見つけたりするのに役立ちます。すべてのグラフ、表、レポートの基礎をなすのが .tsd ファイルのデータです。各負荷テスト中に、Silk Performer は大量のデータをキャプチャし、いくつかの時系列データ (.tsd) ファイルに保存します。負荷テストが完了すると、Performance Explorer でデータ (.tsd ファイル) を読み込み、要求に従って編集できます。すべてのグラフが完全にカスタマイズ可能で、必要な数の測定値を含めることができます。1 つまたは複数のテストからの測定値を使用して、複数のグラフを開き、類似性や差異を表示できます。Performance Explorer は、グラフ、表、レポートのさまざまなテンプレートを提供します。さらに、クライアントとサーバー パフォーマンスに関する情報を 1 つのグラフに置くことができるので、サーバー パフォーマンスがクライアントの動作に与える影響を直接見ることができます。変更をテンプレートとして保存すれば、個々の設定を再利用できます。

Performance Explorer には特別なワークフローはありませんが、通常、次のような手順で行います。

- **追加** : Performance Explorer ワークスペースに結果を追加します。
- **可視化** : 読み込んだ結果を基にして、グラフ、表、レポートを作成します。
- **分析** : 結果を分析します (エラーの分析、グラフの比較、根本原因の特定によって)。
- **処理** : 結果のリサンプルまたはマージ、およびデータをエクスポートします。

## 結果を追加する

結果の表示、分析、処理を始める前に、Performance Explorer に結果を追加する必要があります。一般的な方法は、Silk Performer の **負荷テストの要約** ページにあるグラフやボタンをクリックして、直接結果を開くことです。

しかし、Performance Explorer ワークスペースを空の状態起動して、次のように結果を追加することもできます。

- 指定した負荷テストのすべての結果を追加する
- Silk Central から結果をインポートする
- 単一の .tsd ファイルを追加する

### 負荷テストの結果を追加する

負荷テストの結果を追加する場合、結果フォルダを指定して、すべての .tsd ファイルを Performance Explorer ワークスペースに追加できます。

1. Performance Explorer を開きます。
2. **結果** タブの **追加** グループで、**負荷テストの結果** をクリックします。
3. プロジェクトディレクトリと負荷テストを指定して、**追加** をクリックします。

追加した結果は **検討** ツリーに表示されます。



**ヒント:** ツリーの余白をダブルクリックしたり、ツリーを右クリックしてコンテキストメニューのコマンドを使用して、結果を追加することもできます。

### Silk Central から結果を追加する

Silk Central から結果をインポートしたり (以下で説明します)、Silk Central から直接 Performance Explorer を起動できます。

1. Performance Explorer を開きます。
2. **結果** タブの **追加** グループで、**Silk Central から** をクリックします。
3. ウィザードの指示に従って、結果ファイルをワークスペースに追加します。

追加した結果は **Silk Central** ツリーに表示されます。



**ヒント:** ツリーの余白をダブルクリックしたり、ツリーを右クリックしてコンテキストメニューのコマンドを使用して、結果を追加することもできます。

### 単一の結果ファイルを追加する

単一の .tsd ファイルを選択して (他の負荷テストからでも可)、Performance Explorer ワークスペースに結果を追加できます。

1. Performance Explorer を開きます。
2. **結果** タブの **追加** グループで、**結果ファイル** をクリックします。
3. ウィザードの指示に従って、結果ファイルをワークスペースに追加します。

追加した結果は **検討** ツリーに表示されます。



**ヒント:** ツリーの余白をダブルクリックしたり、ツリーを右クリックしてコンテキストメニューのコマンドを使用して、結果を追加することもできます。

## 結果を可視化する

Performance Explorer ワークスペースに結果を追加すると、グラフ、表、レポートを用いてそれらを可視化することができます。

### グラフを作成する

1. Performance Explorer を開きます。
2. **結果** タブの **新規** グループで、**グラフ** をクリックして、空のグラフを作成します。
3. ダイアログで .tsd ファイルを選択します。  
このダイアログは、ワークスペースに結果を追加していない場合にのみ表示されます。
4. ツリーからグラフに測定値をドラッグします。



**ヒント: グラフ** ドロップダウン矢印をクリックし、**定義済みのグラフ** をクリックして、テンプレートからグラフを作成するか、または **空の仮想ユーザー ベースのグラフ** をクリックして、X 軸上に仮想ユーザー数を表示する空のグラフを作成することもできます。

### 表を作成する

1. Performance Explorer を開きます。
2. **結果** タブの **新規** グループで、**表** をクリックして、空の表を作成します。
3. ダイアログで .tsd ファイルを選択します。  
このダイアログは、ワークスペースに結果を追加していない場合にのみ表示されます。
4. ツリーから表に測定値をドラッグします。



**ヒント: 表** ドロップダウン矢印をクリックして、**定義済みの表** をクリックすると、テンプレートから表を作成することもできます。

### レポートを作成する

1. Performance Explorer を開き、結果をワークスペースに追加します。
2. **レポート** タブの **概要レポート** および **比較レポート** グループで、次のボタンのいずれかをクリックします。
  - **Word**
  - **HTML**
  - **地域**
  - **ユーザー タイプ**
  - **カスタム**

各種レポートについての詳細は、「レポートを作成する」セクションの対応するトピックを参照してください。

## 結果を分析する


Performance Explorer には、負荷テストの結果を分析するために使用できる複数の機能があります。**エラーの分析** 機能を使用すると総合的なエラー分析を実行でき、**グラフの比較** 機能を使用するとさまざまな負荷テストの測定値を比較できます。また、**根本原因の特定** 機能を使用することもできます。

### エラーを分析する

Performance Explorer には、負荷テスト中に発生したエラーの分析に役立つ機能があります。概要レポートにはエラーの情報が表示されますが、**エラーの分析** 機能では、総合的なエラー グラフにすべてのエラーが表示され、詳細な情報が一覧できます。

以下の手順を実行する前に、Silk Performer で負荷テストを設定し、実行してください。

1. 負荷テストが完了すると、**負荷テストの要約** ページが表示されます。**次のステップ** 領域にある **エラーの分析** をクリックします。Performance Explorer が開き、エラー グラフが表示されます。
2. **負荷テストの結果** グラフをクリックしたり、**次のステップ** 領域にある **負荷テストの分析** をクリックすることもできます。Performance Explorer が開きます。その後、**結果** タブの **分析** グループで、**エラーの分析** をクリックします。エラー グラフが開きます。グラフを拡大して、必要な特定の情報を掘り下げることができます。

 **ヒント:** Performance Explorer ワークスペースを空の状態でも起動することもできます。この場合、**結果** タブの **分析** グループで、**エラーの分析** をクリックして、.tsd ファイルを指定します。

## エラー分析グラフ

エラー分析グラフには、以下の情報が表示されます。

表示	説明
エラー グループ	<b>エラー グループ</b> ペインには、分析対象の負荷テスト中に発生したすべてのエラーのグループ情報が表示されます。ズーム機能とシフト機能を利用すれば、特定の時間枠における情報を表示できます。
エラーの詳細	<b>エラーの詳細</b> ペインには、選択した時間枠内に発生したエラーがすべて一覧表示されます。エラーをダブルクリックするか、右クリックしてコンテキストメニューから <b>TrueLog の表示</b> を選択すると、そのエラーの視覚的情報が含まれているそれぞれの TrueLog ファイルを用いて TrueLog Explorer が起動され、選択したエラーの根本原因を突き止めるのに役立ちます。
凡例	<b>凡例</b> ペインには、グラフに表示されている曲線が一覧表示されます。これは、デフォルトではエラー分析グラフのエラー要約にすぎません。また、発生回数、最小値、最大値などに関する詳細な情報も表示されます。カーソルをグラフの上に置くと、特定の時間枠でのエラーの発生回数が表示されます。カーソルをグラフの上で移動させると、それに合わせて <b>値</b> 列が更新されます。


## 根本原因を特定する

この作業を行う前に、グラフのデータソースとして使用される時系列データ (TSD) ファイルを指定します。

Performance Explorer の自動結果相関機能は、クライアント側の問題を、対応するサーバー側の測定値と相関させることによって、ネットワークとサーバーのボトルネックに関する根本原因の分析を容易にします。自動結果相関は、特定のクライアント側のエラーに最も関連が深いサーバー側測定値を識別します。それによって、サーバー側の問題が容易に特定され、デバッグ作業が迅速になります。結果相関は、逆の場合も機能します。つまり、サーバー側の問題をクライアント側の測定値と相関させることもできます。

自動結果相関は、主要測定値を従属測定値と統計的に相関させます。たとえば、サーバー レスポンス時間の大幅な増加がクライアント側の 18 時 20 分 (午後 6 時 20 分) の測定値によって検知されると、クライアント側のパフォーマンスの落ち込みに寄与したサーバー側の測定値を、自動結果相関によって特定できます。

1. 分析する測定値のグラフ内でクリックして、カーソルを右にドラッグして、分析する時間枠を選択します。

 **注:** 時間枠を時間軸上で前後に移動させるには、時間枠を右クリックしマウスを右または左にドラッグします。時間枠は、Y 軸にそって垂直方向にも移動できます。



**注:** 分析のためにより短い期間を選択する場合は、時間枠を右にドラッグします。分析のためにより長い期間を選択する場合は、時間枠を左にドラッグします。

2. ワークフロー バーの **根本原因の特定** をクリックします。

代替方法 : グラフを右クリックして、**根本原因の分析** を選択します。

**根本原因の特定 - 相関関係の設定** ダイアログ ボックスが開きます。基本測定値が、**基本測定値** ボックスに表示されます。

3. **相関先** リスト ボックスで、基本測定値と相関を算出する測定値の種類を選択します。
4. 適切なリスト ボックスで新しい [開始日時] および [終了日時] を選択して、[日時の選択] 設定を調整します。  
時間軸で選択した時間枠に基づいて、[日時の選択] 設定が自動的に定義されます。
5. **結果** 領域で、一致する度合いに基づいて測定値をフィルタ処理する方法を指定します。  
以下のオプションのいずれかを選択します。

- **相関の最も強い測定値** をクリックして、テキスト ボックスに返される測定値数を指定します。
- **相関関係度の下限** をクリックして、テキスト ボックスに値を入力して最小相関関係度を指定します。

6. **次へ** をクリックします。



**注:** [クライアントおよびサーバーの測定値]、[クライアントの測定値]、または [サーバーの測定値] に対して相関させる場合は、ステップ 10 に進みます。

**相関関係にある測定値** ページが開きます。

7. [カスタム] 測定値に相関させる場合は、相関させる測定値グループを定義する必要があります。**測定値グループ** チェック ボックスで、相関させる対象のグループをオンにします。
8. 省略可能 : **ファイルの追加** をクリックし、時系列データ ファイルを参照して選択すると、その他の測定値を追加できます。  
測定値を選択して、**ファイルの削除** をクリックすると、測定値を削除できます。
9. **次へ** をクリックして、相関を実行します。**相関関係付けの結果** ページが開きます。  
デフォルトでは、返された測定値は相関関係の度合いの順序で一覧表示され、すべての相関が選択されます。
- 10 **相関グループ** および **名前** 列見出しをクリックして、返った測定値をソートします。
- 11 相関グラフから除外する測定値のチェック ボックスをオフにします。
- 12 **完了** をクリックして、相関関係グラフを生成します。

## グラフを比較する

関心のある測定値に対するグラフをカスタマイズして作成するとき、さまざまなユーザー タイプ、エージェント、地域あるいは負荷テストの測定値に対して、これらの測定値を比較したい場合があります。Performance Explorer では、この目的のために **グラフの比較** 機能を提供します。**グラフの比較** をクリックすると、Performance Explorer は選択した測定値に対する新しいグラフを作成し、比較対象の測定値を自動的に追加します。

1. 新しいグラフを作成します。**結果** タブの **新規** グループで、**グラフ** をクリックします。
2. 負荷テストの結果ファイルを Performance Explorer にまだ読み込んでいない場合には、ダイアログで .tsd ファイルを選択します。
3. ツリーから調べたい測定値をグラフにドラッグします。
4. **結果** タブの **分析** グループで、**グラフの比較** の下にあるドロップダウン矢印をクリックし、次のオプションから 1 つを選択します。
  - **他のユーザー タイプ** : 作業中のグラフで選択中の測定値で新しいグラフを作成して、すべてのユーザー タイプから同じ測定値を追加します。
  - **他のエージェント** : 作業中のグラフで選択中の測定値で新しいグラフを作成して、すべてのエージェントから同じ測定値を追加します。

- **他の地域** : 作業中のグラフで選択中の測定値で新しいグラフを作成して、すべての地域から同じ測定値を追加します。
  - **他の TSD ファイル** : 作業中のグラフで選択中の測定値で新しいグラフを作成して、異なる TSD ファイルの同じ測定値を追加します。
5. 他の .tsd ファイルとグラフを比較することを選択した場合は、Performance Explorer は .tsd ファイルを選択するためのダイアログを表示します。

## 結果を処理する

Performance Explorer を使用して、さまざまな方法で結果を処理できます。

- 間隔を変更したり時間枠を再定義して、結果をリサンプルできます。
- 複数の負荷テストの結果をマージできます。
- 結果をエクスポートして、表計算ソフトなどのサードパーティ ツールで処理できます。

## 負荷テストの結果をリサンプルする

負荷テストの結果フォルダを指定して、含まれている .tsd ファイルのデータを間隔を変更したり、時間枠を再定義してリサンプルできます。

1. **結果** タブの **データ処理** グループで、**負荷テスト結果のリサンプル** をクリックします。
2. リサンプルする .tsd ファイルを含む負荷テストの **ソースディレクトリ** を指定します。**ターゲットフォルダ** の名前を指定し、**次へ** をクリックします。
3. **出力間隔** を指定します。長時間実行した負荷テストの場合、出力間隔を大きくすると対象ファイルが小さくなるので好都合な場合があります。
4. 以下のマージタイプのいずれかを選択します。
  - **相対** をクリックすると、すべての時系列が同時に開始することを指定できます。
  - **絶対** をクリックすると、時系列ごとに開始時刻が異なることを指定できます。

別々の負荷テストから得られたファイルをマージする場合、それらのファイルの開始時間は異なるため、**相対** マージタイプを使用します。

**絶対** を選択した場合は、時間枠を指定できます。この時間枠内のデータのみ、出力ファイルにマージされます。

5. 新しい開始日時と終了日時を選択して、日時の設定を変更します。  
時間軸で選択した時間枠に基づいて、[日時の選択] 設定が自動的に定義されます。
6. **次へ** または **完了** をクリックすると、リサンプル処理が開始されます。

## 単一の結果ファイルのリサンプルする

結果ファイルを指定して、含まれているデータを間隔を変更したり、時間枠を再定義してリサンプルできます。出力間隔を増加させると、結果ファイルのサイズが小さくなり、対応するグラフが滑らかになります。

1. **結果** タブの **データ処理** グループで、**単一結果ファイルのリサンプル** をクリックします。
2. **ソースファイル** を指定します。これは、リサンプルする .tsd ファイルです。**ターゲットファイル** の名前を指定し、**次へ** をクリックします。  
ターゲットファイルの書式は次のようになります : m@@<sourcefilename>\_r.tsd
3. **出力間隔** を指定します。長時間実行した負荷テストの場合、出力間隔を大きくすると対象ファイルが小さくなるので好都合な場合があります。
4. 以下のマージタイプのいずれかを選択します。
  - **相対** をクリックすると、すべての時系列が同時に開始することを指定できます。
  - **絶対** をクリックすると、時系列ごとに開始時刻が異なることを指定できます。

別々の負荷テストから得られたファイルをマージする場合、それらのファイルの開始時間は異なるため、**相対** マージタイプを使用します。

**絶対** を選択した場合は、時間枠を指定できます。この時間枠内のデータのみ、出力ファイルにマージされます。

5. 新しい開始日時と終了日時を選択して、日時の設定を変更します。  
時間軸で選択した時間枠に基づいて、[日時の選択] 設定が自動的に定義されます。
6. **次へ** または **完了** をクリックすると、リサンプル処理が開始されます。

## 負荷テストの結果をマージする

大規模な負荷テストを実行するには、複数のコントローラから負荷テストを開始する必要があります。結果として、Silk Performer は複数の結果フォルダ (コントローラごとに 1 つのフォルダ) を生成します。**結果のマージ** 機能を使用して、複数の結果を 1 つの結果フォルダにマージできます。これにより、単一の包括的レポートにすべての測定値を表示することができるようになります。

 **注:** 異なるプロジェクトからの結果をマージすることはできません。

1. **結果** タブの **データ処理** グループで、**結果のマージ** をクリックします。
2. 2 つ以上の負荷テスト結果フォルダをファイル エクスプローラからテーブルにドラッグするか、**追加** をクリックして結果フォルダを参照します。
3. **マスター** 列で、マスター負荷テストを指定します。結果ファイルからの値にはマージできないものもあります (プロジェクトの説明など)。この場合、Performance Explorer は、マスター負荷テストの値を使用します。
4. **マージタイプ** を選択します。
  - **絶対** : レポートのタイムラインで実際の開始時間が結果に表示されます。たとえば、コントローラ A で負荷テストを開始した 10 秒後に、コントローラ B で負荷テストを開始した場合、結果はその結果に基づいてタイムライン上に表示されます。
  - **相対** : 負荷テストの実際の開始時間に関わらず、レポートのタイムライン上に同じ開始時間ですべての結果が表示されます。
5. マージされた .tsd ファイル用の **間隔** を指定します。最小の間隔 (**間隔** リストに表示される) は、指定した結果フォルダにあるすべての .tsd ファイルの間隔の最小公倍数です。
6. **ターゲット フォルダ** を指定します。マージ結果がこのフォルダに保存されます。
7. **マージ** をクリックします。


結果が正しくマージされると、Performance Explorer は概要レポートを表示します。この新しく作成されたレポートのワークロードタイプは **Merged Workload** であり、すべての実行したユーザーグループを含みます。

## 単一の結果ファイルをマージする

単一の結果ファイルのマージは、次の状況において有用です。

- 選択したソース .tsd ファイル群をもとにしてカスタム .tsd ファイルを作成する。
  - マージした .tsd ファイル (m@...tsd) を再作成して、負荷テスト中に接続が失われたエージェントの結果を取り込む。コントローラとエージェントの接続が失われた場合、エージェント ファイルはローカルの結果ディレクトリにあります。接続を失ったタイミングにより、ユーザー ファイルのみ、またはマージされたエージェント ファイルが残っています。これらの結果を負荷テスト コントローラの間接ファイルに取り込むには、手動でこれらのファイルをマージする必要があります。
1. 必要に応じて、マージするすべてのファイルを単一のディレクトリにコピーします。
  2. **ウィザード** > **カスタム マージ ウィザード** を選択します。**カスタム マージ ウィザード** が開きます。
  3. すべてのファイルが保存されているディレクトリを選択し、マージするファイルをすべて選択します。
  4. **完了** をクリックします。

すべてのデータを含む新しいマージされたファイルが生成されます。

 **注:** 同じまたは依存したデータを含む .tsd ファイル群を選択しないでください。たとえば、ユーザータイプ .tsd ファイルとマージしたユーザータイプ .tsd ファイルをマージすると、同じ時系列に対して複数の集計が行われるため、通常、意味のあるシナリオではありません。

## TSD ファイルを CSV ファイルとしてエクスポートする

**TSD から CSV** 機能を使用して、.tsd (時系列データ: Time Series Data) ファイルの内容を .csv (カンマ区切り値: Comma Separated Values) ファイルにエクスポートできます。その後、外部の表計算ソフトでデータを分析したり、編集したりできます。

コマンドライン ツールの TSD File Export Tool を使用することもできます。これにより、エクスポート処理を自動化できます。

1. **結果** タブの **データ処理** グループで、**TSD から CSV** をクリックします。
2. **ソース ファイル** を選択し、**エクスポート ファイル** の名前と場所を指定して、**次へ** をクリックします。
3. エクスポートする測定値を選択し、**次へ** をクリックします。
4. **エクスポート タイプ** を選択します。
  - **標準** をクリックした場合、.csv ファイルに含める必要がある結果ファイルのデータ ソース要素を指定します。
  - **ダンプ** をクリックした場合は、すべてのメタ データが .csv ファイルに含まれます。
5. .csv ファイルにヘッダーを含める場合は、**ヘッダー情報を含める** をオンにします。.csv ファイルで使用する区切り文字として、カンマまたはセミコロンいずれかを選択し、小数点として使用する文字を選択します。
6. **完了** をクリックし、データをエクスポートして .csv ファイルを生成します。
7. 出力ファイルを表示する場合は、その後に表示されるダイアログで **はい** をクリックします。

## TSD File Export Tool を使用する

TSD File Export Tool は、.tsd ファイルの内容を .csv ファイルにエクスポートするのを自動化できるコマンドライン ツールです。エクスポート設定のユーザー インターフェイスを提供する Performance Explorer の **TSD から CSV** 機能を使用することもできます。

1. Microsoft Windows のスタート メニューからツールを開きます：**スタート > すべてのプログラム > Silk > Silk Performer <バージョン> > 分析ツール > TSD File Export Tool**。
2. コマンド プロンプトで、以下に示された構文に基づいてファイル エクスポートの指定を入力します。

```
Tsd2Csv tsdFile [csvFile] [-Delimiter char] [-DecimalPoint char] [-dump]
```

コマンド	説明
tsdFiles	エクスポートする .tsd ファイルのパスと名前を入力する。
[csvFile]	作成する .csv ファイルのパスと名前を入力する。
[-Delimiter char]	区切り文字を定義する。
[-DecimalPoint char]	出力ファイルで小数点を表す記号を定義する。
[-dump]	-DecimalPoint パラメータを無視する。

3. Enter を押して .csv ファイルを生成します。

### 例

```
Tsd2Csv "input.tsd" "output.csv" -Delimiter "," -DecimalPoint "."
```



## 結果を表示する

Performance Explorer は、グラフ、レポート、表を HTML ドキュメントに変換して、Web ブラウザで表示できます。これにより、グラフ、レポート、表の各種ドキュメントへの取り込み、オンラインでの公開、または便利な形式での印刷を行うことができます。

### Web ブラウザでグラフを表示する

Web ブラウザに表示するグラフを作成するか、開きます。

1. Web ブラウザに表示するグラフを作成するか、開きます。
2. **グラフ** タブの **データ** グループで、**ブラウザで表示** をクリックします。
3. グラフの **タイトル** と **説明** を入力し、**OK** をクリックします。タイトルと説明は、ブラウザでグラフの上部に表示されます。

### Web ブラウザでレポートと表を表示する

Web ブラウザに表示するレポートを作成するか、開きます。

1. Web ブラウザに表示する HTML 概要レポートまたは表を作成するか、開きます。
2. **レポート** タブの **タスク** グループで、**ブラウザで開く** をクリックします。
3. レポートまたは表の **タイトル** と **説明** を入力し、**OK** をクリックします。タイトルと説明は、ブラウザでグラフの上部に表示されます。

### グラフをエクスポートする

Performance Explorer にアクセスできない人に見せるために、さまざまな形式でグラフをエクスポートできます。

1. エクスポートするグラフを作成または開きます。
2. **グラフ** タブの **グラフ** グループで、**エクスポート** の下にあるドロップダウン矢印をクリックします。
  - グラフ データを .csv ファイルにエクスポートするには、**CSV ファイル** をクリックします。表示されるダイアログで、.htm、.mht、.jpg などの種々のファイル形式から選択します。.csv 形式がデフォルト値です。
  - グラフ データを Microsoft Excel 文書にエクスポートするには、**Excel 文書** をクリックします。グラフは、Excel で自動的に作成されます。



**ヒント:** .mht 形式は、電子メールでグラフを送信するのに適しています。.html 形式は、Web でグラフを公開するのに適しています。.csv 形式は、外部の表計算ソフトでデータを処理するのに適しています。画像形式で保存すると、他のドキュメントに画像としてグラフを挿入できます。


### レポートと表をエクスポートする

Performance Explorer にアクセスできない人に見せるために、さまざまな形式でレポートや表をエクスポートできます。

1. エクスポートするレポートや表を作成または開きます。
2. **レポート** タブの **タスク** グループで、**名前をつけて保存** をクリックします。
3. 開いたダイアログで、ディレクトリを選択し、ファイル名を入力し、ファイルの形式を選択してから、**保存** をクリックします。




**ヒント:** .mht 形式は、電子メールでレポートや表を送信するのに適しています。.html 形式は、Web でレポートや表を公開するのに適しています。.csv 形式は、外部の表計算ソフトでデータを処理するのに適しています。

 **ヒント:** コマンドライン インターフェイスからレポートをエクスポートすることもできます。これにより、エクスポート処理を自動化することができます。

## コマンドライン インターフェイスでレポートをエクスポートする

HTML レポートは、コマンドライン インターフェイスを使って Web アーカイブ (MHT ファイル) としてエクスポートすることもできます。レポートのエクスポートには、コマンドライン パラメータ /EXPORTOVR:<target file> を使います。

このパラメータは、対象ファイルのファイル拡張子の指定に従って、アクティブな OVR を HTML または MHT ファイルとして保存します。

 **注:** 最後にロードされた TSD ファイル、つまりコマンドライン (/TSD:<tsd file>) で指定された最後の TSD ファイルがエクスポートに使用されます。

/ACTION:OVERVIEWREPORT[:<template>] パラメータでは、レポートのテンプレート ファイルを指定できます。

コマンドラインによる Web アーカイブ ファイルのエクスポートの例を次に示します。

```
/TSD:<tsdFile> /ACTION:OVERVIEWREPORT /EXPORTOVR:c:¥test.mht
```

## レポートを作成する


レポートを作成することは、負荷テスト中に収集したデータを要約し、分析するための 1 つの手段です。Performance Explorer が提供するレポートの種類と、レポートを作成して作業する方法について学びます。

### Word 概要レポート

Performance Explorer では、Microsoft Word 文書で負荷テスト結果を表示できます。プレゼンテーション用に結果をさらに加工したり、カスタマイズした書式設定を追加する場合に特に有用です。

#### Word 概要レポートを作成する

1. Performance Explorer ワークスペースに負荷テストの結果を追加します。
2. レポート タブの **概要レポート** グループで、次のボタンのいずれかをクリックします。
  - **Word** または **Word 概要レポート** をクリックして、結果を Word 文書で表示します。
  - **Word 概要レポート (テンプレートから)** をクリックして、.docx ファイルを選択し、**作成** をクリックします。
  - **カスタム Word 概要レポート** をクリックして、Word 概要レポートに含まれるべき情報を選択し、**作成** をクリックします。


 **注:** 今後のすべてのレポートにテンプレートを使用する を有効にすると、常にテンプレートに基づいて Word 概要レポートが作成されます。このオプションを無効にするには、Performance Explorer のアプリケーション ボタン (左上にある) をクリックし、**オプション** をクリックします。**レポート作成** タブをクリックし、**Word 概要レポートの新規作成時にテンプレート (.docx) を使用する** をオフにします。

#### Word 概要レポートのテンプレートを作成する

テンプレートを保存して使用する前に、レポートに表示したいすべての情報を含む Word 概要レポートを作成します。

1. レポート タブの **概要レポート** グループで、**カスタム Word 概要レポート** をクリックします。
2. テンプレートに含まれるべき情報を選択し、**作成** をクリックします。Microsoft Word が開き、指定した情報が表示されます。

- 必要に応じて Word ドキュメントの内容や体裁を適用します。情報やコメントの追加、フィールドの再配置、要素のスタイルの変更、ヘッダー、フッター、ロゴの変更、不要な内容の削除などを行うことができます。
- Word 文書を .docx ファイルとして保存して閉じます。これで、テンプレートが作成されました。
- Performance Explorer に戻ります。**レポート** タブの **概要レポート** グループで、**Word 概要レポート (テンプレートから)** をクリックします。
- 定義した .docx ファイルを参照し、**作成** をクリックします。

 **注: 今後のすべてのレポートにテンプレートを使用する** を有効にすると、常にテンプレートに基づいて Word 概要レポートが作成されます。このオプションを無効にするには、Performance Explorer のアプリケーション ボタン (左上にある) をクリックし、**オプション** をクリックします。**レポート作成** タブをクリックし、**Word 概要レポートの新規作成時にテンプレート (.docx) を使用する** をオフにします。

## HTML 概要レポート

負荷テストが完了すると、**負荷テストの要約** ページが Silk Performer に表示され、概要レポートが自動的に生成されます。概要レポートは、Silk Performer の **結果** ツリーにあります。Performance Explorer で概要レポートを開くには、ツリーでレポートをダブルクリックするか、**負荷テストの要約** ページにある **負荷テストの分析** をクリックします。また、空の Performance Explorer ワークスペースを開き、**レポート** タブの **概要レポート** グループで **HTML** をクリックします。

### HTML 概要レポートを作成する


以下の手順を実行する前に、Silk Performer で負荷テストを設定し、実行してください。負荷テストが完了すると、**負荷テストの要約** ページが表示されます。次のいずれか 1 つのステップを行います：


- 次のステップ領域にある **負荷テストの分析** をクリックします。
- ツリーの下部にある **結果** をクリックし、負荷テスト ノードを展開して **概要レポート** をダブルクリックします。

Performance Explorer が開き、概要レポートが表示されます。

Performance Explorer から HTML 概要レポートを開くには、以下の手順を実行します。

- Performance Explorer ワークスペースに負荷テストの結果を追加します。
- レポート** タブの **概要レポート** グループで、次のボタンのいずれかをクリックします。
  - HTML** または **HTML 概要レポート** をクリックして、結果を HTML 概要レポートで表示します。
  - HTML 概要レポート (テンプレートから)** をクリックして、.ovt ファイル (概要レポート テンプレート) を選択し、**作成** をクリックします。
  - カスタム HTML 概要レポート** をクリックして、HTML 概要レポートに含まれるべき情報を選択し、**作成** をクリックします。


 **注: 今後のすべてのレポートにテンプレートを使用する** を有効にすると、常にテンプレートに基づいて HTML 概要レポートが作成されます。このオプションを無効にするには、Performance Explorer のアプリケーション ボタン (左上にある) をクリックし、**オプション** をクリックします。**レポート作成** タブをクリックし、**HTML 概要レポートの新規作成時にテンプレート (.ovt) を使用する** をオフにします。

 **注:** デフォルトでは、概要レポートは自動的に生成されます。自動生成を無効化するには、Performance Explorer アプリケーション ボタン (左上にある) をクリックし、**オプション** をクリックしてから、**レポート作成** タブをクリックして、**概要レポートを自動的に生成する** をオフにします。

### HTML 概要レポートのテンプレートを作成する

HTML 概要レポートのテンプレートは、.ovt ファイルとして保存されます。


1. Performance Explorer で、HTML 概要レポートを作成または開きます。
2. レポート タブの 概要レポート グループで、**カスタマイズ** をクリックします。
3. テンプレートに含まれるべき情報を選択し、**カスタマイズ** をクリックします。
4. レポート タブの 概要レポート グループで、**テンプレートとして保存** をクリックします。
5. テンプレートを .ovt ファイル（概要レポートテンプレート）として保存します。

 **注:** HTML 概要レポートを Performance Explorer で作成するときに、常にこのテンプレートを使うように設定することができます。Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。**レポート作成** タブをクリックし、**HTML 概要レポートの新規作成時にテンプレート (.ovt) を使用する** をオンにして、ファイルを指定します。

## 概要レポート テンプレートをプロジェクトに割り当てる

Silk Performer プロジェクトに対して概要レポート テンプレートを指定することができます。自動的に生成されるすべての HTML 概要レポートは、このテンプレートを使用し、手動で HTML 概要レポートを作成するときはテンプレートをあらかじめ選択します。

1. Silk Performer のメニュー ツリーで **プロファイル** ノードを展開します。
2. 構成対象のプロファイルを右クリックして、**プロファイルの編集** を選択します。  
代替方法：**設定 > アクティブ プロファイル** を選択します。  
**プロファイル - [<プロファイル名>] - シミュレーション** ダイアログ ボックスが開きます。
3. 左側のショートカット リストで、**結果** アイコンをクリックします。
4. **時系列** タブをクリックします。
5. **概要レポート テンプレート** セクションでテンプレートを参照して、**開く** をクリックします。
6. **OK** をクリックします。

 **注:** Performance Explorer のコマンドライン インターフェイスには、Silk Performer プロジェクトで使用する概要レポート テンプレートを割り当てるための `/OVT:<テンプレート>` コマンドも用意されています。

## 比較レポート

Performance Explorer は、3 種類の比較レポートを提供します。

- 地域比較レポート
- ユーザー タイプ比較レポート
- カスタム比較レポート

比較レポートを使用すると、テスト実行の結果を最大 8 件まで並べて比較できます。その構造は、標準のベースライン レポートの構造とよく似ています。つまり、比較レポートには、要約、トランザクション、エラー、ページ、アクション タイマーに関する統計情報、Web フォーム、順位、その他多くの測定値が含まれます。

比較レポートの下部に、**ベースライン ペイン** が表示されます。このペインには、特定の地域またはユーザーの名前、または全体という名前の列（行がすべてのテスト結果を含む場合）のいずれかの列が含まれます。エージェント 列には、指定した各地域でテストに使用されたエージェントの数が表示されます。

任意のテスト結果をベースラインとして定義できます。ほかのすべての結果は、このベースラインをリファレンスとして使用し、ベースラインと比較されます。レポート順位では、ベースライン結果は常に 1 番目にリストされます。**b** アイコンまたは **ベースライン** タグによって、ベースライン地域を識別します。ベースラインに対して評価されたすべての測定値は、温度分布グラフで確認できます（レポートの色つき棒グラフ）。カーソルを温度分布グラフの上に移動すると、詳細な情報が表示されます。結果をベースラインとして設定するには、それを右クリックして **ベースラインとして設定** をクリックします。

## 地域比較レポート

地域比較レポートを使用すると、クラウド ベースの負荷テストの結果を比較できます。これらは、クラウド エージェントを使用して負荷を生成した負荷テストです。さまざまな地域の結果をならべて比較できます。1 行ごとに 1 つの地域がレポートに表示されます。最大 8 つの地域を 1 つの地域比較レポートに含めることができます。

結果全体と地域固有の結果の両方を比較することができます。ツリーから測定値をレポートにドラッグして追加することもできます。**全体** ノードをレポートにドラッグして、結果全体を追加できます。

## ユーザー タイプ比較レポート

ユーザー タイプ比較レポートによって、複数のユーザー タイプを並べて比較することができます。1 行ごとに 1 つのユーザー タイプがレポートに表示されます。最大 8 つのユーザー タイプを 1 つのユーザー タイプ比較レポートに含めることができます。

結果全体とユーザー タイプ固有の結果の両方を比較することができます。ツリーから測定値をレポートにドラッグして追加することもできます。**全体** ノードをレポートにドラッグして、テストの結果全体を追加できます。データセットは、ベースライン ペインのユーザー タイプ列で、全体 とマークされます。

## カスタム比較レポート

カスタム比較レポートを使用すると、テスト実行の結果を最大 8 件まで並べて比較できます。さらに、負荷テストのユーザー タイプまたは地域での実行を比較することもでき、ユーザーのシナリオによっては役立てることができます。

## 地域比較レポートを作成する

以下の手順を行う前に、Silk Performer でさまざまな地域のクラウド エージェントを使用するクラウド ベースの負荷テストを構成し、負荷テストを実行してください。

1. クラウド ベースの負荷テストが完了すると、**負荷テストの要約** ページが表示されます。**負荷テストの結果** グラフをクリックすると、グラフが Performance Explorer で開かれます。

結果が Performance Explorer に読み込まれ、ツリーに表示されます。

2. **地域** ノードが表示されるまで、ツリーのノードを展開します。結果データ (時系列データ) は、負荷テストに含まれたクラウドの地域それぞれに対して利用できます。



**注:** データは、単一のクラウド エージェントに対してではなく、クラウド地域に対して利用できます。

3. 地域を選択し、**レポート** タブの **比較レポート** グループで **地域** をクリックします。選択した地域を表示するレポートが表示されます。ツリーから他の地域をレポートにドラッグすることもできます。

- すべての地域を一度にレポートに表示するには、**地域** ノードを選択してから、**地域** をクリックします。
- レポートにすべての地域と **全体** 列を表示するには、**クライアントの測定値** ノードを選択し、**地域** をクリックします。



**ヒント:** ツリーのノードを右クリックして、コンテキスト メニューのコマンドを使用することもできます。




**注:** 地域比較レポート追加できるのは、同じ負荷テストからの地域だけです。


4. ある地域の結果をベースラインとして定義するには、レポートの下部にあるベースライン ペインで地域を右クリックし、**ベースラインとして設定** をクリックします。

## ユーザー タイプ比較レポートを作成する

1. Performance Explorer を開き、結果を追加します。結果が Performance Explorer に読み込まれ、ツリーに表示されます。

2. **ユーザー タイプ** ノードが表示されるまで、ツリーのノードを展開します。
3. ユーザー タイプを選択し、**レポート** タブの **比較レポート** グループで **ユーザー タイプ** をクリックします。選択したユーザー タイプを表示するレポートが表示されます。ツリーから他のユーザー タイプをレポートにドラッグすることもできます。
  - すべてのユーザー タイプを一度にレポートに表示するには、**ユーザー タイプ** ノードを選択してから、**ユーザー タイプ** をクリックします。
  - すべてのユーザー タイプとワークロードの全体列を表示するには、**クライアントの測定値** ノードを選択し、**ユーザー タイプ** をクリックします。


 **ヒント:** ツリーのノードを右クリックして、コンテキストメニューのコマンドを使用することもできます。

 **注:** ユーザー タイプ比較レポート追加できるのは、同じ負荷テストからのユーザー タイプだけです。

- 4. あるユーザー タイプの結果をベースラインとして定義するには、レポートの下部にあるベースライン ペインでユーザー タイプを右クリックし、**ベースラインとして設定** をクリックします。

## カスタム比較レポートを作成する

1. Performance Explorer を開き、結果を追加します。結果が Performance Explorer に読み込まれ、ツリーに表示されます。
2. ツリーのプロジェクト ノードを選択し、**レポート** タブの **比較レポート** グループで **カスタム** をクリックします。Performance Explorer に読み込まれているすべての負荷テストを使用してレポートが作成されます。負荷テストは、ツリーのプロジェクト ノードの下に表示されているものです。
3. さらに、負荷テストの結果を、ツリーからレポートにドラッグして追加できます。負荷テストだけでなく、さまざまなプロジェクトのユーザー タイプまたは地域の結果を 1 つのカスタム比較レポートに追加することもできます。最大 8 つのデータ セットを 1 つのカスタム比較レポートに含めることができます。

 **ヒント:** ツリーのノードを右クリックして、コンテキストメニューのコマンドを使用することもできます。
4. ある負荷テストの結果をベースラインとして定義するには、レポートの下部にあるベースライン ペインで負荷テストを右クリックし、**ベースラインとして設定** をクリックします。

## リアルタイム監視

Performance Explorer のリアルタイム監視機能を使用すると、システムのパフォーマンスをリアルタイムに表示するグラフを作成、設定することにより、幅広いシステムを監視できます。同時に複数のグラフを開き、複数のシステム (Web サーバーのパフォーマンスとオペレーティング システムのパフォーマンスなど) のパフォーマンスを同時に監視することができます。グラフに測定値を追加することは、Performance Explorer では直感的で簡単です。追加する測定値 (複数でも可) を、ツリーからグラフにドラックするだけです。

## 監視技術

Performance Explorer は、このセクションで説明する広範な監視テクノロジーをサポートします。

### Windows マシンの監視

このセクションでは、Windows マシンの要件とセットアップについて説明し、トラブルシューティングのヒントを示します。

#### Windows マシン監視の要件


それぞれのリモートシステムを監視するために、次の前提条件を満たしていることを確認してください。

## Windows Vista 以降のクライアント OS で動作しているマシン

リモートレジストリ サービスが、監視するマシンで実行されている必要があります。Windows Vista および Windows 7 のマシンでは、このサービスはデフォルトで実行されません。

## Windows パフォーマンス監視のトラブルシューティング

Windows ベース マシンの監視時には、以下のエラー メッセージが発生する場合があります。

エラー メッセージ	考えられる原因	解決策
コンピュータ名が見つかりません	管理者でないアカウントからリモートコンピュータを監視しようとする と、このエラー メッセージを受け取る 場合があります。	Windows NT コンピュータでリモート監視を行えるようにします。
Error=AGENT_ERROR_TCP_SE ND_REQUEST_FAILED	この問題は、組織でプロキシ自動構成 ファイルを使用してプロキシ サーバ ーにリクエストを送信するときに、そ のプロキシ サーバーの登録名ではな く自動構成スクリプト アドレスを指 定している場合に発生する可能性が あります。  組織でパッシブでないプロキシを使 用する場合は、ほとんどの場合ポート 番号も指定する必要があります。	以下のプロキシ設定を確認します。 <ul style="list-style-type: none"><li>• プロキシ サーバーの名前、ユーザ ー名、およびパスワードが正しく 入力されていること</li><li>• RSM コンピュータからプロキシ サーバーに ping できること</li><li>• RSM コンピュータからポータル コンピュータに接続できること</li></ul>
SYSTEM: 5 - アクセスが拒否されま した	CUSTOM / PERFMON オプションを 使用して Windows 2008 以降のマシ ンを監視する際に、管理者グループに 追加されたユーザーの認証情報を使 用してログインすると SYSTEM: 5 - アクセスが拒否されました エラ ーが生成されます。	Performance Explorer を実行するユ ーザーは、監視対象のリモート コンピ ュータの管理者グループに追加する 必要があります。Performance Explorer の <b>データ ソース ウィザード</b> で異なるユーザーの認証情報を指 定する場合、リモート コンピュータ上 の管理者グループにも追加する必要 があります。リモート マシンを問題 なく監視するためには、リモート マシ ンへの接続に使用したのと同じユー ザー アカウントで Performance Explorer (PerfExp.exe) を起動す る必要があります。   <b>ヒント:</b> 特定のユーザー アカ ウントで PerfExp.exe を起 動するには、Shift + 右クリ ック を押し、 <b>別のユーザーと して実行</b> を選択します。

## JMX 監視

Silk Performer では JMX (Java Management Extensions) 監視がサポートされているため、Java アプリケーション サーバーで公開される MBean 属性を監視することができます。数値データを返す MBean 属性は、Performance Explorer 監視にデータ ソースとして追加されます。



### JMX 監視の前提条件

JMX 監視の前提条件は以下のとおりです。

- Java Runtime (JVM) バージョン 1.5 以上
- 監視対象となる JVM アプリケーションには監視用のオープン ポートが必要

### JMX データ ソースに接続する

Java アプリケーション サーバーへの接続を確立して、MBean 属性を監視します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。
2. **定義済みのデータ ソースから選択する** をクリックし、**次へ** をクリックします。
3. 次のいずれか 1 つのステップを行います：
  - アプリケーション サーバー上の使用可能なすべての MBean 属性を検討するには、**Application Server** フォルダを展開して、ベンダー固有のフォルダから JMX 記録プロファイルを選択します。プロファイル名の終わりには、拡張子 `.jmx` が付いています。
  - 監視する一般的属性を含んでいる一群の MBean を検討するには、**Application Server** フォルダを展開して、ベンダー固有のフォルダから JMX 記録プロファイルを選択します。プロファイル名の終わりには、拡張子 `.ejmx` が付いています。
  - 汎用の JMX 監視データ ソースを作成するには、**Custom Data** フォルダを展開して、**JMX Data** を選択します。
4. **ホスト名** テキスト ボックスで、JMX サーバーを指定します。
5. 省略可能：**エイリアス** ボックスで、エイリアス名を指定します。  
エイリアスは、監視対象サーバーをよく説明する別名にする必要があります。個々のマシン上で測定値をグループ化することをお勧めします。  
たとえば、WebLogic と IIS が両方とも同じコンピュータにインストールされているとしましょう。どちらのサーバーも監視が必要ですが、2 つのパフォーマンス測定値は別々のメニュー ツリーに表示されなければなりません。
6. [ポート]、[ユーザー名]、およびそのユーザーの [パスワード] を指定します。
7. 汎用の JMX 監視データ ソースの作成を選択した場合は、**アプリケーション** リスト ボックスから、サポートされているアプリケーション構成を選択します。
8. **サーバー設定** をクリックします。**JMX Connection Configuration** ダイアログ ボックスが開きます。
9. アプリケーション サーバーの通信ライブラリが置かれている、アプリケーション サーバーのインストール ディレクトリを指定します。  
指定したアプリケーション サーバー インストール ディレクトリと、アプリケーション構成ファイルの相対クラスパス エントリから、JMX クライアントがその下で実行されるクラスパスが形成されます。その結果アクセス対象となるクラスパスは、**Resulting classpath** テキスト ボックスに表示されます。  
 **注:** UNC パスを使用するか、または、サーバーのアプリケーション ライブラリ ディレクトリをローカル マシンにコピーして、そのローカル コピーをアプリケーション サーバーのインストール ディレクトリとして指定することをお勧めします。それには、**App. server install directory** テキスト ボックスの横の参照 (...) ボタンをクリックします。  
 **注:** WebSphere の JMX 監視を行うには、IBM JDK を指定する必要があります。WebSphere サーバーで管理セキュリティが無効化または有効化されていて、インバウンド CSiv2 トランスポート層が TCP/IP の場合は、SUN JDK を指定することができます。
- 10 **Java home directory** テキスト ボックスに、クライアントの通信ライブラリが置かれているインストール ディレクトリを指定します。  
代替方法：参照 (...) ボタンをクリックして、適切なディレクトリに移動して指定します。
- 11 **Additional classpath** テキスト ボックスに、その他のライブラリを指定します。
- 12 **Additional JVM parameters** テキスト ボックスに、その他の仮想マシン パラメータを指定します。  
**Connection** タブと **Visualization** タブのパラメータが各サーバーについて再構成されます。
- 13 **OK** をクリックします。**接続パラメータ** ページが開きます。



14次へ をクリックしてから、完了 をクリックします。Performance Explorer が JMX サーバーに接続されます。

15JMX Data Source Browser に表示された Bean を調べます。

16完了 をクリックします。

### 監視する MBean を指定する

アプリケーション設定プロファイルは、XML ファイルとして C:\Program Files\Silk\Silk Performer 21.0\include\jmx-config に保存されます。必要に応じて、XML ファイル ディレクトリ内のデフォルト構成を変更し、追加のプロファイルをこのディレクトリに作成します。新しく作成したアプリケーション構成プロファイルは、データ ソースとして使用できます。

1. JMX Data Source browser を開きます。

**Type Tree** タブには、MBean の使用可能なすべてのカテゴリがツリー構造で表示されます。**Type List** タブには、使用可能なすべてのカテゴリが一覧形式で表示されます。選択したカテゴリの使用可能なすべての MBean が、**Beans** テキスト ボックスに表示されます。各 MBean はオブジェクト名で識別されますが、そのオブジェクト名はドメイン名と 1 つ以上のプロパティ タイプと値のペアで構成されています。

2. 必要な MBean を選択し、その MBean ノードを展開して使用可能な属性を表示します。

3. 省略可能: **Numeric Attributes** をクリックして、非数値属性をフィルタリングします。

非数値属性は、JMX 監視ではサポートされません。

4. 省略可能: 属性を右クリックして **Get Value** を選択し、任意の属性の値を検索します。

5. 省略可能: オブジェクト名を表す属性を右クリックし、**Follow Reference** を選択して、参照されている MBean へジャンプします。

Bean 属性プールに適切な属性を追加します。

### Bean 属性プールに属性を追加する

この手順を開始する前に、以下のいずれかの手順を実行します。

- 監視する適切な MBean を指定する。
- 簡易 JMX またはカスタム クエリを実行して、必要な MBean を特定する。

1. JMX Data Source Browser を開きます。

2. **Beans** テキスト ボックスで監視する属性を選択します。



**注:** 複数の Bean を選択するには、Shift または Ctrl を押します。



**注:** MBean によっては (JSR-77 ベースのものなど)、複数のサブ測定値や統計で構成されている複合属性が含まれている場合があります。再帰性の一定の深度までは、監視するサブ測定値を選択できます。

3. 選択した属性を右クリックして、**Add As** を選択します。サブメニューが開きます。

4. 属性値を監視する、次のいずれかのオプションを選択します。

- **Average** : カウント値に便利です。デフォルトでは、このオプションが選択されます。
- **Sum** : **Average** と同じですが、監視 GUI 内では別のラベルが表示されます。
- **Incremental** : 増えていく値に便利です。属性値の代わりに、監視間隔範囲内での値の変化が計算されます。

5. **閉じる** をクリックします。**表示する測定値の選択** ページが開きます。

6. 省略可能: MBean を選択して、**Add** をクリックします。Performance Explorer によって、表示可能なすべての MBean 属性がデータ ソースとして追加されます。

7. 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## Easy JMX クエリ

Easy JMX クエリを使用することで、JMX MBeanServer にホストされている MBean の最も重要な属性値にアクセスすることができます。Easy JMX クエリは、あらかじめ構成された詳細クエリで、アプリケーション サーバーの MBeanServer への接続が確立すると自動的に実行されます。Easy JMX クエリの条件に一致する MBean がすべて、JMX データ ソース ブラウザの **Beans** ボックスに自動的に表示されます。Easy JMX プロファイルは、**システムの選択** ダイアログ ボックスの **システム** メニュー ツリーにおいて、プロファイル名の後ろに (EJMX) タグが付いていることで識別できます。

### JMX クエリ言語を使用して Bean をフィルタリングする

特定のフィルタ条件に一致する MBean のみを表示することによって、JMX クエリを絞り込みます。

1. **JMX Data Source Browser** を開きます。選択した JMX MBeanServer 上の使用可能なすべての MBean が、**Beans** テキスト ボックスに表示されます。
2. **Simple Query** タブをクリックします。
3. **Domain** リスト ボックスからドメインを選択します。  
すべてのドメインを検索するには、このフィールドは **<All>** にしておきます。
4. 省略可能:**Property** フィールド リスト ボックスからプロパティを選択します。  
このリストには、ロードされた Bean に含まれているすべてのプロパティが表示されます。
5. 省略可能: **Property** フィールド リスト ボックスの右側にあるリスト ボックスから、選択したプロパティの値を選択します。  
[Query] テキスト ボックスには JMX 1.2 (JSR-003) 標準に基づいた JMX クエリで使用されるオブジェクト名が表示されます。
6. 省略可能: 指定した属性に基づいてリストを絞り込むには、**Attribute filter** テキスト ボックスに属性名を入力します。指定した属性を含む MBean のみが表示されます。
7. **Run** をクリックすると、クエリが実行されます。**Beans** テキスト ボックスには、フィルタ条件に合う MBean のみが表示されます。

### 詳細 JMX クエリを実行する


簡易クエリの組み合わせを作成して、お使いの環境のデータを監視します。

1. [JMX Data Source Browser] を開きます。
2. **Advanced Query** タブをクリックします。  
使用可能なクエリ タイプが、ウィンドウの上部に表示されます。  
たとえば、**Object Name Query** が表示されます。
3. クエリのリスト ボックスから、クエリに使用できる演算子を 1 つ選択します。  
たとえば、**OR** 演算子を選択します。
4. 下位のリスト ボックスから、演算子の値を選択します。  
たとえば、**True** を選択します。  
参照 (...) ボタンが表示され、[JMX Object Name Query] ダイアログ ボックスを表示してドメイン リストを参照できます。
5. 参照 (...) ボタンをクリックして、ドメイン リストから適切なドメイン名を選択します。たとえば、**jboss.cache** を選択します。
6. クエリが完成するまで、要素を追加していきます。  
たとえば、**\*:J2EEserver =local** を追加します。
7. **OK** をクリックします。
8. **Query** メニュー ツリーのルート要素をクリックします。
9. **Advanced Query** タブの下のペインに、クエリの説明を入力します。
- 10 **実行** をクリックします。

クエリの条件と一致するすべての MBean が表示されます。

#### 11 Save をクリックします。


このステップでは、クエリの保存先と名前を指定できます。JMX データ クエリのファイル拡張子は .jmq です。

 **注:** C:\Program Files\Silk\Silk Performer 21.0\include\jmx-config ディレクトリに保存されたクエリは、**Easy JMX** ページにも表示され、そこから直接実行できます。

### 保存したクエリを実行する

既存の JMX クエリを実行して、JMX MBeanServer にホストされている MBean の最も重要な属性値にアクセスします。

1. [JMX Data Source Browser] を開きます。
2. **Advanced Query** タブをクリックします。
3. **Load > Load From File > Load** を選択します。
4. 事前にインストールされているクエリと自分で保存したカスタム クエリのリストを参照します。  
事前にインストールされたクエリ タイプは、さまざまなサーバー タイプに対して使用できます。簡易 JMX プロファイルで使用可能なものと同じクエリ タイプがあります。
5. **実行** をクリックします。

 **注:** C:\Program Files\Silk\Silk Performer 21.0\include\jmx-config に保存されたクエリは、**Easy JMX** ページにも表示され、そこから直接実行できます。

Performance Explorer によって、クエリが実行されます。

## REXEC と UNIX データ ソース

リモート実行 (REXEC) プロトコルを用いて UNIX ベース システムに対するクエリを実行する方法について説明します。

### REXEC と UNIX データ ソースの概要

UNIX ベース システム上のデータ ソースに対しては、多くの場合、リモート実行プロトコルを用いてクエリを実行することができます。確認済みのデータ ソースが定義済みデータ ソース リストに表示されます。たとえば、Solaris システム上で **Context Switches/sec** (コンテキスト スイッチ回数/秒) を監視するとしましょう。定義済みリストからこのエントリを選択することで、リモート実行の詳細を指定しなくてもよくなります。該当するデータ ソースがリストに見つからない場合は、カスタム データ ソースを作成します。

### リモート実行

リモート実行とは、コマンドライン インターフェイスを使ってリモート ホスト上でコマンドを実行することです。シェル スクリプトを実行することもできます。シェル スクリプトを確認するには、telnet クライアントを使用して UNIX システムに接続し、コマンドやスクリプトをいくつか実行します。

### 単独実行

単独実行の場合は、新しいデータ ポイントが要求されるたびに、プライマリ ドメイン コントローラ エミュレータ (PDCE) がリモート マシンに接続しコマンドを実行します。実行間隔が短いと、この設定ではリモート マシンに負荷をかけるおそれがあります。ただし、PDCE は、最後に実行したコマンドが終了するまで、リモート マシン上で新しいコマンドを起動しません。

単独実行でない場合、PDCE は監視セッションの初めにリモート マシンに接続し、コマンドを起動して、コマンドの出力をパーサーに転送します。パーサーは、所定の監視間隔で起動され、使用可能なデータを解析します。コマンドは 1 回だけ実行され、リモート マシンへの接続は監視セッション全体を通して有効なままです。(たとえば、リモート マシン上のコマンドが終了したために) 接続が失われると、PDCE は一定の間隔を空けて再接続を試みます。

### 単独実行の例

```
ps -ef | egrep -c ".*"
```

上記のコマンドは、UNIX システム上で動作しているプロセスの数を数えるものです。telnet セッション中にこのコマンドを入力します。コマンドは値 (実行中のプロセスの数) を出力した後、終了します。このような動作を "単独実行" と呼びます。

このようなコマンドを 'while' ループで囲むことができます。

### 複数実行の例

```
while [ true ]; do // Mind the spaces
  ps -ef | egrep -c ".*";
  sleep 5;
done
```

上記の例では、プロセス数が 5 秒間隔で継続的に出力されます。このような動作を "複数実行" と呼びます。

## REXEC を使用して UNIX を検索する

リモート実行プロトコルを使用する UNIX ベースのシステムへの接続を確立し、監視する測定値を追加します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。
2. **定義済みのデータ ソースから選択する** をクリックし、**次へ** をクリックします。
3. **Custom Data** フォルダを展開し、**Rexec Data** をクリックしてから、**次へ** をクリックします。
4. **ホスト名** ボックスで、監視対象となるマシンを指定します。
5. 省略可能 : **エイリアス** ボックスで、エイリアス名を指定します。

エイリアスは、監視対象サーバーをよく説明する別名にする必要があります。個々のマシン上で測定値をグループ化することをお勧めします。

たとえば、WebLogic と IIS が両方とも同じコンピュータにインストールされているとしましょう。どちらのサーバーも監視が必要ですが、2 つのパフォーマンス測定値は別々のメニュー ツリーに表示されなければなりません。

6. [ポート]、[ユーザー名]、およびそのユーザーの [パスワード] を指定します。
7. **次へ** をクリックします。 **Rexec 測定値の追加** ページが開きます。
8. 以下のように適切な値を指定して、監視する測定値を作成します。

#### オプション

**測定値タイプ** 指定したコマンドについて、タイプ名を入力します。同じタイプ名を持つすべての測定値が、1 つのグループにアセンブルされます。

**測定値名** 指定した rexec コマンドの測定値名を入力します。この名前は、Performance Explorer のグラフに表示されます。

**平均測定値** rexec コマンドに対してこのチェック ボックスをオンにすると、平均測定値とみなされます。このオプションは、生成された TSD ファイルにのみ影響します。

**説明** 指定した rexec コマンドの追加情報を入力します。この情報は Performance Explorer に表示されます。

**コマンド** リモート マシン上でデータ収集を実行する次のような rexec コマンドを指定します : ps -ef | egrep -c ".\*"

## オプション 説明

**単独実行** このチェックボックスをオンにすると、PDCE はリモート マシンと接続し、新しいデータポイントがリクエストされるたびにコマンドを実行します。実行間隔が短いと、この設定ではリモート マシンに負荷をかけるおそれがあります。しかし PDCE は、最後に実行したコマンドが終了していなければ、リモート マシン上で新しいコマンドを開始しません。

**単独実行** チェックボックスをオフにすると、複数実行が有効になります。

**正規表現** リモート マシンから返されるデータを、所定の正規表現を使用して 1 行ずつフィルタリングします。正規表現と一致しない行は破棄されます。残りの行のみがその先のステップで使用されます。デフォルトでは、`^.*$` が入力されます。

たとえば、コマンドのレスポンスが各間隔で次のように表示されるとします。

```
XXXXXXXXX XXXXXXXXX XXXXXXXXX
          123400  103093  121092
```

意味があるのは、数字の入っている行のみです。次の正規表現により、数字と空白からなる行のみが許可されます。

```
[1234567890 ].*
```

その他の行は無視されます。

**フィールドインデックス** データ生成に使用された行は、フィールドに分割されます。分割を行うには、空白文字かカスタム区切り文字のいずれかを使用します。

このテキスト ボックスを使用して、フィールド インデックスを指定します。これは数値に変換され、次の測定値として使用されます。

このインデックスは、1 から始まるインデックスです。

**区切り文字** このテキスト ボックスを使用して、カスタム区切り文字を指定します。この文字を使用してデータ生成用の行をフィールドに分割します。

デフォルトでは、**空白を区切り文字とする** チェックボックスがオンになっています。

**空白を区切り文字とする** このチェックボックスをオンにすると、空白があるたびに区切り文字が挿入されます。

**スキップする行数** このテキスト ボックスを使用して、リモート マシンでコマンドを開始した後で PDCE が無視する行数を指定します。このオプションは、開始時にコマンドが返す不要なデータをフィルタリングして除外するのに便利です。与えられた正規表現と一致する行だけが、この設定を使用します。ある間隔で複数の行が与えられる場合、データの計算には最後の行が使用されます。

9. **追加** をクリックします。

10 測定値の収集に必要なコマンドをすべて入力するまで、前のステップを繰り返した後、**閉じる** をクリックします。**表示する測定値の選択** ページが開きます。

11 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

### 例

複数実行を定義したときに使用されていたコマンドを、少し修正した形で入力します。

```
while [true]; do
  ps -ef | egrep -c "\. *" ;
  sleep {%Interval};
done
```



**注:** コマンド `sleep {%Interval}` は、Performance Explorer が提供する数が入るブレースホルダです。この数は収集間隔を表します。たとえば、Performance Explorer は 5 秒ごとに新しいデータを収集します。

コマンドのレスポンスが各間隔で次のように表示されるとします。

```
XXXXXXXX XXXXXXXX XXXXXXXX
123400 103093 121092
```

**フィールド インデックス** 値には、定義済みの測定値に関連する列を指定します。**フィールド インデックス** に 1 を指定すると、テーブルから値 123400 が返されます。**フィールド インデックス** に 2 を指定すると、値 103093 が返されます。

上記の例の列は、空白文字で区切られていますただし、他の文字 (コロンなど) も使用できます。そのような値は、**区切り文字** テキスト ボックスで指定します。空白は、デフォルトの区切り文字です。

### 空きメモリのカスタム例

Linux コンピュータに次のコマンドを入力します。

```
cat /proc/meminfo
```

次のテーブルが返ったとします。

	total:	used:	free:	shared:	buffers:	cached:
Mem:	13108428	54214656	76869632	26947584	7938048	28471296
Swap:	70148096	3387392	66760704			
MemTotal:	128012					
	kB					
MemFree:	75068					
	kB					
MemShared:	26316					
	kB					
Buffers:	7752					
	kB					
Cached:	27804					
	kB					
SwapTotal:	68504					
	kB					
SwapFree:	65196					
	kB					

次の表では、前のテーブルから **MemFree** の値を取得するために使用するフィールド値を示します。

コマンド	<code>cat /proc/meminfo</code>
単独実行	オン
正規表現	<code>^MemFree:.*\$</code>
フィールド インデックス	2
区切り文字	<b>空白を区切り文字とする</b> をオンにします。
スキップする行数	0

### SNMP サポート

標準 SNMP サポートは、MIB が提供するすべての SDMP データ ソースに対する汎用インターフェイスを含みます。特定の測定値がオブジェクト ID (OID) とインスタンス ID (オブジェクトに複数のインスタ

スがある場合)の組み合わせを使用してアクセスされます。このアプローチの課題は、どのカウンター インスタンスが適用されるかを正確に知る必要があることです。インスタンスはインスタンス名ではなく、インスタンス ID でのみ区別されます。この課題をさらに難しくするのは、サーバーを再起動したときに、監視中のオブジェクトが動的に作成されるためインスタンス番号が変更される可能性があるという事実です。よって、監視スクリプトは、監視中の対象を動的に扱うことができなければなりません。

Silk Performer の強化した SNMP サポートの値は、OID とインスタンス ID によってではなく、OID とインスタンス名によって値にアクセスすることができます。

次の例では、Silk Performer で利用可能な強化した SNMP サポートについて説明します。この例は、事前定義された ESNMP 測定値がどのように定義されるか、そしてどのように使用されるか、さらに、追加の測定値をどのように事前定義した測定値のリストに追加するかです。

### インスタンス ID と OID によって識別する

事前定義された ESNMP で作業している最初のステップは、OID とインスタンス ID、およびインスタンス名 (WebLogic カウンターの名前) を識別することです。あるインスタンスに対する JDBC Connection Pool Runtime Active Connections Current Count は、OID でアクセスできます：

1.3.6.1.4.1.140.625.190.1.25 + <Instance ID>

強化した SNMP サポートがないと、値と現在のインスタンス ID を含むテーブルの OID を提供しなければなりません。

JDBC Connection Pool Runtime Object Name と JDBC Connection Pool Runtime Name は、人間が読める形式のインスタンスの名前を含み、OID でアクセスできます：

1.3.6.1.4.1.140.625.190.1.5 + <Instance ID> for "Object Names"

および...

1.3.6.1.4.1.140.625.190.1.15 + <Instance ID> for "Names"

ESNMP サポートは、インスタンス名を指定する、(object-) name テーブルを使ってインスタンス ID を取得する、そしてそれから、現在のカウンターに対する値を取得するという選択肢を提供します。ESNMP サポートでは、以下のものが提供される必要があります。

- 値を含んでいるテーブルの OID
- インスタンス名
- インスタンス名を含んでいるテーブルの OID

### SNMP を介してサーバーのパフォーマンスを監視する

Performance Explorer では、サーバー監視用のデータソースとして使用するために、さまざまなソースからの MIB (Management Information Base) の OID (オブジェクト ID) を選択できるようになっています。たとえば、Oracle などのデータベースシステムは、SNMP プロトコルを介して収集できる幅広いパフォーマンス データを提供します。

1. Performance Explorer ワークフロー バーの **サーバーの監視** をクリックします。 **データソースウィザード** が開き、 **システムの選択** ダイアログ ボックスが表示されます。
2. **定義済みのデータソースから選択する** オプション ボタンをクリックして、 **次へ** をクリックします。 **システムの選択** ダイアログ ボックスが開きます。
3. **システム** メニュー ツリーで、 **Custom Data** フォルダを展開します。
4. SNMP Data を選択して、 **次へ** をクリックします。 **接続パラメータ** ダイアログ ボックスが開きます。
5. **接続パラメータ** 領域で、該当するサーバー システムのホスト名または IP アドレス、ポート番号、その他のデータなど、データソースへの接続に必要な接続パラメータを指定します。  
指定したサーバーへのアクセス権を所有している必要があります。また、サーバー システム上で SNMP サービスが正しく構成されている必要があります。
6. **次へ** をクリックします。 **SNMP 測定値の追加** ダイアログ ボックスが開きます。

7. Performance Explorer に付属のコンパイル済みの MIB の中から 1 つを選択するか、**MIB の新規コンパイル** を選択してサーバー システムに同梱されていた MIB を使用します。

Silk Performer で MIB の内容を使用するには、まず MIB をコンパイルする必要があります。よく使用される MIB がコンパイル済み形式で多数用意されています。必要に応じて、新しい MIB をコンパイルできます。

コンパイル済みの MIB を選択すると、その内容がメニュー ツリーに表示されます。

8. 関心のある OID をクリックし、**値の取得** をクリックします。サーバーへの接続が確立され、実際のデータ項目が表示されます。

9. 対応するデータ項目をクリックし、**追加** をクリックします。

10. 必要な OID をすべて選択するまで、上記 2 つの手順を繰り返します。

11. **閉じる** をクリックします。 **表示する測定値の選択** ダイアログ ボックスが開きます。

12. 監視する要素を選択して、**完了** をクリックします。監視グラフには、指定した要素のサーバー パフォーマンスがリアルタイムで色分けされて表示されます。

## MIB を新規にコンパイルする

ハードウェア システムやソフトウェア システムのベンダーは、MIB (Management Information Base) によって、それぞれのシステムの特徴についての情報を提供します。この情報は、SNMP プロトコルを利用して、適切な OID (オブジェクト ID) を指定することによって取得できます。MIB ブラウザは、MIB に保存されている適切な OID を選択する際に役立ちます。

1. **SNMP 測定値の追加** ダイアログ ボックスで、サーバー システムに付属の MIB に対して **MIB の新規コンパイル** をクリックします。 **MIB の新規コンパイル** ダイアログ ボックスが開きます。

2. コンパイル対象の MIB ファイルを参照して **開く** をクリックします。

3. **コンパイル** をクリックします。コンパイルが正常に終了すると、コンパイル済みの MIB のリストに、新しい名前が追加されます。

4. メッセージ <名前>-MIB をインポートできませんでしたが返された場合は、このエラー メッセージで参照されている MIB をコンパイルします。

5. **閉じる** をクリックします。これで、SNMP を介してサーバーのパフォーマンスを監視するために、この新しい MIB を使用できます。

## SNMP データ ソースの定義

新しいデータ ソースを定義するには、REALTIME.INI ファイルを編集してはいけません。編集すると、次に Silk Performer をアップグレードするときに、変更が失われます。これを避けるために、<インストールパス>\¥Silk¥Silk Performer 15.0¥Include¥DataSrcWzd サブフォルダに新しいファイル (myESNMP\_realtime.ini など) を作成します。このファイルには、データ ソースを定義する行 S= と M= を含む必要があります。

"WebLogic" と "ESNMP" を REALTIME.INI ファイルで検索すると、次のエントリが見つかります：

```
S= Application Server¥BEA WebLogic preconfigured¥SNMP, ESNMP:BEA-WEBLOGIC-MIB,
```

```
M= ESNMP:1.3.6.1.4.1.140.625.340.1.25, BEA WebLogic¥JVM Runtime¥HeapFreeCurrent, eAvgOnlyCounter, 0, kbytes, 0, 1, 2, 3, 4, 5, Current Heap Free, measureMatchName=Runtime Object Name;measureMatchOID=1.3.6.1.4.1.140.625.340.1.5;
```

```
M= ESNMP:1.3.6.1.4.1.140.625.340.1.30, BEA WebLogic¥JVM Runtime¥HeapSizeCurrent, eAvgOnlyCounter, 0, kbytes, 0, 1, 2, 3, 4, 5, Current Heap Size, measureMatchName=Runtime Object Name;measureMatchOID=1.3.6.1.4.1.140.625.340.1.5;
```

```
M= ESNMP:1.3.6.1.4.1.140.625.180.1.25, BEA WebLogic¥Queue Runtime ¥ExecuteThreadCurrentIdleCount, eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Thread Idle Count, measureMatchName=Queue Runtime Object Name;measureMatchOID=1.3.6.1.4.1.140.625.180.1.5;
```

```
M= ESNMP:1.3.6.1.4.1.140.625.180.1.35, BEA WebLogic¥Queue Runtime ¥PendingRequestCurrentCount, eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Current Requests,
```



```
measureMatchName=Queue Runtime Object
Name;measureMatchOID=1.3.6.1.4.1.140.625.180.1.5;
```

```
M= ESNMP:1.3.6.1.4.1.140.625.180.1.40, BEA WebLogic¥Queue Runtime
¥ServicedRequestTotalCount, eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Total Requests,
measureMatchName=Queue Runtime Object
Name;measureMatchOID=1.3.6.1.4.1.140.625.180.1.5;
```

```
M= ESNMP:1.3.6.1.4.1.140.625.430.1.50, BEA WebLogic¥WebApp Component Runtime
¥OpenSessionsCurrentCount, eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Open sessions,
measureMatchName=Runtime Object
Name;measureMatchOID=1.3.6.1.4.1.140.625.430.1.25;
```


```
M= ESNMP:1.3.6.1.4.1.140.625.430.1.55, BEA WebLogic¥WebApp Component Runtime
¥OpenSessionsHighCount, eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Session High Count,
measureMatchName=Runtime Object
Name;measureMatchOID=1.3.6.1.4.1.140.625.430.1.25;
```

```
M= ESNMP:1.3.6.1.4.1.140.625.550.1.25, BEA WebLogic¥Execute Queue¥ThreadCount,
eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Thread Count, measureMatchName=Queue
Object Name;measureMatchOID=1.3.6.1.4.1.140.625.550.1.5;
```

S= で始まる最初の行は、データソース定義ヘッダーです。データソースごとに1つの定義ヘッダーだけが必要となります。つまり、既存の WebLogic ESNMP データソースを強化したい場合には、データソース定義ヘッダーはそのままにしておくことができます

それ以降の行は、個々の測定値用の定義です。

- M= に続くのは OID で、インスタンス ID なしにクエリするためのものです (1.3.6.1.4.1.140.625.190.1.25 など)。
- 次に、GUI に表示される測定値の名前が続きます。バックスラッシュは、階層を作成するために使用されることに注意してください (BEA¥JDBC Connection Pool¥Active Connections など)。
- そして、標準エントリが続きます : measure class、scale、unit、 scale factor、および 5 個の未使用フィールド (eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5 など)。
- 次のフィールドは、Performance Explorer のツールチップ ヘルプでの説明です。「Current count of active connections in the JDBC connection pool runtime.」のように記述できます。測定値の名前自身は短くすることができます。
- 最後のエントリは、2 つの部分に分割されます。
  - omeasureMatchName= これは、インスタンスを説明するためにデータソースウィザードで使用されます (「JDBC Connection Pool Runtime Object Name」など)。
  - omeasureMatchOID= これは、インスタンス名を含むテーブルの OID です (1.3.6.1.4.1.140.625.190.1.15 など)。

 **注:** 最終的な測定値名は、ここで指定した測定値名 (BEA WebLogic¥JDBC Connection Pool ¥Active Connections) と丸括弧で囲ったインスタンス名から構成されます。

つまり、測定値に対するキーは、次のようになります。

```
M= ESNMP:1.3.6.1.4.1.140.625.190.1.25, BEA WebLogic¥JDBC Connection Pool¥Active
Connections, eAvgOnlyCounter, 0, , 0, 1, 2, 3, 4, 5, Current count of active connection
in the JDBC connection pool runtime, measureMatchName=JDBC Connection Pool Runtime
Object Name;measureMatchOID=1.3.6.1.4.1.140.625.190.1.15;
```

## Secure Shell (セキュア シェル) を介したパフォーマンス監視

Performance Explorer を使用すると、Secure Shell を用いてパフォーマンス指標を取得できます。たとえば、CPU 使用率やメモリ使用率の統計データを SunOS システムから取得します。Performance Explorer には、Secure Shell を使った任意の監視のスク립トを作成できる高度に構成可能なパフォーマンス監視プロジェクトが付属しています。この処理には、Silk Performer 内から Performance Explorer

プロジェクトを操作します。この機能では、システムのパフォーマンスをリアルタイムで追跡することもできます。

Silk Performer プロジェクトは、Ssh.sep という名前の SilkEssential パッケージにあり、これは C:\Program Files\Silk\Silk Performer 21.0\Monitors にあります。

### セキュア シェルを使用してパフォーマンスを監視する

セキュア シェルを使用して監視するシステムは、セキュア シェル デモンを実行している必要があります。監視に使用するユーザーが、複数回ログオンする権限を持っていることを確認します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。**システムの選択** ページが開きます。
4. **Miscellaneous** フォルダを展開して、**Secure Shell** をクリックします。
5. **次へ** をクリックします。

**接続パラメータ** ページが開きます。

6. **ホスト名** テキスト ボックスに、監視するホストを指定します。
7. **次へ** をクリックします。**属性設定** ページが開きます。
8. 以下の監視固有の属性を定義します。

- **Username** : セキュア シェルを使用してリモート システムにログオンするときに使用するユーザー。
- **Password** : ユーザーのパスワード。
- **Command(x)** : 前に定義されたパッケージにより、セキュア シェルを使用して、5 つまでパフォーマンス測定値を取得できます。次の表に示します。

属性	タイプ	値
Command1	string	ps -ef   egrep -c ¥".*¥"
Command1.Active	boolean	true
Command1.Column	number	1
Command2	string	mpstat   head -n 2   tail -1
Command2.Active	boolean	true
Command2.Column	number	13

9. **OK** をクリックします。**表示する測定値の選択** ページが開きます。

- 10 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

#### 例 Command2

この例は、SunOS システムで機能します。次のコマンドがサーバーに送信されます。

```
mpstat | head -n 2 | tail -1
```

このコマンドからは、次の形式のレスポンスが 1 行返されます。

```
0 0 0 0 228 25 106 1 0 0 0 219 0 0 0 99
```

属性 Command2.Column で、パフォーマンス カウンタを追跡するために Performance Explorer に渡す列を指定します。この場合は、列 13 を渡しますが、SunOS システムでのユーザー時間比率を表します。

Command2.Active は、この測定値を収集するかどうかを指定するだけです。

#### 例 Command1

次のコマンドがサーバーに送信されます。

```
ps -ef | egrep -c ¥".*¥"
```

このコマンドは、SunOs システム上で動作しているプロセスの数を数えるものです。このコマンドからは、次の形式のレスポンスが 1 行返されます。

```
87
```

この場合、1 行には 1 列しかありません。

属性 Command1.Column では、Performance Explorer に転送する列を指定します。この場合、この列は 1 です。

Command1.Active は、この測定値を収集するかどうかを指定するだけです。

## BDL 監視プロジェクトを生成する

Silk Performer プロジェクト (具体的には BDL スクリプト) を使用して、監視対象システムのパフォーマンス データを収集できます。このセクションでは、Performance Explorer で使用する BDL 監視プロジェクトを組み立て、収集したデータのリアルタイム表示を有効にするために必要な手順について説明します。

### Silk Performer BDL 監視プロジェクト

Performance Data Collection Engine (PDCE) は、BDL スクリプトで収集されるパフォーマンス データを処理します。このようなスクリプトで収集されるパフォーマンス データは、Performance Explorer にリアルタイムで表示できます。PDCE で収集されるエンティティは測定値と呼ばれ、次の関数を使用して BDL に指定することができます。

- MeasureInc
- MeasureIncFloat
- MeasureStart および MeasureStop
- MeasureSetTimer
- MeasureSet

これらの関数はトランザクション内で呼び出されます。各トランザクション内では、1 対多の測定値を使用できます。また、監視プロジェクトでは、ユーザー グループを 1 つだけ定義するスクリプトを 1 つだけ使用できます。このユーザー グループでは、1 つの init トランザクション、1 つの end トランザクション、および 1 つの main トランザクションを定義できます。

bdlMonitor.bdh で定義されるラッパー関数は、BDL リアルタイム監視に必要な追加機能を配布するために使用されます。次の関数の使用の詳細については、補足チュートリアルを参照してください。

- MonitorInc
- MonitorIncFloat
- MonitorStart および MonitorStop
- MonitorSet
- MonitorSetFloat
- MonitorSetTimer

### BDL 監視プロジェクトを作成する

1. Silk Performer を起動し、BDL 監視プロジェクトを作成します。

- a) ワークフロー バーの **プロジェクトの概要設定** ボタンをクリックします。**ワークフロー - プロジェクトの概要設定** ダイアログ ボックスが開きます。
- b) **名前** テキスト ボックスにプロジェクトの名前を入力し、**説明** テキスト ボックスにオプションの説明を入力します。

- c) **アプリケーションの種類** リストから、**Monitoring > Bdl Monitor for Performance Explorer** を選択します。
- d) **次へ** をクリックします。

2. **スクリプトの作成** ダイアログ ボックスで、**既存のスクリプトを開く** オプション ボタンを選択します。

3. BdlMonitorSample.bdf スクリプトを選択して、**OK** をクリックします。

これにより、テンプレートとして使用できる構成済みの監視スクリプトが起動します。

4. **プロジェクト > プロジェクト属性**を選択し、プロジェクト属性を定義します。

Performance Explorer にエクスポートされた各測定値には、プロジェクト属性の固定セットが必要です。

Performance Explorer で表示されるようにするため、各測定値には少なくとも次の 3 つのプロジェクト属性が必要です。

- **Name** : Performance Explorer に表示される名前。
- **Type** : 平均値または累積値。
- **Enabled** : 予約済み。常に True に設定します。

5. サンプル監視スクリプトを確認する

プロジェクト属性の指定に従って、プロジェクトはリアルタイムに Performance Explorer で表示できる 2 つの測定値をエクスポートします。

TMon トランザクションで MonitorInc および MonitorIncFloat を検索します。これは、Performance Explorer に最後のスナップショットが提供される場所です。

```
use "bdlMonitor.bdh"

const
  nMeasure1 := 1;
  nMeasure2 := 2;

dclrand
  rRand : RndExpN (1..10000: 100.000000);
  rRandf : RndExpF (5.500000);

var
  fScale : float;

dcluser
  user
    VMon
  transactions
    TInit          : begin;
    TMon           : 1;
    TEnd           : end;

dcltrans
  transaction TInit
  begin
    fScale := AttributeGetDouble("scale");
  end TInit;

  transaction TMon
  var
    n : number;
    f : float;
  begin
    n := rRand;
    MonitorInc(1, n);
```

```
f := rRandf;
f := f * fScale;
MonitorIncFloat(2, f);
end TMon;
```

6. .conf ファイルを編集します。

サンプル .conf ファイルは、プロジェクト ツリーの **データ ファイル** ノードで利用できます。.conf ファイルをダブルクリックして開き、Type エントリの値を変更します。これは、プロジェクトが Performance Explorer 階層のどこに配置されているかを示します。たとえば、<Type>Monitoring ¥Sample Project</Type> を使用して、場所 **Monitoring¥Sample Project** で、**監視 > データソースの追加 > 定義済みのデータ ソース** にある Performance Explorer の監視プロジェクトを探します。

7. プロジェクトを単一の ZIP アーカイブにエクスポートします。

**ファイル > プロジェクトのエクスポート** を選択します。**Zip** を使用して **1 つのアーカイブ ファイルに圧縮する** チェック ボックスをオンにします。

**エクスポート場所** に C:¥Program Files¥Silk¥Silk Performer <バージョン>¥Monitors ¥BdlMonitorSample.sep を設定します。



**注:** .sep 拡張子でファイルを保存します。

8. リアルタイム BDL 監視プロジェクトを開始する

a) Performance Explorer を起動します。**監視** タブの **監視** グループで、**システム** をクリックします。**定義済みのデータ ソースから選択する** オプション ボタンを選択して、**次へ** をクリックします。

b) 新しく作成した監視プロジェクト (たとえば、**監視 > サンプルプロジェクト**) を選択し、**次へ** をクリックします。

c) 監視対象のホストの名前を入力します (localhost を選択するか、Performance Explorer によって推奨されたホストを選択し、**次へ** をクリックします)。

これにより、プロジェクト属性を変更できるダイアログが開きます (このインスタンスでは、1 つの属性だけを変更できます)。

d) 0 よりも大きい値を入力します。この値は、監視プロジェクトで使用されます。詳細については、サンプル スクリプトを参照してください。

e) **OK** をクリックします。これにより、BDL 監視プロジェクトで指定された測定値を選択できるようになります。

f) 両方の測定値を選択し、**終了** をクリックします。これにより、監視プロジェクトが起動します。これで、監視グラフを使用してリアルタイムで値を表示できるようになりました。

## Silk Performance Manager インフラストラクチャ監視を作成する

1. **ファイル > プロジェクトの新規作成** をクリックするか、プロジェクトを開いていない場合は、ワークフロー バーの **ここから開始する** をクリックします。**ワークフロー - プロジェクトの概要設定** ダイアログ ボックスが開きます。

2. **名前** および **説明** (省略可能) を入力します。

3. アプリケーションの種類リストから、**Monitoring > Performance Manager - Infrastructure Monitor** を選択し、**次へ** をクリックします。Performance Explorer が開き、**データ ソース ウィザード** が表示されます。

4. ウィザードに従ってデータ ソースを選択します。監視グラフが表示され、監視が自動的に開始されます。

5. **リアルタイム監視** タブの **エクスポート** グループで、**プロジェクトとして** をクリックします。

6. **監視再利用ウィザード** に従って監視グラフをエクスポートします。

7. **プロジェクトのアップロード** ダイアログで設定を行い、**OK** をクリックします。

## Performance Data Collection Engine (PDCE)

Performance Data Collection Engine (PDCE) は、オンライン監視に使用される Performance Explorer エンジンです。この機能には、BDL (Benchmark Description Language) を介してアクセスすることもできます。PDCE の BDL インターフェイスは、pdce.bdh インクルード ファイルで説明されています。

Performance Explorer が監視する各測定値は、BDL スクリプトを使用して Silk Performer 経由で監視することもできます。対象とする測定値を登録すると、PDCE API のポーリング機能を介して、PDCE から監視データセットを取得できます。この機能により、データおよびステータス情報、ある場合は測定値の説明、を取得できます。

## チュートリアル

このチュートリアルは、(1) 初歩的な簡易ユース ケースと (2) 詳細なユース ケースという 2 つのユース ケースから構成されます。最初のユース ケースには、1 つのトランザクションと 1 つの測定値を含む 1 つのユーザー グループから構成される監視プロジェクトがあります。

2 番目のユース ケースでは、1 つのプロジェクトに複数のユーザー グループ、トランザクション、測定値を含める方法について説明します。

### 基本ユース ケース

このユース ケースでは、SunOS で実行されるプロセス数を追跡するプロジェクトを作成します。一般的に、SunOS にはリモート実行でアクセスできます。Win2000 コマンドライン ツール rexec と対比してください。SunOS で実行されるプロセス数をカウントするには、X-Terminal、Telnet セッション、rexec、または他の望ましいツール内で `'ps -ef | egrep -c ".*"'` を実行します。たとえば、DOS プロンプトタイプで次を入力します：`c:¥>rexec yourSunHost -l username "ps -ef | egrep -c ¥".*¥"`

これは、SunHost で実行されるプロセス数を返します。ここでは、Performance Explorer リアルタイムグラフでこの値を継続的に追跡および表示することを目的としています。

### 新しいプロジェクトを作成する

このセクションで前述したように、アプリケーション タイプ Monitoring/Bdl Monitor for Performance Explorer の新しいプロジェクトを作成します。Performance Explorer に表示される監視プロジェクトの目的全体を反映する、シンプルなプロジェクトの説明 (たとえば、「SunOS システムで実行されているプロセスの数を収集するために使用する強力なプロジェクト」) を入力します。

### プロジェクト属性の計画

CountNrOfProcesses など、プロジェクト属性における測定値の名前を最初に入力します。この名前は Performance Explorer 階層でも使用されます。階層は '¥' で作成されます。

たとえば、以下の階層を作成するには、次の手順を実行します。

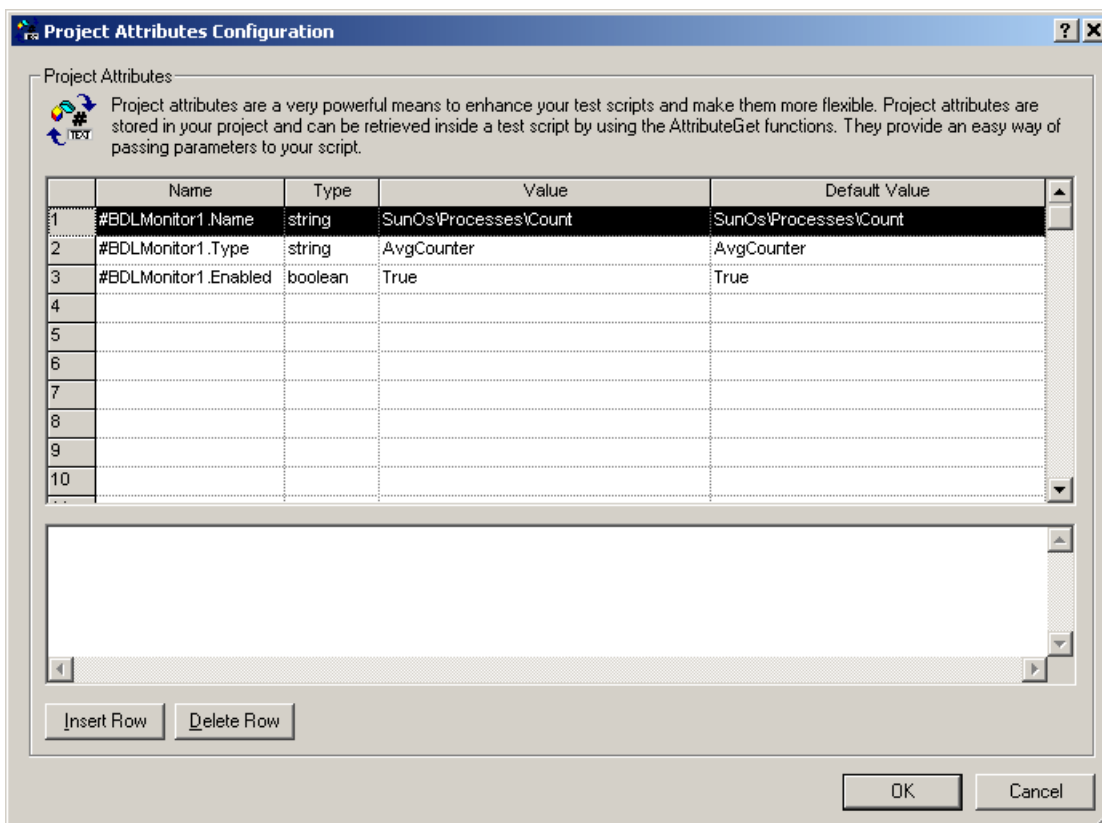
OS

```
Processes
  CountNrOfProcesses
  Another Counter
```

測定値の名前を `SunOs¥Processes¥CountNrOfProcesses` として指定します。

これは AvgCounter (平均の場合) カウンターとなり、デフォルトで収集される必要があります。

これらの値はプロジェクトの属性に転送される必要があります。プロジェクト > プロジェクト属性 の順に選択して属性エディタを開きます。



プロジェクト属性の値によって単一の測定値が識別されます。ただし、測定値を作成するために複数の属性（ダイアログ内の複数の行）が必要になる場合があります。たとえば、上記の例では #BDLMonitor1 で始まるすべてのエントリが該当します。これは、測定値に対してどのように設定を定義するかを示しています。

これで、測定値は名前、タイプ、有効な設定によって定義されました。2 番目の測定値は #BDLMonitor2 で始まります（高度なユースケースでカバーされます）。

### BDL 監視スクリプトを作成する

SunOS の「ps」など、BDL で特定のコマンドライン ツールをリモートから実行するには、3 つの関数が必要です。

- リモート マシンの「exec」サーバーに接続します。
- リモート マシンで実行されるコマンドを送信します。
- 接続を切断します。

```
// hCon will be assigned the connection handle
function Connect(/*inout*/hCon : number; sHost : string)
begin
    WebTcipConnect(hCon, sHost, 512);
end Connect;

// Send a request to the remote execution server
// remote execution protocol:
// What does a request look like in binary:
// 00username00password00shellCommandToExecute00
// What does the response look like
// 00responseData
// sample request:
// 00root00labpass00ps -ef | egrep -c ".*"00
```

```

function Request(hCon: number; sUser: string; sPwd: string;
                sCmd: string):number
var
  sSend : string;
  sBuf  : string;
  nSize : number;
  nRec  : number;
begin
  sSend := "¥h00";
  SetString(sSend, 2, sUser);
  SetString(sSend, Strlen(sUser) + 3, sPwd);
  SetString(sSend, Strlen(sUser) + Strlen(sPwd) + 4, sCmd);

  nSize := 3 + Strlen(sUser) + Strlen(sPwd)
           + Strlen(sCmd) + 1;

  WebTcipSendBin(hCon, sSend, nSize);
  WebTcipRecvExact(hCon, NULL, 1);
  WebTcipRecv(hCon, sBuf, sizeof(sBuf), nRec);
  Request := number(sBuf);
end Request;

// Closes the connection to the remote exec server
function Close(hCon : number)
begin
  WebTcipShutdown(hCon);
end Close;

```

Silk Performer MeasureInc の関数には、ラッパー関数が必要です。この関数は、すべての監視プロジェクトで使用できます。プロジェクト属性にアクセスするため、MonitorInc という名前の関数が作成されま  
す。この関数は、以前に指定した属性にアクセスします。

また、MonitorInc 関数を既存の bdh である bdlMonitor.bdh からインポートすることもできます。

```

function MonitorInc(nMon : number; nVal : number)
var
  sMeasure : string;
begin
  // This will check whether the attribute
  // "#BDLMonitor1.Enabled" was set to true
  if AttributeGetBoolean("#BDLMonitor" + string(nMon)
                        + ".Enabled") then
    // If yes then let's read the name of the measure.
    // To do this we read the the project attribute
    // "#BDLMonitor1.Name" and store it
    // to a local variable named sMeasure.
    // sMeasure will have the value:
    // "SunOs¥Processes¥CountNrOfProcesses"
    AttributeGetString("#BDLMonitor" + string(nMon)
                    + ".Name", sMeasure, sizeof(sMeasure));


    // Set a new value for
    // "SunOs¥Processes¥CountNrOfProcesses"
    MeasureInc(sMeasure, nVal, MEASURE_KIND_AVERAGE);
  end;
end MonitorInc;

```

これで、定義されているすべての関数を使用してスナップショットを取得するトランザクションをコーディングすることができます。また、このトランザクションは、プロジェクト ファイル属性にもアクセスし



ます。目標は、これらの属性を後から Performance Explorer で設定することです。ただし、ここでは、スクリプトが機能することを確認するため、4つの属性をプロジェクト属性に追加する必要があります。

 **注:** 属性の名前は、大文字小文字が区別されます。

- ホスト : sunserver などのサンプル値を割り当てます。
- コマンド : ps -ef | egrep -c ".\*" などのサンプル値を割り当てます。
- ユーザー : root などのサンプル値を割り当てます。
- パスワード : テストのためサンプル値を割り当てます。

**プロジェクト > プロジェクト属性**を選択してプロジェクト属性エディタを開き、これらの追加属性を追加します。これらは、パスワード型の属性パスワードを除いて、すべて文字列型です。テストのため属性に値を割り当てます。属性の目的を示す各属性の説明を選択します。

```
const
  nMeasure := 1;

dcluser
  user
    VMonitor
  transactions
    TSnap : 1;

dclfunc
... // your functions here

dcltrans
  transaction TSnap
  var
    hCon  : number init 0;
    sHost : string;
    sCmd  : string;
    sUser : string;
    sPwd  : string;
    nVal  : number;
  begin
    AttributeGetString("host", sHost, sizeof(sHost));
    AttributeGetString("command", sCmd, sizeof(sCmd));
    AttributeGetString("user", sUser, sizeof(sUser));
    AttributeGetString("password", sPwd, sizeof(sPwd));

    Connect(hCon, sHost);
    nVal := Request(hCon, sUser, sPwd, sCmd);
    MonitorInc(nMeasure, nVal);
    Close(hCon);
  end TSnap;
```

これで、プロジェクトが新しく作成されたスクリプトによって構成されます。プロジェクトを保存し、TryScript の実行を開始して、機能することを確認します。nVal を印刷するログ ファイルに書き込むことにより、スクリプトが機能することを確認します。スクリプトが機能した場合は、プロジェクトを保存して閉じます。

#### BDL 監視スクリプトのパッケージ化

プロジェクトを単一の ZIP ファイルとしてエクスポートするには、**ファイル > プロジェクトのエクスポート**に進みます。**Zip を使用して 1 つのアーカイブ ファイルに圧縮する** チェック ボックスをオンにして、ファイルを C:\Program Files\Silk\Silk Performer 10.0\Monitors\CountProcess.sep にエクスポートします (.sep 拡張子に注意してください)。

.sep ファイルが Performance Explorer で使用できるようにするには、.sep の .conf ファイルを変更しておく必要があります。 .conf ファイルの例を次に示します。

```
<?xml version='1.0' encoding='UTF-8'?>
<Project>
  <Type>Sample¥Remote</Type>
  <Copyright>Borland</Copyright>
  <Author>Borland</Author>
  <Version>5.1</Version>
</Project>
```

タイプの設定では、「¥」で区切られた階層を指定してください。この階層は Performance Explorer に反映されます。

#### Performance Explorer 内部での監視プロジェクトの使用

1. Performance Explorer を起動します。**リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**定義済みのデータソースから選択する** オプション ボタンを選択して、**次へ** をクリックします。
2. 新たに作成された監視プロジェクトを選択して **次へ** をクリックします。
3. 監視対象のホストを指定します。SunOs が稼働中のコンピュータを指定することをお勧めします。**次へ** をクリックしてプロジェクト属性を設定するダイアログを開きます。このダイアログに、設定した属性 (すでに指定済みのホストを除く) が表示されます。

そのため、ダイアログには次の属性が表示されます。

- user
  - password
  - command
4. 測定値 (この場合は測定値が 1 つだけです) を選択して **OK** をクリックします。測定値はデータ収集メカニズムにすぐに引き渡されます。

#### 高度なユース ケース

このユース ケースには、以前のユース ケースからのユーザー グループ、および HTTP リクエストを測定する新しいユーザー グループの 2 つのユーザー グループがあるプロジェクトが含まれます。2 つの URL とホストは、プロジェクト属性から構成可能です。

#### Silk Performer プロジェクトを作成する

Performance Explorer に表示されるシンプルなプロジェクトの説明を入力します。説明は、監視プロジェクトの目的全体を表すものにする必要があります。たとえば、SunOS システムで実行されているプロセス数、および 2 つの設定可能な HTTP リクエストの測定値を収集するために使用する強力なプロジェクトなどです。

#### プロジェクト属性の計画

このチュートリアルでは 3 つの測定値を計画します。階層と測定値の目的を示す測定値名を選出します。この例では、測定値は SunOs システム上のプロセス数をカウントし、HTTP 要求を測定する 2 つの測定値が含まれます (WebUrl を使用して実行されます)。

- SunOs¥Processes¥CountNrOfProcesses
- SunOs¥Http Server¥URI1
- SunOs¥Http Server¥URI2

階層の例は次のとおりです。

```
SunOs
  Processes
    CountNrOfProcesses
```

Http Server  
URI1  
URI2

2つのユーザーグループと2つのトランザクションを持つBDLスクリプトが記述されます。測定値がトランザクションにどのように割り当てられ、トランザクションがユーザーグループにどのように関係しているかについては、下記のテーブルを参照してください。

ユーザーグループ	トランザクション	測定値
VMonitor	TSnap	CountNrOfProcesses
VWebMon	TWebSnap	URI1
VWebMon	TWebSnap	URI2

多数のトランザクションとユーザーグループの一方または両方を含むプロジェクトでは、測定値は対応するユーザーグループと、割り当て済みの測定値を使用して指定される必要があります。これは、以前概要を説明した、1つのトランザクションと1つのユーザーグループを含む基本的なユースケースとは異なります。

**プロジェクト > プロジェクト属性** でプロジェクト属性エディタを開き、次のデータを入力します。

名前	タイプ	値
#BDLMonitor1.Name	文字列	SunOs¥Processes ¥CountNrOfProcesses
#BDLMonitor1.Type	文字列	AvgCounter
#BDLMonitor1.Enabled	論理値	True
#BDLMonitor1.Script	文字列	remote.bdf
#BDLMonitor1.Usergroup	文字列	VMonitor
#BDLMonitor1.Transaction	文字列	TSnap
#BDLMonitor2.Name	文字列	SunOs¥Http Server¥URI1
#BDLMonitor2.Type	文字列	AvgCounter
#BDLMonitor2.Enabled	論理値	true
#BDLMonitor2.Script	文字列	remote.bdf
#BDLMonitor2.Usergroup	文字列	VWebMon
#BDLMonitor2.Transaction	文字列	TWebSnap
#BDLMonitor3.Name	文字列	SunOs¥Http Server¥URI2
#BDLMonitor3.Type	文字列	AvgCounter
#BDLMonitor3.Enabled	論理値	True
#BDLMonitor3.Script	文字列	remote.bdf
#BDLMonitor3.Usergroup	文字列	VWebMon
#BDLMonitor3.Transaction	文字列	TWebSnap
-host	sunserver	sunserver
command	文字列	ps -ef   egrep -c ".*"
user	文字列	root
password	password	*****
URI1	文字列	/

名前	タイプ	値
URI2	文字列	/manual/ibm/index.html

#### BDL 監視スクリプトを作成する

SunOS の「ps」など、BDL で特定のコマンド ライン ツールをリモートから実行するには、3 つの関数が必要です。

- リモート マシンの「exec」サーバーに接続します。
- リモート マシンで実行されるコマンドを送信します。
- 接続を切断します。

```
// hCon will be assigned the connection handle
function Connect(/*inout*/hCon : number; sHost : string)
begin
    WebTcipConnect(hCon, sHost, 512);
end Connect;

// Send a request to the remote execution server
// remote execution protocol:
// What does a request look like in binary:
// 00username00password00shellCommandToExecute00
// What does the response look like
// 00responseData
// sample request:
// 00root00labpass00ps -ef | egrep -c ".*"00
function Request(hCon: number; sUser: string; sPwd: string;
                sCmd: string):number
var
    sSend : string;
    sBuf  : string;
    nSize : number;
    nRec  : number;
begin
    sSend := "¥h00";
    SetString(sSend, 2, sUser);
    SetString(sSend, Strlen(sUser) + 3, sPwd);
    SetString(sSend, Strlen(sUser) + Strlen(sPwd) + 4, sCmd);

    nSize := 3 + Strlen(sUser) + Strlen(sPwd)
            + Strlen(sCmd) + 1;

    WebTcipSendBin(hCon, sSend, nSize);
    WebTcipRecvExact(hCon, NULL, 1);
    WebTcipRecv(hCon, sBuf, sizeof(sBuf), nRec);
    Request := number(sBuf);
end Request;

// Closes the connection to the remote exec server
function Close(hCon : number)
begin
    WebTcipShutdown(hCon);
end Close;
```

Silk Performer MeasureInc の関数には、ラッパー関数が必要です。この関数は、すべての監視プロジェクトで使用できます。プロジェクト属性にアクセスするため、MonitorInc という名前の関数が作成されます。この関数は、以前に指定した属性にアクセスします。

また、MonitorInc 関数を既存の bdh である bdlMonitor.bdh からインポートすることもできます。

```
function MonitorInc(nMon : number; nVal : number)
var
  sMeasure : string;
begin
  // This will check whether the attribute
  // "#BDLMonitor1.Enabled" was set to true
  if AttributeGetBoolean("#BDLMonitor" + string(nMon)
    + ".Enabled") then
    // If yes then let's read the name of the measure.
    // To do this we read the the project attribute
    // "#BDLMonitor1.Name" and store it
    // to a local variable named sMeasure.
    // sMeasure will have the value:
    // "SunOs¥Processes¥CountNrOfProcesses"
    AttributeGetString("#BDLMonitor" + string(nMon)
      + ".Name", sMeasure, sizeof(sMeasure));

    // Set a new value for
    // "SunOs¥Processes¥CountNrOfProcesses"
    MeasureInc(sMeasure, nVal, MEASURE_KIND_AVERAGE);
  end;
end MonitorInc;
```

これで、定義されているすべての関数を使用してスナップショットを取得するトランザクションをコーディングすることができます。また、このトランザクションは、プロジェクト ファイル属性にもアクセスします。目標は、これらの属性を後から Performance Explorer で設定することです。ただし、ここでは、スクリプトが機能することを確認するため、4 つの属性をプロジェクト属性に追加する必要があります。



**注:** 属性の名前は、大文字小文字が区別されます。

- ホスト : sunserver などのサンプル値を割り当てます。
- コマンド : ps -ef | egrep -c ".\*" などのサンプル値を割り当てます。
- ユーザー : root などのサンプル値を割り当てます。
- パスワード : テストのためサンプル値を割り当てます。

**プロジェクト > プロジェクト属性**を選択してプロジェクト属性エディタを開き、これらの追加属性を追加します。これらは、パスワード型の属性パスワードを除いて、すべて文字列型です。テストのため属性に値を割り当てます。属性の目的を示す各属性の説明を選択します。

```
const
  nMeasure := 1;

dcluser
  user
    VMonitor
  transactions
    TSnap : 1;

dclfunc
.... // your functions here

dcltrans
  transaction TSnap
  var
    hCon : number init 0;
    sHost : string;
    sCmd : string;
    sUser : string;
```

```

sPwd  : string;
nVal  : number;
begin
  AttributeGetString("host", sHost, sizeof(sHost));
  AttributeGetString("command", sCmd, sizeof(sCmd));
  AttributeGetString("user", sUser, sizeof(sUser));
  AttributeGetString("password", sPwd, sizeof(sPwd));

  Connect(hCon, sHost);
  nVal := Request(hCon, sUser, sPwd, sCmd);
  MonitorInc(nMeasure, nVal);
  Close(hCon);
end TSnap;

```

これで、プロジェクトが新しく作成されたスクリプトによって構成されます。プロジェクトを保存し、TryScript の実行を開始して、機能することを確認します。nVal を印刷するログ ファイルに書き込むことにより、スクリプトが機能することを確認します。スクリプトが機能した場合は、プロジェクトを保存して閉じます。

### プロジェクトのパッケージ化

プロジェクトを単一の ZIP ファイルにエクスポートします。 **ファイル > プロジェクトのエクスポート** に進みます。 **Zip を使用して 1 つのアーカイブ ファイルに圧縮する** チェック ボックスをオンにして、ファイルを C:\Program Files\Silk\Silk Performer\Monitors\Advanced.sep にエクスポートします (SEP ファイル拡張子に注意してください)。

SEP ファイルが Performance Explorer で使用できるようにするには、SEP の CONF ファイルを変更しておく必要があります。CONF ファイルの例を次に示します。

```

<?xml version='1.0' encoding='UTF-8'?>
<Project>
  <Type>Sample¥Advanced¥HttpHit</Type>
  <Copyright>Borland</Copyright>
  <Author>Borland</Author>
  <Version>5.1</Version>
</Project>

```

タイプの設定では、バックスラッシュ (「¥」) で区切られた階層を指定してください。この階層は Performance Explorer に反映されます。

### Performance Explorer での監視プロジェクトの使用

1. Performance Explorer を起動します。 **リアルタイム監視** タブの **監視** グループで、 **システム** をクリックします。 **定義済みのデータ ソースから選択する** を選択します。
2. 新たに作成された監視プロジェクトを選択して **次へ** をクリックします。
3. 監視対象のサーバー名を入力します。たとえば、sunserver と入力して **次へ** をクリックします。ダイアログ ボックスが表示されます。次の属性を定義します。
  - **command** : このチュートリアルですでに設定した基本的なテスト ケースと同様に、プロセスの数を数えるコマンド。
  - **user** : リモート コンピュータにアクセスするための有効なユーザー アカウント。
  - **password** : パスワード。
  - **URI1** : レスポンス時間が測定される URI。
  - **URI2** : レスポンス時間が測定される 2 番目の URI。
4. 測定値 (この場合 1 つだけ) を選択して **OK** をクリックします。測定値はデータ収集メカニズムにすぐに引き渡されます。選択項目はプロジェクト属性で定義された階層に対応します。

## ベストプラクティス

このセクションでは、監視スクリプトの書き込みに関連するリアルタイム監視プロジェクトでの作業に関する提案について、概要を示します。

### ループ

Performance Explorer は、標準インターバルでスナップショットを実行するトランザクションを実行します。たとえば、スナップショット トランザクションは 10 秒ごとに実行される場合があります。このようなケースで、Performance Explorer は、トランザクション全体がこの 1 つのインターバルで実行されると見なします。そうでない場合、Performance Explorer はインターバルのスナップショットを収集できないことを意味する警告を戻します。

次のような無限ループを使用しないでください：

```
// Forbidden
transaction TSnap
begin
  while true do
    Snap();
  end;
end TSnap;
```

### 思考時間と待機

待機文に注意してください。思考時間は無視されます。監視トランザクションは所定のインターバル内に実行する必要があります。待機文により実行がブロックされ、トランザクションが時間内に戻らないことがあります。

#### 例

```
// Forbidden
transaction TSnap
begin
  wait 500000.0;
end TSnap;
```

### 初期化

ランタイムを停止させる可能性のある TInit トランザクションに、関数呼び出しを配置しないでください。たとえば、ログオンが失敗すると、OraLogon によって ProcessExit が生成されることがあります。Performance Explorer はこのメッセージを捕捉せず、プロジェクトが引き続き実行されていると見なします。当然ながら、Performance Explorer はデータを収集できないことをレポートしますが、これが発生した理由をレポートすることはできません。

### その他のベストプラクティス

#### 名前の長さ

属性値の長さが 79 文字を超えてはいけません。

#### プロジェクトに対するスクリプトの試行を実行する

スクリプトの試行を実行して、SEP ファイルを作成する前に、プロジェクトが動作していることを確認します。

## 特別なプロジェクト属性

監視プロジェクト属性 host は、次の関数呼び出し : AttributeGetString("host", sHost, sizeof(sHost)); を使用して BDL スクリプト内にアクセスできます。

host の代わりに、#MonitorHost という属性が照会されることもあります。これらの 2 つの属性は均等に扱われます。

Performance Explorer で、この属性の値は次のダイアログを使用して設定されます。

## データ ソースを監視する

Performance Explorer を使用してシステムの監視を開始する前に、システムや測定値を指定する必要があります。この方法は 2 通りあります。

- 定義済みのデータ ソースを選択する。

Performance Explorer には、最も普及している Web サーバー、アプリケーション サーバー、データベース サーバー、オペレーティング システム用の定義済みデータ ソースが用意されています。なお、使用するデータ ソースがこれらに限られるわけではありません。これらのデータ ソースが付属しているのは、監視対象となる測定値を手早く選択できるようにするためです。

- データ ソース スキャナを使用してデータ ソースを検出する。

データ ソース スキャナでは、お使いのシステムを調べて、さまざまなプラットフォームでよく見かけるさまざまなデータ ソースを探し、使用可能なデータ ソースの一覧を返します。

システムを指定するには、**リアルタイム監視** タブの **監視** グループで **システム** をクリックし、ウィザードの指示に従います。

## 定義済みのデータ ソース

Performance Explorer では、Performance Explorer で監視可能なサーバーの種類に従って、設定をアシストするウィザードを利用できます。

### Apache サーバーの監視を有効にする

Apache Web Server では、組み込みのステータス レポート機能の使用を選択できます。この機能を利用するには、mod\_status モジュールを組み込む必要があります (これはデフォルトで行われます)。


1. コンパイル オプションの **ExtendedStatus On** オプション ボタンをクリックすると、詳細情報が表示されます。
2. 構成ファイル access.conf にコードを追加し、foo.com ドメインからのブラウザについてのみステータス レポートを有効にします。

Apache の構成は、一連のファイルで管理されます。これらのファイルは、一般に、/usr/local/apache/etc、または /usr/local/apache/conf (Solaris)、/etc/httpd/conf (Linux) に置かれています。

3. 次のコードを Apache のメイン構成ファイル httpd.conf に追加します。

```
<Location /server-status> SetHandler server-status
order deny, allow deny from all allow from .foo.com
</Location>
```

既存のこのコードがコメントアウトされている場合は、コメントを解除してください。

 **注:** 監視が適切に機能するには、すべての allow 文が正しく設定されていることが必要です。

4. サーバーを再起動します。
5. サーバーのパフォーマンス モジュールを確認するため、ブラウザに http://<ホスト名>/server-status?auto と入力します。次のメッセージのようなレスポンスが返れば、監視機能は有効です。  
Total Accesses: 210 Total kBytes: 94 CPUload: .000278279 Uptime: 366539 ReqPerSec: .000572927 BytesPerSec: .262608 BytesPerReq: 458.362 BusyServers: 1 IdleServers: 6  
Scoreboard: \_\_\_\_\_W.....





**注:** デフォルトの Apache 監視は、Telnet が有効になっている場合にのみ機能します。Telnet が無効の状態では Apache サーバーを監視するには、デフォルトの Rexec コマンドの代わりに、次の例に示すような WGet コマンドを使用する必要があります。

```
wget -q -O - http://<hostname>/server-status | grep  
"requests/sec" | cut -c5- | cut -d¥ -f1
```

## Silverstream サーバーへの SNMP サポートを有効にする

1. SilverStream デザイナを実行します。
2. **SilverMaster3 database > Objects** を選択します。
3. AgSNMPGetStats をデザイン モードで開きます。
4. トリガ アイコンをダブルクリックします。 **プロパティ** ページが開きます。
5. **Enabled** オプションを **true** に設定します。
6. 変更を保存して、ファイルを閉じます。

## Silk Central パフォーマンスを監視する

Performance Explorer を使用して、Silk Central アプリケーション サーバーおよびフロントエンド サーバーのパフォーマンスと信頼性指標を監視します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。 **データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。 **システムの選択** ページが開きます。
4. 監視するバージョンで **Silk** フォルダおよび **Silk Central** フォルダを展開し、次にアプリケーション サーバーまたはフロントエンド サーバーを選択します。
5. **次へ** をクリックします。  
**接続パラメータ** ページが開きます。
6. 以下の設定を指定します。

- **ホスト名** テキスト ボックスに、Silk Central が実行されているホスト名または IP アドレスを指定します。
- 省略可能 : **エイリアス** テキスト ボックスに、監視されるサーバーのエイリアスを指定します。
- Silk Central がデフォルト ポートでリスンしない場合は、**ポート** テキスト ボックスで JMX ポート番号を指定します。
- JMX 認証が有効の場合、**ユーザー名** と **パスワード** を入力します。デフォルト JMX 認証が無効の場合は、**ユーザー名** および **パスワード** は空である必要があります。



**注:** ローカル ファイアウォールがオフになっているか、またはポート 19140 および 19142 がファイアウォール経由で通信できることを確認します。

7. **次へ** をクリックします。 **JMX Data Source Browser** が開きます。
8. 初期監視ビューに表示する測定値を選択し、**Add** をクリックします。測定値の追加を終了したら、**閉じる** をクリックします。



**ヒント:** **Simple Query** タブを選択し、borland.com による測定値にフィルタを適用して、Silk Central 測定値にすばやくアクセスします。

**表示する測定値の選択** ページが開きます。

9. 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## IBM DB2 のパフォーマンス監視

Performance Explorer を使用すると、データベース プラットフォームを問わず、IBM DB2 データベース システムからパフォーマンス指標を取得することができます。たとえば、データベースは SunOS、Linux、Windows システムのいずれでも稼働できます。

Performance Explorer では、プラットフォームに依存しない形で DB2 データベース システムを監視できます。それには、Silk Performer 内から、データベースのパフォーマンスと利用状況の測定値を取得するための DB2 固有のコマンドを実行し、Performance Explorer プロジェクトを操作します。これにより、Performance Explorer ではデータベース システムのパフォーマンスをリアルタイムに追跡することもできます。

Silk Performer プロジェクトは、DB2Monitor.sep という名前の Essential パッケージに格納されています。このパッケージは、C:\Program Files\Silk\Silk Performer 21.0\Monitors にあります。

### DB2 監視の前提条件

DB2 を監視するには、お使いの環境が以下の要件を満たす必要があります。

- Performance Explorer を実行する予定のコンピュータに IBM DB2 クライアント ソフトウェア (DB2 Connect) がインストールされていること。
- クライアント インストールの一環として、DB2 コマンドライン プロセッサ db2cmd.exe がインストールされていること。これはデフォルトでの前提条件となります。
- 該当する DB2 データベース ユーザーが次の権限のいずれかを所有していること。
  - sysadm
  - sysctrl
  - sysmaint
- DB2 スナップショット モニタが正しくセットアップされていること。

### DB2 スナップショット監視をテストする

#### 1. DB2 データベースに接続します。

DB2 コマンド:

```
=> connect to sample user <dbusername> using <password>
```

#### 2. DB2 インスタンスにアタッチします。

DB2 コマンド:

```
=> attach to <db2instancename> user <dbusername> using <dbusername>
```

#### 3. 監視スイッチがオンになっているか確認します。

DB2 コマンド:

```
=> get monitor switches
```

### 監視記録スイッチ

ノード 0 のスイッチ リスト

```
Buffer Pool Activity Information (BUFFERPOOL) = ON 02-06-2002
18:27:48.722132
Lock Information (LOCK) = ON 02-06-2002
18:28:00.095212
Sorting Information (SORT) = ON 02-06-2002
18:28:12.263183
SQL Statement Information (STATEMENT) = ON 02-06-2002
18:28:24.323446
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

4. スイッチをオンにします。

DB2 コマンド:

=> update monitor switches using bufferpool on

=> update monitor switches using lock on

=> update monitor switches using sort on

5. DB2 UDB システム監視スナップショットからデータを取得します。

DB2 コマンド:

=> get snapshot for all databases

**データベース スナップショット:**

```
Database name = SAMPLE
Database path = /home/db2inst1/db2inst1/
NODE0000/SQL00001/
Input database alias =
Database status = Active
Catalog node number = 0
Catalog network node name =
Operating system running at database server= LINUX
Location of the database = Remote
First database connect timestamp = 02-06-2002 18:01:31.198883
Last reset timestamp =
Last backup timestamp =
Snapshot timestamp = 02-06-2002 18:37:39.254985
High water mark for connections = 1
Application connects = 1
Secondary connects total = 0
Applications connected currently = 1
Appls. executing in db manager currently = 0
Agents associated with applications = 1
Maximum agents associated with applications= 1
Maximum coordinating agents = 1
Locks held currently = 0
Lock waits = 0
Time database waited on locks (ms) = 0
Lock list memory in use (Bytes) = 792
Deadlocks detected = 0
Lock escalations = 0
Exclusive lock escalations = 0
Agents currently waiting on locks = 0
Lock Timeouts = 0
Total sort heap allocated = 0
Total sorts = 0
Total sort time (ms) = 0
Sort overflows = 0
Active sorts = 0
Buffer pool data logical reads = 0
Buffer pool data physical reads = 0
Asynchronous pool data page reads = 0
Buffer pool data writes = 0
Asynchronous pool data page writes = 0
Buffer pool index logical reads = 0
Buffer pool index physical reads = 0
Asynchronous pool index page reads = 0
Buffer pool index writes = 0
Asynchronous pool index page writes = 0
```

Total buffer pool read time (ms) = 0  
Total buffer pool write time (ms) = 0  
Total elapsed asynchronous read time = 0  
Total elapsed asynchronous write time = 0  
Asynchronous read requests = 0  
LSN Gap cleaner triggers = 0  
Dirty page steal cleaner triggers = 0  
Dirty page threshold cleaner triggers = 0  
Time waited for prefetch (ms) = 0  
Direct reads = 0  
Direct writes = 0  
Direct read requests = 0  
Direct write requests = 0  
Direct reads elapsed time (ms) = 0  
Direct write elapsed time (ms) = 0  
Database files closed = 0  
Data pages copied to extended storage = 0  
Index pages copied to extended storage = 0  
Data pages copied from extended storage = 0  
Index pages copied from extended storage = 0  
Host execution elapsed time = 0.000000  
Commit statements attempted = 1  
Rollback statements attempted = 0  
Dynamic statements attempted = 0  
Static statements attempted = 1  
Failed statement operations = 0  
Select SQL statements executed = 0  
Update/Insert/Delete statements executed = 0  
DDL statements executed = 0  
Internal automatic rebinds = 0  
Internal rows deleted = 0  
Internal rows inserted = 0  
Internal rows updated = 0  
Internal commits = 1  
Internal rollbacks = 0  
Internal rollbacks due to deadlock = 0  
Rows deleted = 0  
Rows inserted = 0  
Rows updated = 0  
Rows selected = 0  
Rows read = 9  
Binds/precompiles attempted = 0  
Log space available to the database (Bytes)= 20400000  
Log space used by the database (Bytes) = 0  
Maximum secondary log space used (Bytes) = 0  
Maximum total log space used (Bytes) = 0  
Secondary logs allocated currently = 0  
Log pages read = 0  
Log pages written = 0  
Appl id holding the oldest transaction = 0  
Package cache lookups = 0  
Package cache inserts = 0  
Package cache overflows = 0  
Package cache high water mark (Bytes) = 51824  
Application section lookups = 0  
Application section inserts = 0  
Catalog cache lookups = 0  
Catalog cache inserts = 0

```
Catalog cache overflows = 0
Catalog cache heap full = 0
Number of hash joins = 0
Number of hash loops = 0
Number of hash join overflows = 0
Number of small hash join overflows = 0
```

IBM DB2 パフォーマンス指標を取得する

IBM DB2 DBMS システムを監視するためにデータソースへの接続を確立し、監視する測定値を含む初期ビューを作成します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データソースウィザード** が開きます。
2. **定義済みのデータソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。**システムの選択** ページが開きます。
4. **Database System** フォルダを展開して、**IBM Universal Database DB2 (snapshot dll)** を選択します。
5. **次へ** をクリックします。  
**接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスに、DB2 データベースが実行されているホストを指定します。
7. **次へ** をクリックします。**属性設定** ページが開きます。
8. 以下の監視固有の属性を定義します。
  - **Alias** : データベースエイリアスは、監視プロジェクトにより "connect to 'alias' user 'user' using 'password'" という DB2 固有のコマンドの中で使用されます。
  - **Instance** : インスタンス名は、監視プロジェクトにより "attach to 'Instance' user 'user' using 'password'" という DB2 固有のコマンドを実行するために使用されます。
  - **User** : 次の DB2 権限を持つユーザーを指定します。
    - sysadm
    - sysctrl
    - sysmaint
  - **Password** : ユーザーのパスワード。
9. **OK** をクリックします。**表示する測定値の選択** ページが開きます。
- 10 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## IBM WebSphere Application Server の監視

Performance Explorer では、以下の IBM WebSphere Application Server を監視できます。

- IBM WebSphere Application Server 6.1 (JMX)
- IBM WebSphere Application Server 7.0 (JMX)
- IBM WebSphere Application Server 8.0 (JMX)
- IBM WebSphere Application Server 8.5 (JMX)

*WebSphere Application Server 6.1* に関する代表的なパフォーマンス測定値

以下の表で、WebSphere Application Server の代表的なパフォーマンス測定値および定義を示します。

パフォーマンス測定値	説明
BeanModule	作成された Bean の数などの配置済みの EJB に関する固有のデータ。
ConnectionPoolModule	現在プールにある JDBC 接続の数など、JDBC 接続に関するパフォーマンス カウンタを含む。
JvmRuntimeModule	Java 仮想マシン固有のパフォーマンス カウンタ。
ServletSessionModule	アクティブなセッションの数やセッションの平均継続時間などのサーブレットセッションに関するパフォーマンス カウンタを含む。
TransactionModule	実行中のトランザクションの数や 1 件のトランザクションの実行に必要な平均時間に関する情報を提供するパフォーマンス カウンタを含む。
WebApplicationModule	サーブレットがリクエストを処理するのにかかる平均時間などの配置済みのサーブレットや JSP に関する情報を提供。

### IBM WebSphere Application Server を監視する

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。**システムの選択** ページが開きます。
4. **Application Server** フォルダ、**IBM WebSphere Application Server** フォルダの順に展開します。
5. **IBM WebSphere <バージョン> (JMX MBean Server)** をクリックし、**次へ** をクリックします。**接続パラメータ** ページが開きます。
6. **ホスト名** フィールドに、WebSphere が実行されているホストを指定します。
7. **ユーザー名** および **パスワード** フィールドで、上で指定したホストに対して管理者権限を持つユーザーを指定します。
8. 省略可能：**エイリアス** フィールドに、監視されるサーバーのエイリアスを指定します。
9. **サーバー設定** をクリックします。**JMX Connection Configuration** ウィンドウが開きます。
10. **Java ホーム ディレクトリ** および **アプリケーション サーバーのインストール ディレクトリ** を指定します。UNC パスを使用してください (¥¥<サーバー名>¥c\$¥IBM¥WebSphere¥AppServer¥java など)。




**注:** UNC パスを指定できない場合は、Application Server のインストール ディレクトリをローカル コンピュータにコピーして、ローカル パスを指定します。

11. WebSphere のセキュリティ設定に基づき、追加の JVM パラメータを指定します。

JVM ベンダー	IBM	SUN
<b>セキュリティ設定</b>		
無効な管理セキュリティ	追加の JVM パラメータなし	追加の JVM パラメータなし
有効な管理セキュリティ	追加の JVM パラメータ：-	追加の JVM パラメータ：-
インバウンド CSIv2 トランスポート層 = TCP/IP	Dcom.ibm.CORBA.ConfigURL (すでに定義済み)	Dcom.ibm.CORBA.ConfigURL (定義済み)
有効な管理セキュリティ	追加の JVM パラメータ：-	機能しない
インバウンド CSIv2 トランスポート層 = SSL サポート	Dcom.ibm.CORBA.ConfigURL (定義済み)- Dcom.ibm.SSL.ConfigURL	

JVM ベンダー	IBM	SUN
<b>セキュリティ設定</b>		
有効な管理セキュリティ	追加の JVM パラメータ: -	機能しない
インバウンド CSIv2 トランスポート層 = SSL 必須	Dcom.ibm.CORBA.ConfigURL (すでに定義済み)  -Dcom.ibm.SSL.ConfigURL	

 **注:** WebSphere サーバーで管理セキュリティが有効になっていて、インバウンド CSIv2 トランスポート層が SSL サポート (または SSL 必須) の場合、Application Server のインストールディレクトリをローカル コンピュータにコピーします。上記の表に従い **Additional JVM parameters** (追加の JVM パラメータ) を指定します。 **Additional JVM parameters** フィールドは、定義済みパラメータ -Dcom.ibm.CORBA.ConfigURL のままになります。

-Dcom.ibm.CORBA.ConfigURL パラメータの値は、sas.client.props ファイルへの URL パスにする必要があります。デフォルトの sas.client.props ファイルは、%INSTALL\_DIR%/Include/jmx-config/WebSphere%MAJOR\_VERSION% フォルダにあります。例: WebSphere 7 の場合、URL パスは C:¥Program%20Files%20(x86)¥Silk¥Silk%20Performer%2010.0¥Include¥jmx-config¥WebSphere7 になります。サーバーに特定の設定がある場合、WebSphere プロファイルの properties フォルダから sas.client.props ファイルを使用できます。例: file:/C:¥Program%20Files%20(x86)¥IBM¥WebSphere¥AppServer1¥profiles¥AppSrv01¥properties¥sas.client.props

sas.client.props ファイルにある次の 2 つの変数とその値  
com.ibm.CSI.performTransportAssocSSLTLSRequired=false および  
com.ibm.CSI.performTransportAssocSSLTLSSupported=true に注意してください。

-Dcom.ibm.SSL.ConfigURL パラメータの値は、WebSphere プロファイルの properties フォルダにある ssl.client.props ファイルへの URL パスにする必要があります。例: file:/C:¥Program%20Files%20(x86)¥IBM¥WebSphere¥AppServer1¥profiles¥AppSrv01¥properties¥ssl.client.props。

ssl.client.props ファイルの変数 user.root に注意してください。これには key- および trust- ファイルが格納された etc ディレクトリを含む profile フォルダへの有効なパスが含まれている必要があります。sas.client.props ファイル および ssl.client.props ファイルへの URL パスに含まれるブランクは %20 で置き換える必要があります。

**12OK** をクリックします。

**13次へ** をクリックします。 **JMX Data Source Browser** を開きます。

**14初期監視ビュー**に表示する測定値の MBean を選択します。測定の種類 (**Average**、**Sum**、**Incremental**) を選択して **追加** をクリックします。

**15閉じる** をクリックします。 **表示する測定値の選択** ページが開きます。

**16初期監視ビュー**に表示する測定値のチェック ボックスをオンにします。

**17完了** をクリックします。

## Internet Information Server (IIS) のパフォーマンス監視

*Internet Information Services (IIS)* を監視する

- リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。 **データ ソース ウィザード** が開きます。
- 定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
- 次へ** をクリックします。 **システムの選択** ページが開きます。
- Web サーバー フォルダ、IIS <バージョン> フォルダの順に展開します。
- System Statistics (PERFMON) ノードをクリックして、**次へ** をクリックします。 **接続パラメータ** ページが開きます。

6. **ホスト名** テキストボックスに、IIS が実行されているホストを指定します。
7. **ユーザー名** および **パスワード** テキストボックスで、上で指定したホスト上で管理者権限を持つユーザーを指定します。
8. 省略可能：**エイリアス** テキストボックスに、監視されるサーバーのエイリアスを指定します。
9. **次へ** をクリックします。 **表示する測定値の選択** ページが開きます。
- 10 初期監視ビューに表示する測定値のチェックボックスをオンにします。
- 11 **完了** をクリックします。

利用可能な IIS パフォーマンス測定値

### IIS 6.0 に関するパフォーマンス測定値

測定値は、perfmon オブジェクトに基づいて複数のグループに分類できます。

#### Web サービスの測定値

- **Web Service(\_Total)¥Bytes Received/sec**  
サービスによって転送された合計バイト数の速度(受信)
- **Web Service(\_Total)¥Bytes Sent/sec**  
サービスによって転送された合計バイト数の速度(送信)
- **Web Service(\_Total)¥Bytes Total/sec**  
サービスによって転送された合計バイト数の速度(送信バイト数と受信バイト数の合計)
- **Web Service(\_Total)¥Current Connections**  
サービスへの現在の接続数
- **Web Service(\_Total)¥Get Requests/sec**  
WWW サービスが受信した HTTP GET リクエストの合計数
- **Web Service(\_Total)¥Post Requests/sec**  
POST メソッドを使用した HTTP リクエストの数

#### Web サービス キャッシュの測定値

- **Web Service Cache(\_Total)¥Current Files Cached**  
コンテンツがユーザー モード キャッシュに格納されているファイルの現在の数
- **Web Service Cache(\_Total)¥Current Metadata Cached**  
現在ユーザー モード キャッシュに格納されているメタデータ情報ブロックの現在の数
- **Web Service Cache(\_Total)¥Current URIs Cached**  
現在ユーザー モード キャッシュに格納されている URI 情報ブロック
- **Web Service Cache(\_Total)¥File Cache Hits %**  
キャッシュ リクエストの合計数に対するユーザー モード ファイル キャッシュ ヒットの比率 (サービス起動時から)。注: カーネル URI キャッシュ ヒット率が高い場合、この値は低くなる可能性があります。
- **Web Service Cache(\_Total)¥Metadata Cache Hits**  
キャッシュ リクエストの合計数に対するユーザー モード メタデータ キャッシュ ヒットの比率 (サービス起動時から)
- **Web Service Cache(\_Total)¥URI Cache Hits %**  
キャッシュ リクエストの合計数に対するユーザー モード URI キャッシュ ヒットの比率 (サービス起動時から)
- **Web Service Cache(\_Total)¥Kernel:URI Cache Hits %**  
静的非認証コンテンツとキャッシュに格納可能としてマークされた動的コンテンツに適用



## IIS 7.0、7.5 に関するパフォーマンス測定値

すべての IIS 6.0 測定値は IIS 7.0/7.5 でも使用できます。IIS7 では、実行時のワーカー プロセスのレベルを深く理解するための追加のパフォーマンス カウンタが用意されています。

**Web サービスの測定値** - IIS 6.0 と同じ

**Web サービス キャッシュの測定値** - IIS 6.0 と同じ

**W3SVC\_W3WP キャッシュの測定値**

- **W3SVC\_W3WP(\_Total)¥Requests / Sec**  
ワーカー プロセスによって処理されている HTTP リクエスト数/秒
- **W3SVC\_W3WP(\_Total)¥Active Requests**  
ワーカー プロセスによって処理されているリクエストの現在の数
- **W3SVC\_W3WP(\_Total)¥Active Threads Count**  
ワーカー プロセスでリクエストをアクティブに処理しているスレッドの数
- **W3SVC\_W3WP(\_Total)¥Current File Cache Memory Usage**  
ユーザー モード ファイル キャッシュが使用している現在のバイト数
- **W3SVC\_W3WP(\_Total)¥Current Files Cached**  
コンテンツがユーザー モード キャッシュに格納されているファイルの現在の数
- **W3SVC\_W3WP(\_Total)¥Current URIs Cached**  
現在ユーザー モード キャッシュに格納されている URI 情報ブロック
- **W3SVC\_W3WP(\_Total)¥Current Metadata Cached**  
現在ユーザー モード キャッシュに格納されているメタデータ情報ブロックの現在の数
- **W3SVC\_W3WP(\_Total)¥Metadata Cache Hits**  
ユーザー モード メタデータ キャッシュで成功した検索の合計数 (サービス起動時から)
- **W3SVC\_W3WP(\_Total)¥Metadata Cache Misses**  
ユーザー モード メタデータ キャッシュで失敗した検索の合計数 (サービス起動時から)
- **W3SVC\_W3WP(\_Total)¥Metadata Cache Flushes**  
ユーザー モード メタデータ キャッシュのフラッシュの合計数 (サービス起動時から)
- **W3SVC\_W3WP(\_Total)¥File Cache Hits / sec**  
最後のサンプル期間中にファイル キャッシュで成功した検索の速度
- **W3SVC\_W3WP(\_Total)¥File Cache Misses / sec**  
最後のサンプル期間中にファイル キャッシュで失敗した検索の速度
- **W3SVC\_W3WP(\_Total)¥Metadata Cache Hits / sec**  
最後のサンプル期間中にメタデータ キャッシュで成功した検索の速度
- **W3SVC\_W3WP(\_Total)¥Metadata Cache Misses / sec**  
最後のサンプル期間中にメタデータ キャッシュで失敗した検索の速度
- **W3SVC\_W3WP(\_Total)¥Uri Cache Hits / sec**  
最後のサンプル期間中に URI キャッシュで成功した検索の速度
- **W3SVC\_W3WP(\_Total)¥Uri Cache Misses / sec**  
最後のサンプル期間中に URI キャッシュで失敗した検索の速度

IIS 監視のトラブルシューティング

### x64/x86 カウンタの問題

サポートについては、<http://support.microsoft.com/kb/891238> を参照してください。

## W3SVC\_W3WP での問題

カウンタのこのグループでは、ランタイム時のワーカー処理が監視されます。ワーカーが実行されていない場合、このグループのカウンタは NULL 値を返します。

## Microsoft SQL Server のパフォーマンス監視

Microsoft SQL Server を監視する

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。**システムの選択** ページが開きます。
4. Database System フォルダ、SQL Server <version> フォルダの順に展開します。
5. System Statistics (PERFMON) ノードをクリックして、**次へ** をクリックします。**接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスに、SQL Server が実行されているホストを指定します。
7. **ユーザー名** および **パスワード** テキスト ボックスで、上で指定したホスト上で管理者権限を持つユーザーを指定します。
8. 省略可能：**エイリアス** テキスト ボックスに、監視されるサーバーのエイリアスを指定します。
9. **次へ** をクリックします。**表示する測定値の選択** ページが開きます。
- 10 初期監視ビューに表示する測定値のチェック ボックスをオンにします。
- 11 **完了** をクリックします。

利用可能な Microsoft SQL Server パフォーマンス測定値

### Microsoft SQL Server 2005 に関するパフォーマンス測定値

次の指標は、Microsoft SQL Server 2005 システムで監視できます。

- **Process(sqlservr)¥% Processor Time**  
% Processor Time は、このプロセスのすべてのスレッドが命令を実行するためにプロセッサを使用した経過時間の割合です。
- **SQLServer:Access Methods¥Full Scans/sec**  
制限されていないフル スキャンの数です。ベース テーブルまたはフル インデックス スキャンのどちらかです。
- **SQLServer:Access Methods¥Index Searches/sec**  
インデックス検索の数です。インデックス検索は、範囲スキャンの開始、1 つのインデックス レコードのフェッチ、インデックス内での位置変更を実行するために使用されます。
- **SQLServer:Access Methods¥Table Lock Escalations/sec**  
テーブルのロックがエスカレートされた回数です。
- **SQLServer:Buffer Manager¥Buffer cache hit ratio**  
ディスクから読み取ることなく、バッファ キャッシュで見つかったページの割合です。
- **SQLServer:Buffer Manager¥Checkpoint pages/sec**  
すべてのダーティー ページをフラッシュする必要があるチェックポイントなどの操作によってフラッシュされたページの数です。
- **SQLServer:Buffer Manager¥Lazy writes/sec**  
バッファ マネージャのレイジー ライターによって書き込まれたバッファの数です。
- **SQLServer:Buffer Manager¥Page lookups/sec**  
バッファ プールでページを検索するように要求するリクエストの数です。

- **SQLServer:Buffer Manager¥Page reads/sec**  
発行された物理データベース ページ読み取りの数です。
- **SQLServer:Buffer Manager¥Page writes/sec**  
発行された物理データベース ページ書き込みの数です。
- **SQLServer:Buffer Manager¥Readahead pages/sec**  
使用されると予想して、読み取られたページの数です。
- **SQLServer:Cursor Manager by Type¥Active cursors**  
アクティブなカーソルの数です。
- **SQLServer:Cursor Manager by Type¥Cursor memory usage**  
カーソルが消費したメモリの量 (キロバイト (KB) 単位) です。
- **SQL Server:Databases¥ Active Transactions**  
データベースのアクティブな更新トランザクションの数です。
- **SQL Server:Databases¥ Shrink Data Movement Bytes/sec**  
自動圧縮、DBCC SHRINKDATABASE、または SHRINKFILE によって移動されているデータの 1 秒当たりの量です。
- **SQL Server:Databases¥ Transactions/sec**  
データベースに対して開始されたトランザクションの数です。
- **SQLServer:General Statistics¥User Connections**  
このシステムに接続しているユーザーの数です。
- **SQLServer:Locks(\_Total)¥Average Wait Time (ms)**  
待機することになった各ロック リクエストの平均待ち時間 (ミリ秒) です。
- **SQLServer:Locks(\_Total)¥Lock Waits/sec**  
すぐに満たすことができず、呼び出し側はロックが許可されるまで待機する必要があったロック リクエストの数です。
- **SQLServer:Locks(\_Total)¥Number of Deadlocks/sec**  
デッドロックとなったロック リクエストの数です。
- **SQLServer:Memory Manager¥Target Server Memory (KB)**  
サーバーが使用できる動的メモリの総量です。
- **SQLServer:Memory Manager¥Total Server Memory (KB)**  
サーバーが現在使用している動的メモリの総量です。
- **SQLServer:SQL Statistics¥Batch Requests/sec**  
サーバーが受信した SQL バッチ リクエストの数です。
- **SQLServer:SQL Statistics¥SQL Compilations/sec**  
1 秒当たりの SQL コンパイル数です。
- **SQLServer:SQL Statistics¥SQL Re-Compilations/sec**  
1 秒当たりの SQL 再コンパイル数です。
- **SQLServer:Transactions¥Free Space in tempdb (KB)**  
tempdb の使用可能な領域 (キロバイト単位) です。 スナップショット分離レベルのバージョンストアと、このデータベース エンジン インスタンスで作成されたすべての新しい一時オブジェクトの両方を保持できる十分な空き領域が必要です。
- **SQLServer:Transactions¥Transactions**  
すべての種類の現在アクティブなトランザクションの数です。

## Microsoft SQL Server 2008、2008 R2 に関するパフォーマンス測定値

すべての Microsoft SQL Server 2005 カウンタは Microsoft SQL Server 2008 と 2008 R2 システムでも使用できます。それ以外に次の指標も使用できます。

- **SQL Server: Databases¥Tracked transactions/sec**  
データベースのコミットテーブルに記録されたコミットされたトランザクションの数です。
- **SQL Server: Databases¥Write Transactions/sec**  
最後の 1 秒でデータベースに書き込まれたトランザクションの数です。
- **SQL Server: General Statistics¥Connection reset/sec**  
1 秒当たりの接続リセットの合計数です。
- **SQL Server: General Statistics¥Tempdb rowset id**  
生成された重複する tempdb 行セット ID の数です。
- **SQL Server: SQL Statistics¥Misguided plan executions/sec**  
プラン生成時にプラン ガイドを受け付けることができなかったプラン実行の 1 秒当たりの数です。
- **SQL Server: SQL Statistics¥Guided plan executions/sec**  
プラン ガイドを使用してクエリ プランが生成されたプラン実行の 1 秒当たりの数です。

## Oracle Forms のパフォーマンス監視

Performance Explorer を使用すると、Oracle Forms Dynamic Monitoring System (DMS) からパフォーマンス指標を取得できます。Performance Explorer では、DMS モジュールがインストールされると、Oracle Forms HTTP サーバーを通じて提供される DMS パフォーマンス Web ページを監視できます。

Web ページ リクエストとパフォーマンス指標解析は、Silk Performer を通じて Performance Explorer プロジェクトを操作することで実現されます。これにより、アプリケーション サーバーのパフォーマンスをリアルタイムに追跡できます。

Silk Performer プロジェクトは、OraFormsDMS.sep という名前の SilkEssential パッケージにあり、これは C:\Program Files\Silk\Silk Performer 21.0\Monitors にあります。

Oracle Forms アプリケーションには、次の 3 つの階層が含まれます。

- Oracle データベース
- Oracle アプリケーション サーバー
- Client アプレット

Performance Explorer を使用して、データベース サーバーとアプリケーション サーバーからサーバー側測定値を収集する必要があります。


データベース サーバーを監視するために、事前定義 Oracle データベース測定値のセットが使用可能です。

### Oracle Forms 監視の前提条件

Oracle Forms を監視するには、お使いの環境が以下の要件を満たす必要があります。

- Oracle Forms アプリケーション サーバーに Oracle Forms Dynamic Monitoring System がインストールされていること。
- 監視を行っているコンピュータからパフォーマンス ページにアクセスできること。

### Oracle Forms を監視する

 **注:** アプリケーション サーバーのデフォルト測定値に加え、Performance Explorer では、Oracle Forms Dynamic Monitoring System (DMS) からパフォーマンス データを問い合わせることができる BDL 監視が提供されます。使用している Oracle Application Server の DMS での設定については、Oracle Technology Network の説明 (<http://otn.oracle.com>) を確認してください。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。**システムの選択** ページが開きます。

省略可能：アプリケーション サーバーを監視するには、アプリケーション サーバーが実行されるプラットフォームで提供される測定値を使用できます。定義済みセットから選択するか、PERFMON、SNMP または REXEC を使用し、該当する測定値を手動で定義します。

4. **Application Server** フォルダ、**Oracle** フォルダの順に展開します。
5. **Oracle Forms AS** を選択し、**次へ** をクリックします。**接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスに、Oracle Application Server が実行されているホストを指定します。必要に応じて、アプリケーション サーバーの **エイリアス** 名を追加できます。
7. **次へ** をクリックします。**属性設定** ページが開きます。
8. 以下の設定を指定します。

- **DMS-Site** の URL。標準的な形式は、http://server:port/dms0 です。
- **Proxy** : プロキシ (HTTP プロキシを使用して接続する必要がある場合)。
- HTTP プロキシの Proxy-Port。
- **Username** : ユーザー名 (DMS サイトにアクセスするために Web サーバーに対して認証が必要な場合)。
- **Password** : DMS サイトにアクセスするためのパスワード。

この監視では、次のような、DMS によって提供される該当の測定値が提供されます。

- 共通の Apache 測定値
- HTTP、OC4J、および PLSQL のモジュール情報
- データベース接続の統計情報

9. **OK** をクリックします。**表示する測定値の選択** ページが開きます。
- 10 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

#### Oracle Forms クライアント側測定値

クライアント側測定値は、Oracle Forms の 負荷テスト中に負荷テスト エージェントによって自動的に生成されます。これらは、負荷テスト コントローラによって 1 つの結果ファイルに集約されます。

エージェントでは、次のような測定値が作成されます。

- コントロールでアクション (たとえば、ボタンを押すアクションまたはコントロールの編集など) が行われた時刻
- アクションで受信したバイト数
- アクションに送信したバイト数
- アクションで受信したメッセージ数
- アクションに送信したメッセージ数
- アクションのラウンドトリップの回数

#### Oracle のパフォーマンスを監視する

Performance Explorer を使用すると、データベース プラットフォームを問わず、Oracle データベース システムからパフォーマンス指標を取得することができます。たとえば、データベースは SunOS、Linux、Windows システムのいずれでも稼働できます。


*Perfmon* を介して Oracle データベースを監視する

## Oracle データベース監視の前提条件

### Windows 版の Oracle クライアントと Oracle カウンタをインストールする

Oracle データベースを監視するには、以下の要件が満たされている必要があります。

- Performance Explorer を実行するコンピュータに SQL\*NET クライアント ソフトウェアがインストールされていること。
- Performance Explorer を実行するマシン上で SQL\*Plus を使用できること。
- Windows パフォーマンス モニタ対応の Oracle カウンタを使用できること。
- Oracle クライアントは、DB 管理者権限を持ち、目的とするデータベース サーバーを認識できること。

 **注:** Windows パフォーマンス モニタ対応の Oracle カウンタが、デフォルトのインストール オプションでインストールされていないこと (たとえば、InstantClient や管理者インストール)。カスタム インストール オプションでのみ使用できます。

### Windows パフォーマンス モニタへの Oracle カウンタの追加

1. ファイル <Oracle Home>%network%admin%tnsnames.ora を開き、監視するデータベースに、接続文字列を追加します。
2. コマンド ラインを起動し、コマンド sqlplus <user>/<password>@<database>を使用して接続をテストします。
3. 次の operfcfg コマンドを使用して監視するデータベースを構成します: <Oracle Home>%bin%operfcfg [-U username] [-P password] [-D TNS\_Alias\_for\_database]  
成功すると、operfcfg: New registry values have been successfully set というメッセージを受信します。
4. **Windows パフォーマンス モニタ > を起動します。スタートファイル名を指定して実行 > Perfmon**
5. これで、Windows パフォーマンス モニタ用の Oracle カウンタを、**カウンタの追加** ダイアログ ボックスから追加できます。

### Oracle のパフォーマンスを監視する (Perfmon)

1. **監視 > データ ソースの追加** を選択します。 **データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。 **システムの選択** ページが開きます。
4. データベース システム フォルダと Oracle 9/10/11 (perfmon) フォルダを展開します。
5. System Statistics ( PERFMON ) ノードをクリックして、**次へ** をクリックします。 **接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスで、Oracle クライアントと Windows パフォーマンス モニタ対応のカウンタが実行されているホストを指定します。
7. **ユーザー名** および **パスワード** テキスト ボックスで、上で指定したホスト上で管理者権限を持つユーザーを指定します。
8. 省略可能: **エイリアス** テキスト ボックスに、Oracle データベースが実行されているホストのエイリアスを指定します。
9. **次へ** をクリックします。
- 10 **表示する測定値の選択** ページが開きます。
- 11 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

## パフォーマンス カウンタ

### Oracle データベース バッファ キャッシュ

カウンタは `phyrds/gets %` です。「`phyrds/gets %`」(物理的読み込み/取得)の割合は、ミス率として計算されます。Miss カウンタの値が低いほど、よい状態です。パフォーマンスを向上させるには、コンピュータ上に使用できるメモリーがあれば、バッファ キャッシュ内のバッファの数を増やします。この値は、時間に基づいて算出される値ではありません。

### Oracle データベース やり直し ログ バッファ

カウンタは `redo log space requests` です。このカウンタの値は、ゼロに近い値である必要があります。この値が継続して増加する場合、REDO ログ バッファ内の領域を探すためにプロセスが待機していることを示しています。この場合、REDO ログ バッファのサイズを大きくすることが必要な場合があります。

### Oracle データベース データ ディクショナリ キャッシュ

カウンタは `getmisses/gets %` です。頻繁にアクセスされるデータ ディクショナリ キャッシュの場合、このカウンタの値は 10 から 15% 未満である必要があります。アプリケーション実行中にこの割合が継続的にしきい値を超える場合は、データ ディクショナリ キャッシュで使用できるメモリーのサイズを大きくします。

キャッシュが使用できるメモリーを増やすには、初期化パラメータ `SHARED_POOL_SIZE` の値を増やします。この値は、時間に基づいて算出される値ではありません。

### Oracle データベース ライブラリ キャッシュ

カウンタは `reloads/pins %` です。これは、再解析を必要とする SQL 文、PL/SQL ブロックおよびオブジェクト定義の割合です。再ロードの合計は、ゼロに近い数である必要があります。確保に対する再ロードの割合が 1% を超える場合は、ライブラリ キャッシュ ミスを減らします。この値は、時間に基づいて算出される値ではありません。

### Oracle データベース DBWR stats1

使用できる 2 つのカウンタ、「`buffers scanned/sec`」(スキャン済バッファ/秒) および「`LRU scans/sec`」(LRU スキャン/秒) は、バッファ キャッシュのチューニングに有効です。スキャン済バッファ/秒は、DBWR によって 1 秒間にスキャンされるバッファの数です。スキャンされるバッファは、LRU (最低使用頻度) リスト内にあります。LRU スキャン/秒は、DBWR によって、LRU (最低使用頻度) バッファ リストが 1 秒間にスキャンされた回数です。

### Oracle データベース DBWR stats2

使用できる 2 つのカウンタ、「`timeouts/sec`」(タイムアウト/秒) および「`checkpoints/sec`」(チェックポイント/秒) は、DBWR が実行を要求されている作業量の判断に有効です。タイムアウト/秒は、DBWR が 1 秒間にタイムアウトする回数です。DBWR のタイムアウトは、3 秒間隔です。DBWR が 3 秒間ポストされない場合、タイムアウトします。

チェックポイント/秒は、データベース ライターによって 1 秒間に処理されるチェックポイント メッセージの数です。チェックポイントに達するたびに、使用済バッファをディスクに書き込むように求めるメッセージが DBWR に送られます (ポストされます)。

### Oracle データベース 動的容量管理

カウンタは `recursive calls/sec` です。動的拡張により、ユーザー プロセスによって発行された SQL 文以外の SQL 文も Oracle データベースによって実行されます。これらの SQL 文は、再帰的呼び出しと呼ばれます。

## Oracle データベース空きリスト

カウンタは free list waits/requests % です。空きリストの競合は、バッファ キャッシュ内の空きデータブロックの競合により影響されます。V\$WAITSTAT を検索すると、空きリストの競合によりパフォーマンスが低下しているかどうかを判断できます。


空きブロックを待機する空きリストの数が、要求の総数の 1% を超える場合は、空きリストを追加することで競合を減らすことを検討してください。

## Oracle データベース ソート

使用可能カウンタは、「sorts in memory/sec」(メモリー ソート/秒) と 「sorts on disk/sec」(ディスク ソート/秒) です。ほとんどのソートの場合、すべてのデータを保持するにはデフォルトのソート領域サイズで十分です。ただし、ソート領域に収まらないデータに対する大規模ソートをアプリケーションで行うことが多い場合は、ソート領域サイズを増やすことを検討してください。

### 既知の問題

Silk Performer では、32 ビットおよび 64 ビット版 Oracle データベースのパフォーマンスを監視できませんが、サポートされているのは 32 ビット版の Oracle クライアント (Windows パフォーマンス モニタ対応 Oracle カウンタ) だけです。

 **注:** 32 ビット版 Oracle クライアントは 64 ビット OS 上で良好に動作し、64 ビット OS が実行されているクライアント マシンから、問題なく 64 ビット データベースを監視できます。

### V\$SYSSTAT を介して Oracle データベースを監視する

Oracle データベースの V\$SYSSTAT テーブルからパフォーマンス情報を取得することで、プラットフォームに依存しない形で DB2 データベース システムを監視できます。

### Oracle 監視の前提条件

Oracle を監視するには、以下の要件が満たされている必要があります。

- Performance Explorer を実行するコンピュータに Oracle Net (Oracle Net Services の一部) クライアント ソフトウェアがインストールされていること。
- Oracle クライアントは 32 ビットバージョンである必要があります、DB 管理者権限を有していて目標のデータベース サーバーを認識できる必要があります。
- 関係のあるプロジェクト属性が、インストールされている Oracle に合わせて既に変更されていること (メニューから **プロジェクト > プロジェクト属性** を選択)。
- Performance Explorer を実行するコンピュータで SQL\*Plus が使用可能であること。
- 指定されたユーザー名、パスワード、サーバー情報を使ってログオンし、クエリ SELECT \* FROM V\$SYSSTAT を実行します。

このログオンの試みが成功すれば、環境はおそらく正しくセットアップされていると思われます。

このログオンの試みが失敗した場合、SQL\*Plus で生成されるエラー メッセージから、問題の原因を判断できます。

### Oracle のパフォーマンスを監視する

Silk Performer プロジェクトを Performance Explorer 内から操作すると、データベース システムのパフォーマンスをリアルタイムに追跡できます。Silk Performer プロジェクトは、OracleMonitoring.sep という名前の Essential パッケージにあり、これは C:\Program Files\Silk\Silk Performer 21.0\Monitors にあります。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。





3. **次へ** をクリックします。 **システムの選択** ページが開きます。
4. **Database System** フォルダ、 **Oracle (v\$sysstat)** フォルダの順に展開します。
5. **System Statistics (v\$sysstat)** ノードを選択して、 **次へ** をクリックします。 **接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスに、 Oracle データベースが実行されているホストを指定します。
7. 省略可能: **エイリアス** テキスト ボックスに、 Oracle データベースが実行されているホストのエイリアスを指定します。
8. **次へ** をクリックします。 **属性設定** ページが開きます。
9. 必要に応じて、属性値を変更します。  
通常、これらの設定には、 Oracle TNS 名、 ユーザー ID、 およびパスワードが含まれます。
- 10 **OK** をクリックします。 **表示する測定値の選択** ページが開きます。
- 11 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、 **完了** をクリックします。

指定したホストへの接続が確立され、 選択した測定値が初期ビューに表示されます。

### SAP のパフォーマンス監視

Performance Explorer を使用して、 SAP システムのパフォーマンスと信頼性指標を監視します。 Performance Explorer には、 SAP インストールを監視するために特に設計された 2 つの監視があります。

1. **リアルタイム監視** タブの **監視** グループで、 **システム** をクリックします。 **データソースウィザード** が開きます。
2. **定義済みのデータソースから選択する** オプション ボタンをクリックします。
3. **Application Server** フォルダ、 **SAP** フォルダの順に展開します。
4. 使用する SAPGUI 監視のタイプに応じて、適切な監視タイプを選択します。  
以下の種類の SAPGUI 監視が使用可能です。
  - SAPGUI Monitoring (ST02): バッファ関連の指標を監視します。 SAP トランザクション ST02 は、この監視が実行されている場合に実行されます。
  - SAPGUI Monitoring (ST03N): アプリケーション固有の指標を監視します。 SAP トランザクション ST03n は、この監視が実行されている場合に実行されます。
  - SAPGUI Monitoring (ST04): データベース関連の指標を監視します。 SAP トランザクション ST04 は、この監視が実行されている場合に実行されます。
  - SAPGUI Monitoring (ST07): SAP システム上のユーザー分散を監視します。 SAP トランザクション ST07 は、この監視が実行されている場合に実行されます。
  - SAPGUI OS-Monitoring (ST06): オペレーティングシステム固有の指標を監視します。 SAP トランザクション ST06 は、この監視が実行されている場合に実行されます。
5. **次へ** をクリックします。  
**接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスに、監視対象となるマシンのホストを指定します。  
この値は、表示目的でのみ使用されるもので、監視自体の内部では使用されません。
7. **次へ** をクリックします。 **属性設定** ページが開きます。
8. 以下の監視固有の属性を定義します。
  - **ConnectionString** : SAP Server への完全な接続文字列。 確かでない場合は、 Silk Performer を使用して、ログオンシーケンスを記録します。 接続文字列は、 SapGuiOpenConnection の最初のパラメータです。
  - **Username** : 監視トランザクションへのアクセス権限を持つ SAP ユーザー名。
  - **Password** : SAP ユーザーのパスワード。
  - **ClientNum** : ログオンプロシージャのクライアント番号 (850 など)。
  - **Language** : ログオンに使用する言語 (EN など)。

- **Entity** : 監視するデータ エンティティ (ダイアログ、RFC、バックグラウンドなど)。
  -  **注: Entity** 属性は、[SAPGUI OS-Monitoring (ST06)] の場合指定する必要はありません。
- **TimeFrame** : SAP は定義した間隔の平均値を提供します。
  -  **注: Timeframe** 属性は、[SAPGUI OS-Monitoring (ST06)] の場合指定する必要はありません。
- **Server** : (ST03N のみ) 監視が必要なサーバー。ST03N で選択されるツリー ノードの名前です。合計では、SAP システム全体からの測定値が返されます。

9. **OK** をクリックします。 **表示する測定値の選択** ページが開きます。

10 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、 **完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## WebLogic (SNMP) の監視

この節では、Performance Explorer を使用して BEA WebLogic を監視する方法について説明します。また、重要な WebLogic パフォーマンス測定値についても説明します。

デフォルトでは、WebLogic SNMP エージェントは稼働していません。SNMP サポートは、WebLogic Administration Console を使って有効にすることができます。このコンソールを開くには、ブラウザに `http://weblogichost:port/console` という URL を入力します。左側のメニュー ツリーで [SNMP] ノードを選択します。 **SNMP コンフィグレーション** ページが開きます。このページで、SNMP エージェントが有効になっているかどうか確認します。SNMP サポートを有効にするには、WebLogic を再起動する必要があります。

### BEA WebLogic (SNMP) を監視する

BEA WebLogic を SNMP 経由で検索するためにデータ ソースへの接続を確立し、監視する測定値を含む初期ビューを作成します。

1. **リアルタイム監視** タブの **監視** グループで、 **システム** をクリックします。 **データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。
3. **次へ** をクリックします。 **システムの選択** ページが開きます。
4. **Application Server** フォルダ、 **BEA WebLogic (SNMP)** フォルダの順に展開します。
5. **SNMP** ノードを選択し、 **次へ** をクリックします。 **接続パラメータ** ページが開きます。
6. **ホスト名** テキスト ボックスに、WebLogic が実行されているホストを指定します。
7. **次へ** をクリックします。 **SNMP 測定値の追加** ページが開きます。使用可能な MIB のリスト ボックスで、 **BEA-WEBLOGIC-MIB** が選択されています。

WebLogic Server のパフォーマンスを監視するエントリとして、次のエントリが使用可能です。

- `bea.wls.jdbcConnectionPoolRuntimeTable`
  - `bea.wls.servletRuntimeTable`
  - `bea.wls.ejbEntityHomeRuntimeTable`
  - `bea.wls.ejbStatelessHomeRuntimeTable`
  - `bea.wls.executeQueueRuntimeTable`
8. 使用可能な JDBC 接続プールをすべて取得するには、 **jdbcConnectionPoolRuntimeEntry** を選択し、 **値の取得** をクリックします。  
たとえば、サンプル アプリケーションの Pet Store の監視の場合、 `ServerRuntime:petstoreServer` という名前の接続プールが返されます。  
接続プールに表示される興味深いパフォーマンス カウンタには、次のカウンタがあります。
    - **JdbcConnectionPoolRuntimeActiveConnectionsCurrentCount** : 現在使用されている接続数を返します。

- **JdbcConnectionPoolRuntimeWaitingForConnectionCurrentCount** : 接続待ちの数を返します。この数は、すべての接続が使用中で、それを超える接続リクエストが接続プールに送られている場合に 0 より大きくなります。
  - **JdbcConnectionPoolRuntimeWaitSecondsHighCount** : 接続プールから接続をチェックアウトするためにクライアントが待つ必要のある最大秒数を返します。
9. アクティブな接続数を監視するには、以前返されたテーブルから **jdbcConnectionPoolRuntimeActiveConnectionsCurrentCount** 列を検索し、セルを選択します。



**注:** 接続プールには、大量のパフォーマンス カウンタがあります。関心のある列を表示するには、いくつかの列のサイズ変更が必要な場合があります。

10 **追加** をクリックします。

11 測定値の収集に必要なコマンドをすべて入力するまで、前のステップを繰り返した後、**閉じる** をクリックします。**表示する測定値の選択** ページが開きます。

12 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## WebLogic (JMX) を監視する

このセクションでは、IIOP および JMX 管理サーバの設定を含む、WebLogic (JMX) Server の監視の前提条件について説明します。

### JMX 管理サーバの有効化

Oracle WebLogic のコンソールで、**コンフィグレーション > 全般 > 詳細** を選択し、JMX 管理サーバを有効にします。MBean サーバの互換性を有効化 と 管理 EJB を有効化 両方をオンにします。これにより、新旧両方の JMX インターフェイスが有効になります。

この変更を有効にするにはシステムを再起動する必要があります。

### IIOP の有効化と構成


WebLogic (JMX) を監視できるようにするには、Oracle WebLogic 11g R1 で IIOP プロトコルを有効にし、構成する必要があります。

1. Oracle WebLogic 管理コンソールのチェンジ センターで、**ロックして編集** をクリックします。
2. コンソールの左ペインで、**環境** を展開し、**サーバー** を選択します。
3. **プロトコル** タブを選択し、**IIOP** を選択します。
4. **IIOP を有効化** チェック ボックスをオンにして、IIOP プロトコルを有効にします。
5. デフォルトの構成を変更するには、**詳細** をクリックします。
6. デフォルトの IIOP ユーザー名とパスワードを指定するには :
  - a) **デフォルト IIOP ユーザー名** フィールドに、ユーザー名を入力します。
  - b) **デフォルト IIOP パスワード** フィールドに、パスワードを入力します。
7. この変更を有効にするには、管理コンソールのチェンジ センターで、**Activate Changes** をクリックします。

この変更を有効にするにはシステムを再起動する必要があります。

### WebLogic (JMX) を監視する

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。**データ ソース ウィザード** が開きます。
2. **定義済みのデータ ソースから選択する** オプション ボタンをクリックします。

3. **次へ** をクリックします。 **システムの選択** ページが開きます。
  4. Application Server を展開し、監視される WebLogic サーバーのバージョンを選択します。  
EJXM または JMX のいずれかを選択します。
  5. **次へ** をクリックします。 **接続パラメータ** ページが開きます。
  6. **ホスト名** テキスト ボックスに、WebLogic が実行されているホストを指定します。
  7. WebLogic **ポート** を指定します (デフォルトは 7001 です)。
  8. ユーザーまたはパスワードは指定しないでください。
  9. **サーバー設定** をクリックします。
  10. **Java ホーム ディレクトリ** および **アプリケーション サーバーのインストール ディレクトリ** を指定します。  
フォルダ lib を含む UNC パスを使用します。例：¥¥<server name>¥Oracle¥Middleware¥wlserver\_10.3¥server¥lib。
-  **注:** UNC パスを指定できない場合は、フォルダ <WL\_HOME>¥server¥lib からローカル マシンへ wljmxclient.jar をコピーし、それをクラスパスに追加します。
11. **OK** をクリックします。
  12. **次へ** をクリックします。 **JMX Data Source Browser** を開きます。
  13. 初期監視ビューに表示する測定値を選択します。
  14. **終了** をクリックします。

## カスタム データ ソース

定義済みのデータ ソースに加えて、Performance Explorer には以下の監視インターフェイスも用意されています。

- PerfMon : すべての Windows 監視データにアクセスできるようにします。
- REXEC : リモート実行プロトコルを介して UNIX ベース システム上のデータ ソースにアクセスできるようにします。
- SNMP : SNMP (Simple Network Management Protocol) を用いて公開される監視データにアクセスできるようにします。

### PerfMon のデータ ソースを監視する

PerfMon は、パフォーマンス カウンタを問い合わせる、Microsoft 固有の監視インターフェイスです。PerfMon のデータ ソースへの接続を確立し、監視する測定値を含む初期ビューを作成します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。
2. **定義済みのデータ ソースから選択する** をクリックし、**次へ** をクリックします。
3. **Custom Data** フォルダを展開して、**NT Performance Monitor Data** をクリックし、**次へ** をクリックします。
4. **ホスト名** ボックスで、監視対象となるマシンを指定します。  
監視対象となるホストは、別の NT ドメインに存在する場合があります。
5. 省略可能 : **エイリアス** ボックスで、エイリアス名を指定します。  
エイリアスは、監視対象サーバーをよく説明する別名にする必要があります。個々のマシン上で測定値をグループ化することをお勧めします。  
たとえば、WebLogic と IIS が両方とも同じコンピュータにインストールされているとしましょう。どちらのサーバーも監視が必要ですが、2 つのパフォーマンス測定値は別々のメニュー ツリーに表示されなければなりません。
6. **ユーザー名** テキスト ボックスに、管理セキュリティ権限を持つユーザーを指定します。
7. **パスワード** テキスト ボックスに、そのユーザーの適切なパスワードを指定します。
8. **次へ** をクリックします。

9. システムで使用可能なパフォーマンス カウンタを検討し、それらを監視テンプレートに追加します。  
リストから複数のカウンタを選択するには、Ctrl または Shift を押します。
- カウンタを選択します。
  - インスタンスの数を選択します。
  - 追加** をクリックします。**カウンタ使用** ページが開きます。
  - 平均測定値** チェック ボックスがオンになっていることを確認します。
  - 次へ** をクリックします。
  - 上記手順を繰り返して、さらにカウンタを追加します。
  - 作業を終えたら、**閉じる** をクリックします。

10 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

### SNMP 経由で検索する

Performance Explorer は、SNMP 経由で検索する定義済みデータ ソースのセットを提供します。ただし、ORADB-MIB のエントリを SNMP データ ソースに追加できます。

SNMP 経由で検索するためにデータ ソースへの接続を確立し、監視する測定値を含む初期ビューを作成します。

SNMP は SNMP エージェントを必要としますが、これは通常、アプリケーション ベンダーから提供されません。たとえば、Oracle データベースは、SNMP エージェントを使用してデータ ソースを公開します。クライアント側のカウンタ パートは Performance Explorer で、これがパフォーマンス データを収集します。

- リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。
- 定義済みのデータ ソースから選択する** をクリックし、**次へ** をクリックします。
- Custom Data** フォルダを展開し、**Snm Data** をクリックしてから、**次へ** をクリックします。
- ホスト名** フィールドに、SNMP エージェントを実行しているホストを指定します。
- 省略可能：**エイリアス** ボックスで、エイリアス名を指定します。

エイリアスは、監視対象サーバーをよく説明する別名にする必要があります。個々のマシン上で測定値をグループ化することをお勧めします。

たとえば、WebLogic と IIS が両方とも同じコンピュータにインストールされているとしましょう。どちらのサーバーも監視が必要ですが、2 つのパフォーマンス測定値は別々のメニュー ツリーに表示されなければなりません。

- 適切な [ポート]、[コミュニティ]、および [バージョン] を指定して、**次へ** をクリックします。バージョン 3 を選択した場合は、コミュニティ フィールドは表示されません。
- バージョン 3 を選択した場合、SNMPv3 の詳細な認証情報を指定し、**次へ** をクリックします。
- 以下のように適切な値を指定して、監視する測定値を作成します。
  - MIB、オブジェクトを 1 つ選択し、オブジェクトの値 (複数) を取得する** リスト ボックスから MIB を選択します。MIB を選択すると、オブジェクトおよびオブジェクト値がリスト ボックスの下の階層 ツリーに表示されます。
  - 測定するオブジェクトを選択します。オブジェクトのプロパティが、リスト ボックスの横のテキスト ボックスに表示されます。
  - 値の取得** をクリックして、オブジェクトを使用できるかどうかを確認します。  
パフォーマンス カウンタの問い合わせに成功すると、オブジェクトの値がリスト ボックスの下のグリッドに表示されます。
- 追加** をクリックして、SNMP 測定値を追加します。

10 測定値の収集に必要なコマンドをすべて入力するまで、前のステップを繰り返した後、**閉じる** をクリックします。**表示する測定値の選択** ページが開きます。

11 初期監視ビューに表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## MIB ファイルを追加して SNMP エージェントを参照する

使用可能なパフォーマンス カウンタを求めて SNMP エージェントを参照するには、パフォーマンス カウンタをテキストで記述した MIB ファイルを持つことが必要です。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。
2. **定義済みのデータ ソースから選択する** をクリックし、**次へ** をクリックします。
3. **Custom Data** フォルダを展開し、**Snmp Data** をクリックしてから、**次へ** をクリックします。
4. **ホスト名** フィールドに、SNMP エージェントを実行しているホストを指定します。
5. 省略可能：**エイリアス** ボックスで、エイリアス名を指定します。  
エイリアスは、監視対象サーバーをよく説明する別名にする必要があります。個々のマシン上で測定値をグループ化することをお勧めします。  
たとえば、WebLogic と IIS が両方とも同じコンピュータにインストールされているとしましょう。どちらのサーバーも監視が必要ですが、2 つのパフォーマンス測定値は別々のメニュー ツリーに表示されなければなりません。
6. 適切な [ポート]、[コミュニティ]、および [バージョン] を指定して、**次へ** をクリックします。
7. **MIB、オブジェクトを 1 つ選択し、オブジェクトの値 (複数) を取得する** リストから **新規 MIB のインポート...** を選択します。
8. コンパイルする MIB ファイルの名前を **MIB ソース ファイル** フィールドに入力します。  
代替方法：参照 (...) ボタンをクリックして、ファイルを検索して選択します。  
一般的に使用される MIB ファイルは、<パブリック ユーザー ドキュメント>%Silk Performer21.0\Data%Mibs%VendorMibs にあります。
9. **インポート** をクリックします。MIB ファイルが **コンパイル済み MIB** リストに追加されます。

## データ ソースの自動検出

データ ソース スキャナは、マシンを調べて監視可能なデータ ソースを検出します。つまり、ユーザーが手動で検索する必要はありません。自動検出は、定義済みのデータ ソースに良く似た設定に基づいて行われるため、すべての使用できるデータ ソースを見つけることができない可能性があります。定義済みのデータ ソースのリストを調べて、それが使用できるかどうかを確認します。

たとえば、データ ソース スキャナは、マシンを調べて、プロセッサの数、ネットワーク インターフェイスの数、データベース インスタンスの数、およびこれらの構成要素の監視方法を確認します。

お使いのシステムが稼働するプラットフォーム (NT、AIX、Linux のいずれか) などの詳細は重要ではありません。CPU やハード ディスクの使用率などの監視対象コンポーネントはプラットフォームに依存しないからです。

### データ ソース スキャナを使用する

データ ソース スキャナは、マシンを調べて監視可能なデータ ソースを検出します。

1. **リアルタイム監視** タブの **監視** グループで、**システム** をクリックします。
2. **データ ソースを検出させる** をクリックして、**次へ** をクリックします。
3. **ホスト名** フィールドで、スキャン対象となるマシンを指定します。
4. 省略可能：**エイリアス** ボックスで、エイリアス名を指定します。  
エイリアスは、監視対象サーバーをよく説明する別名にする必要があります。個々のマシン上で測定値をグループ化することをお勧めします。  
たとえば、WebLogic と IIS が両方とも同じコンピュータにインストールされているとしましょう。どちらのサーバーも監視が必要ですが、2 つのパフォーマンス測定値は別々のメニュー ツリーに表示されなければなりません。
5. **次へ** をクリックします。  
データ ソース スキャナは、システムを調べてさまざまなプラットフォーム上で次のデータ ソースの検出を始めます。

- .NET
- Active Server Pages
- Apache
- Coldfusion
- IBM Universal Database DB2
- IIS
- iPlanet Directory Servers
- Microsoft Proxy Servers
- Microsoft SQL Servers
- Netscape Directory Servers
- Netscape Enterprise 3.0
- Netscape/iPlanet Directory Server
- オペレーティング システムのデータ (ハード ディスク、CPU、ネットワーク インターフェイスなど)
- Oracle データベース
- Rstat (多くの UNIX ベースのシステムで問い合わせ可能で、オペレーティング システム関連のデータを多数提供します)
- SilkPerformer エージェント
- SilkPerformer コントローラ
- Silverstream

システムがスキャンされている間、データ ソース スキャナが特定のデータ ソースに対する資格情報を要求するダイアログ ボックスが表示される場合があります。

スキャナは、システムがリモート NT ドメインにあるかどうかもチェックします。そのような場合、指定されたホストのドメイン名を入力する必要があります。



**注:** 指定されたユーザーは、管理セキュリティ権限を持つ必要があります。NT ベースのシステムの監視にはこの権限が必要です。

**6. 進行状況** セクションに **完了** が表示されたら、**次へ** をクリックします。

**7. 初期監視ビュー** に表示する測定値のチェック ボックスをオンにした後、**完了** をクリックします。

指定したホストへの接続が確立され、選択した測定値が初期ビューに表示されます。

## 負荷テストを監視する

Performance Explorer では、サーバー データの他に、クライアント側の監視のためのデータを収集することができます。

### 負荷テストの監視を構成する

組み込みグラフや監視レポートで、リアルタイムのクライアント側測定値を利用するには、クライアント側データ ソースの接続エントリを追加します。

- 1. リアルタイム監視** タブの **監視** グループで、**負荷テスト** をクリックします。
- 2. ホスト名** フィールドに、負荷テスト コントローラのホスト名を指定します。
- 3. エイリアス** フィールドに、クライアント側測定値を提供するコントローラのエイリアスを指定します。エイリアスは、**クライアント側測定値** ツリーに表示されます。
- 4. 完了** をクリックします。

ツリーにホストが表示されます。このデータ ソースで利用可能なクライアント側測定値が、ツリー内のフォルダに一覧表示されます。



**注:** **ワークロードの設定** ダイアログで **リアルタイム測定値を有効にする** チェック ボックスをオンにして、ランタイム システムがホストの測定値を提供するように指示します。**クライアント側測定値** ツ

リーにある緑色のアイコンは、データソースに正常に接続していることを示します。赤色のアイコンは、データソースに接続されていないことを示します。

## クライアント側測定値を削除する


ツリーからクライアント側測定値ノードを削除できます。ただし、Silk Performer は、その測定値の更新情報を受信した場合、その測定値ノードを置き換えます。


1. ツリー領域から **クライアント側測定値** タブをクリックします。
2. 削除する測定値ノードを右クリックし、**クライアント側データソースの削除** をクリックします。
3. **はい** をクリックします。

## 監視グラフと監視出力

Performance Explorer は、測定した値を表示、保存するために監視グラフと監視レポートを提供します。

- 監視グラフは、データソースの測定した値をリアルタイムに表示します。監視グラフのポーリング間隔を調整できます。
- 監視出力は、データソースの測定した値を記録し、.tsd ファイルに保存します。監視出力は、負荷テストが完了したときに、サーバー側測定値とクライアント側測定値を比較するためにも有用です。


 **注:** ポーリング間隔を短くすると、コンピュータのパフォーマンスを低下させ、結果としてテストの結果に影響を及ぼす可能性があります。規模の大きな負荷テストの場合、間隔は 60 秒以上に設定してください。

 **注:** 監視グラフに表示した値を、後から .tsd ファイルに書き出すことはできません。

## 監視グラフを作成する

監視グラフは、システムの状態を視覚的に把握する場合に便利です。

1. **リアルタイム監視** タブの **新規** グループで、**監視グラフ** をクリックします。
2. **監視** ツリーからグラフに測定値をドラッグします。
3. 測定値リスト内の測定値を右クリックし、**プロパティ** をクリックします。
4. **全般** タブで、タイトルと説明を設定できます。
5. **監視** タブでは、監視と表示に関するオプションを設定できます。**間隔** では、新しい値を取得する更新間隔を定義します。**ページサイズ** では、各ページに表示する間隔の数を定義します。**履歴サイズ** では、グラフの履歴に保存する間隔の数を指定します。
6. **OK** をクリックします。


 **ヒント:** **トランザクションレート監視** を作成することもできます。Performance Explorer は、このグラフに自動的にトランザクション測定値を追加します。監視を開始する前に、**ワークロードの設定** ダイアログの **リアルタイム測定値を有効にする** を確認してください。

## 監視出力を作成する


監視出力を定義すると、負荷テスト中の測定値をキャプチャすることができます。監視出力は、時系列データ (.tsd) ファイルを書き出します。このファイルは、概要レポートに含めたり、グラフのベースとして使用できます。

1. **リアルタイム監視** タブの **新規** グループで、**監視出力** をクリックします。
2. **監視** ツリーから監視出力に測定値をドラッグします。
3. **リアルタイム監視** タブの **コントロール** グループで、**記録** をクリックします。
4. .tsd ファイルのディレクトリを指定して、**保存** をクリックします。Performance Explorer はキャプチャしたすべての値を .tsd ファイルに書き出します。



 **注:** 負荷テストの実行中で、プロファイルに割り当てたワークスペースに監視出力が含まれている場合、すべての監視出力は、テストが開始したときに自動的に書き出しを始め、テストが完了すると停止します。

すべての監視出力は負荷テストの結果ディレクトリに保存されます。

 **注:** SAPGUI データソースについては、監視間隔を 20 秒以上にするをお勧めします。間隔を変更するには、**レポート** タブの **レポート** グループで、**プロパティ** をクリックします。そして、**TSD 出力** タブをクリックして、間隔を変更します。

## ホスト データ

ホストの変更、名前の変更、複製を行ったり、ホストにアクセスするための資格情報を更新することができます。

### ホストの変更


指定したホストを変更したり、ホストのエイリアスを変更することができます。どちらの場合でも、ツリーに表示されている指定した測定値は失われません。


1. **監視** ツリーでホスト (トップレベルのノード) を選択します。
2. **リアルタイム監視** タブの **ホスト** グループで、**変更** をクリックします。
3. ホストを変更する場合は、**ホスト名** に新しいホストの名前を入力し、**エイリアス** を入力して、**次へ** をクリックします。
4. ホストのエイリアス (**監視** ツリーに表示される名前) を変更するには、**ホスト名** に現在のホストを入力し、**エイリアス** に新しい名前を入力して、**次へ** をクリックします。
5. **ユーザー名** と **パスワード** を入力して、**完了** をクリックします。

### ホストを複製する

ホストを複製するとき、Performance Explorer は **監視** ツリーに新しいホスト ノードを作成し、選択したホストに対して指定されているすべての測定値を追加します。

1. **監視** ツリーでホスト (トップレベルのノード) を選択します。このホストが複製されます。
2. **リアルタイム監視** タブの **ホスト** グループで、**複製** をクリックします。
3. **ホスト名** に新しいホストの名前を入力し、**エイリアス** を入力して、**次へ** をクリックします。エイリアスは、**監視** ツリーで新しいホストの名前として表示されます。指定されたデータソースに応じて、さまざまな接続パラメータを持つフィールドが表示されます。これらのフィールドには、元の値が自動的に入力されます。  
指定されたデータソースに構成可能な属性がある場合、**属性の変更** ボタンが表示されます。
4. ホストの資格情報を入力し、**完了** をクリックします。これらの測定値は有効性が確認されます。無効な測定値が検出されると、接続パラメータを変更するか、測定値を無視するかどうかを選択できます。

 **注:** 無効な測定値の複製や名前の変更はできません。

 **注:** 測定値は、データソースおよび接続パラメータが同じ場合に、測定値グループとしてグループ化されます。たとえば、異なる資格情報を持つ PerfMon 測定値がある場合は、別々の測定値グループができます。

### ホストの資格情報を編集する

多くの企業のセキュリティポリシーでは、一定期間を過ぎるとユーザーパスワードが無効になることを要求します。このようなポリシーは、Performance Explorer のようなツールの自動アクセスの妨げになります。そのような環境に対応して、Performance Explorer では既存のホストのユーザー名とパスワードを変更できます。

1. **監視** ツリーでホスト (トップレベルのノード) を選択します。
2. **リアルタイム監視** タブの **ホスト** グループで、**資格情報の編集** をクリックします。
3. 選択したホストの **ホスト名** と **エイリアス** は、あらかじめ指定されています。**次へ** をクリックします。
4. 必要に応じて **属性の変更** をクリックします。
5. 必要に応じて **ユーザー名** と **パスワード** を入力して、**完了** をクリックします。

## CSV から TSD にデータを変換する

Silk Performer は、コマンド ライン ツール (csv2tsd.exe) を提供しており、.csv ファイル (Comma-Separated Values: カンマ区切り値) を .tsd ファイル (Time Series Data: 時系列データ) に変換できます。

.tsd 形式へのデータの変換は、セキュリティ上の理由で Silk Performer から監視できない外部 Windows データを監視する際に役に立ちます。.csv データを .tsd 形式に変換することで、外部データも内部 Silk Performer 監視データと一緒に Performance Explorer 内で分析することができます。

たとえば、パフォーマンス カウンタでリモート サーバーの動作を測定し、そのモニタの結果を .csv ファイルに保存した場合は、その結果を Performance Explorer にインポートして、同じサーバーを必要とする Silk Performer 負荷テストの結果と一緒に分析することができます。Silk Performer を使用すると、2 つのデータ セット間の相関関係を見つけることができます。

## Windows 7/2008 R2 で CSV ファイルをリモート サーバー上で生成する

コマンドライン ツールで指定するパフォーマンス カウンタ ログ (CSV) ファイルはすべて、Windows のパフォーマンス モニタ ツール ( **管理ツール > パフォーマンス** ) で生成される形式にしてください。そのため、csv2tsd.exe を使用する前に、データのインポートが許可される設定でパフォーマンス カウンタを作成してください。

1. Windows のパフォーマンス モニタ ツールを開きます ( **管理ツール > パフォーマンス** )。
2. ツリーを **パフォーマンス > データ コレクター セット > ユーザー定義** の順に展開します。
3. 右クリックしてコンテキスト メニューから **新規作成 > データ コレクター セット** を選択します。
4. 名前を入力し、**手動で作成する (詳細)** を選択して **次へ** をクリックします。
5. データ ログを作成し、**パフォーマンス カウンター** を選択します。 **次へ** をクリックします。
6. CSV ファイルの場所を指定し、**次へ** をクリックします。
7. オプション: **別のユーザーとして実行** で異なるユーザーを指定します。 **保存** をクリックしてダイアログ ボックスを閉じます。
8. 定義したデータ コレクター セットを右クリックし、コンテキスト メニューから **新規作成 > データ コレクター** を選択します。
9. 名前を入力し、**パフォーマンス カウンター データ コレクター** を選択します。 **次へ** をクリックします。
10. **追加** をクリックして、パフォーマンス カウンタを追加します。
11. 結果を追跡するコンピュータ (ローカルとリモートの両方の場合もあります) およびパフォーマンス カウンタ インスタンスを指定します。
12. **このデータ コレクター セットのプロパティを開く** を選択して、**次へ** をクリックします。
13. **パフォーマンス カウンター** タブで、**カンマ区切り** とするログ形式を指定します。
14. **ファイル** タブを選択してファイル名を指定します。
15. **OK** をクリックして、ダイアログを閉じます。
16. パフォーマンス モニタを閉じます。

ログの結果が csv2tsd.exe で TSD 形式に変換されるようにパフォーマンス カウンタを構成すると、パフォーマンス カウンタを使用した動作データの取得を開始できるようになります。

## パフォーマンス カウンタ ログ データを取得する

ログの結果が csv2tsd.exe で TSD 形式に変換されるようにパフォーマンス カウンタを構成すると、パフォーマンス カウンタを使用したりリモート サーバー ログ データの取得を開始できるようになります。

1. Windows のパフォーマンス モニタ ツールを開きます (管理ツール > パフォーマンス)。
2. パフォーマンス ログと警告 ツリー ノードを展開します。
3. カウンタ ログ ノードをクリックします。
4. 右側のウィンドウのカウンタ ログを右クリックし、コンテキスト メニューの **開始** をクリックします。アクティブなカウンタが緑色で表示されます。非アクティブなカウンタは赤色で表示されます。生成された CSV カウンタ ログ ファイルは、次の形式になります。

```
"(PDH-CSV 4.0) (W. Europe Standard Time)(-60)","¥¥JONES¥Processor(_Total)¥% Interrupt Time","¥¥JONES¥Processor(_Total)¥% Privileged Time","¥¥JONES ¥Processor(_Total)¥% Processor Time"
"02/25/2008 23:24:23.826"," "," "
"02/25/2008 23:24:38.841","0","0.10406093659265107","0.10150087105497141"
"02/25/2008
23:24:53.841","0.052084666700800877","1.8229633345280307","2.4454192693999621"
"02/25/2008
23:25:08.840","0.052084666700800877","3.0729953353472514","5.4663299380464103"
"02/25/2008
23:25:23.840","0.052084666700800877","0.57293133370880955","1.0391332684783428"
```

## 適切な形式の CSV ファイルを TSD 形式へ変換する

1. csv2tsd.exe を起動します (スタート > すべてのプログラム > Silk > Silk Performer 21.0 > 分析 ツール > CSV to TSD Converter)。
2. ファイルのインポートおよびエクスポートの設定を次の形式で入力します。Csv2Tsd <csvFile> <tsdfile> [[+/-]<time offset>]。  
<csvFile> は、インポートする CSV ファイルのフルパスです。<tsdFile> は、生成する TSD ファイルのフルパスです。<time offset> は、秒単位の時差です (デフォルトは 0)。
3. Enter キーを押して、変換を実行します。

例 :

```
Csv2Tsd C:¥perflogs¥Test1.csv C:¥perflogs¥Test1.tsd
```

## 組み込みの測定値

測定値は負荷テスト中、または特定の測定ポイントに対する監視セッション中に収集される値をグループ化した最小のエンティティです。測定値は名前と型を持ちます。測定値に関連して測定された一連の値が時系列データ (.tsd) ファイルに格納されます。

測定値にはタイマーとカウンタがあります。タイマーは、レスポンス時間に関するデータを収集し、カウンタは、スループットおよび負荷テスト イベントに関するデータを収集します。

Performance Explorer には大量の測定値がありグループに分類されています。

## 要約測定値

要約測定値 グループは、グローバル レベルの要約された測定値です。他の測定値グループの個別の測定値を集約した測定値や、他の測定値グループには含まれないグローバル レベルの情報を表す測定値タイプです。

このグループは、カウンタのみで構成されており、タイマーは含まれません。測定期間中にのみ収集された、一連のグローバルな測定値 (固定値) であり、負荷テストの準備期間およびシャットダウン期間は除外され

ます。要約測定値グループには、リアルタイム データおよび時系列データ (.tsd) として使用できる測定値タイプがあります。

## 一般的な測定値


概要レポートの先頭に表示される測定値を示します。

測定値	説明
アクティブなユーザー	<p>アクティブな仮想ユーザーの数。仮想ユーザーがアクティブとみなされるのは、開始済みで、現在次のいずれかの状態にある場合です。</p> <ul style="list-style-type: none"> <li>• 実行中</li> <li>• ウォームアップ</li> <li>• データベース待ち</li> <li>• ドキュメントを読み込み中</li> <li>• リソース待ち</li> <li>• 待ち合わせ</li> <li>• 思考時間</li> <li>• シャットダウン</li> </ul> <p>仮想ユーザーが非アクティブとみなされるのは、現在次のいずれかの状態にある場合です。</p> <ul style="list-style-type: none"> <li>• 開始中</li> <li>• キュー待ち</li> <li>• ユーザー待ち</li> <li>• 停止待ち</li> <li>• ペーシング時間</li> <li>• 中断</li> </ul>
トランザクション	<p>正常に終了したかどうかにかかわらず、終了した Silk Performer トランザクションの数。</p> <p>概要レポートには、以下の内容が表示されます。</p> <ul style="list-style-type: none"> <li>• 1 秒間に終了したトランザクションの数を示すグラフ</li> <li>• 1 秒間に終了したトランザクション数の平均</li> <li>• 終了したトランザクションの合計数</li> </ul>
エラー	<p>インターネット、データベース、およびミドルウェアの API などの API エラーの数。深刻度が エラー 以上 (トランザクション終了またはプロセス終了) の場合に、API エラーと見なされます。重大度が 通知 または 警告 の場合は、無視されます。</p>
実行中のユーザー	<p>実行中の仮想ユーザーの数。仮想ユーザーが実行中とみなされるのは、開始済みで、停止されていない場合です。測定値 <b>アクティブ ユーザー</b> と異なり、次のいずれかの状態にある仮想ユーザーも含まれます。</p> <ul style="list-style-type: none"> <li>• キュー待ち</li> <li>• ユーザー待ち</li> <li>• 停止待ち</li> <li>• ペーシング時間</li> </ul>

## エージェントの健全度管理測定値

エージェントの健全度を分析するために使用する測定値について説明します。

測定値	説明
CPU 使用率	<p>プロセッサがタスクを実行するのにビジー状態である時間のパーセンテージ。プロセッサ使用率が 100% か 100% に近い場合、そのコンピュータの利用可能な処理能力が不足しています。</p>

測定値	説明
メモリ	<p>ドライバマシンによって割り当てられた (コミットされた) メモリ容量が、物理 RAM の総量に占めるパーセンテージ。高いメモリ使用率は、そのドライバマシンの利用可能なメモリが少ないことを示し、コンピュータ全体のパフォーマンスが低下するディスク スワップの原因となることがあります。</p> <p> <b>注:</b> オペレーティングシステムによっては、タスク マネージャが異なるメモリ統計値を示すことがあります。</p>
応答性	<p>大部分の Silk Performer プロジェクトタイプでは、応答性の測定は内部測定システムの健全度によって直接決定されます。</p> <p>SAP GUI、Citrix、ブラウザ駆動負荷テスト (BDLT) タイプの Silk Performer プロジェクトでは、応答性の測定はテスト対象のアプリケーション (SAP GUI または perfBrowserHost.exe) の健全度も考慮に入れます。この健全度は内部のアプリケーションの健全度の値によって表されます。全体の応答性の測定値はこれらの 2 つの健全度の値の悪い方の値となります。</p>

## インターネット関連の測定値

インターネット接続に関する測定値について説明します。

測定値	説明
同時接続数	現在開かれている TCP/IP 接続の数。
送信されたリクエスト	低レベルの Web 関数が送信した HTTP リクエストの数、TCP/IP レベルの関数が送信したデータ パケットの数、あるいは POP3、SMTP、FTP、または LDAP の関数呼び出しによって送信されたコマンドの数。Silk Performer では、LDAP トラフィックもこの測定値に含まれ、送信された LDAP リクエストの数が報告されます。IIOP は別のカウンタを使用します。
失敗したリクエスト	送信または実行が正常に行われなかった、上述のリクエストの数。IIOP は別のカウンタを使用します。
送信されたリクエスト データ	サーバーに送信されたデータ量 (KB)。TCP/IP 関連のすべてのトラフィック (HTTP、ネイティブ TCP/IP、IIOP、POP3、SMTP、FTP、および LDAP)、SSL/TLS を介したセキュアなトラフィックだけでなく、ヘッダー情報およびボディのコンテンツ情報も含まれます。ただし、SSL/TLS を使用した暗号化、またはその他の API 関連のトラフィック (TUXEDO ATMI、ODBC、Oracle OCI など) によって発生するデータのオーバーヘッドは含まれません。
受信したレスポンス	低レベルの Web 関数が受信した HTTP レスポンスの数、TCP/IP レベルの関数が受信したデータ パケットの数、および POP3、SMTP、FTP、および LDAP の関数呼び出しによって受信されたレスポンスの数。IIOP は別のカウンタを使用します。
失敗したレスポンス	受信するはずだったが、正常に受信されなかったレスポンスの数。IIOP は別のカウンタを使用します。
受信したレスポンス データ	サーバーから受信したデータ量 (KB)。TCP/IP 関連のトラフィック (HTTP、ネイティブ TCP/IP、IIOP、POP3、SMTP、FTP、および LDAP)、SSL/TLS を介したセキュアなトラフィックだけでなく、ヘッダー情報およびボディのコンテンツ情報も含まれます。API 関連のトラフィック (TUXEDO ATMI、ODBC、Oracle OCI など) は含まれません。
正常な接続	低レベル、TCP/IP レベル、IIOP、SMTP、POP3、FTP、または LDAP の各関数によって確立された、リモートホストに対する正常な TCP/IP 接続の数。

測定値	説明
失敗した接続	無効なホスト名、無効な IP アドレス、無効なポート番号、利用できないリモート ホスト、ネットワーク関連の問題、または認証の失敗が原因で失敗した TCP/IP 接続の数。このカウンタは、低レベル、TCP/IP レベル、IIOP、SMTP、POP3、FTP、および LDAP の関数で使用されます。デフォルトでは、低レベルの Web 関数は 3 回まで (再試行は 2 回まで) 接続を試みます。試行がすべて失敗した場合のみ、失敗が記録されます。
再試行された接続	1 回目の接続に失敗した場合に、サーバーへの接続を試みた回数。
ページ データ	リクエストと、すべての埋め込みオブジェクトを含む Web ページ全体の取得に関する、すべてのリクエスト データおよびレスポンス データの合計 (KB)。
埋め込みオブジェクト データ	Web ページに埋め込まれたオブジェクトのリクエストおよび取得に関連するすべてのリクエスト データおよびレスポンス データの合計 (KB)。

## Web 固有の測定値

Web に固有の測定値について説明します。

測定値	説明
再試行された HTTP リクエスト	クライアントがリクエストを再試行しなかった回数。クライアントがリクエストを送信しようとしている間にタイムアウトが発生したために、サーバーが継続的な接続 (キープアライブが有効、または HTTP/1.1 を使用した場合に利用可能) を切断した場合に、発生することがあります。このカウンタは、低レベルの Web 関数でのみ使用されません。
HTTP リダイレクト	Web サーバーのレスポンスのステータス コードが 3xx だったために、実行された HTTP リダイレクトの数。この測定値を増加させるのは、Web 関数のみです。
HTTP 再認証	Web サーバーから返されたステータス コードが 401 または 407 だったために、必要となった再認証の数。このカウンタは、基本認証を実行する Web の関数でのみ使用されます。
HTTP ヒット数	Web サーバーに届いた HTTP リクエストの数。この測定値を増加させるのは、Web 関数のみです。
失敗したヒット数	失敗した HTTP リクエストの数。この測定値を増加させるのは、Web 関数のみです。
HTTP 1xx レスポンス	ステータス コードが、コンテンツのステータスが informational であることを示す、HTTP レスポンスの数。この測定値を増加させるのは、Web 関数です。
HTTP 2xx レスポンス	ステータス コードが、リクエストが成功したことを示す、HTTP レスポンスの数。この測定値を増加させるのは、Web 関数です。
HTTP 3xx レスポンス	ステータス コードが、リクエストがリダイレクトされたことを示す、HTTP レスポンスの数。この測定値を増加させるのは、Web 関数です。
HTTP 4xx レスポンス	ステータス コードが、クライアント エラーを示す、HTTP レスポンスの数。この測定値を増加させるのは、Web 関数です。
HTTP 5xx レスポンス	ステータス コードが、サーバー エラーを示す、HTTP レスポンスの数。この測定値を増加させるのは、Web 関数です。
HTTP キャッシュの条件付き再ロード	コンテンツが最後にキャッシュに置かれてから、その内容が変更された場合に、項目をダウンロードするために、サーバーに送信されたリクエストの数。
HTTP キャッシュのヒット数	リクエストがキャッシュで処理された回数。

測定値	説明
送信された HTTP クッキー	低レベルの Web 関数が送信したクッキーの数。HTTP クッキー リクエストのヘッダー行は複数のクッキーを含み、そのヘッダー内の各クッキーが測定されます。
受信した HTTP クッキー	低レベルの Web 関数が受信したクッキーの数。HTTP レスポンスは、複数のクッキー ヘッダー行を含み、それらのヘッダー行の各クッキーが測定されます。
HTTP ページ	ページレベルの Web API がリクエストした HTML ページの数。

## IIOP 関連の測定値

IIOP 操作に関する、スクリプト化された測定値について説明します。

測定値	説明
IIOP メッセージ	Silk Performer が送信した IIOP メッセージの数。
IIOP リクエスト	Silk Performer が送信した IIOP リクエスト メッセージの数。
IIOP ロケート リクエスト	Silk Performer が送信した IIOP ロケート リクエスト メッセージの数。
送信された IIOP フラグメント	Silk Performer が送信したフラグメント化された IIOP メッセージの数。
IIOP レスポンス (OK)	Silk Performer が受信した、ステータスが OK の IIOP レスポンス メッセージの数。
IIOP レスポンス (例外)	Silk Performer が受信した、ステータスが 例外発生 の IIOP レスポンス メッセージの数。ユーザー例外とシステム例外が含まれます。
IIOP レスポンス (オブジェクトが移動)	ステータスが オブジェクト移動済み の IIOP レスポンス メッセージの数。このステータスのメッセージを受信すると、Silk Performer は新しい場所にリクエストを再発行します。
IIOP リクエスト タイムアウト	保留されたレスポンスを待っている間に、データ受信の時間制限 (負荷テスト スクリプトで指定) を超過した回数。時間制限を超過した場合は常に、レスポンスは受信されず、Silk Performer は IIOP の CancelRequest メッセージを送信して、中止された操作をサーバーに通知します。
IIOP ロケート レスポンス (オブジェクトが存在)	Silk Performer が受信した、ステータスが オブジェクトが存在 の IIOP の LocateReply メッセージの数。
IIOP ロケート レスポンス (オブジェクトが不明)	Silk Performer が受信した、ステータスが オブジェクトが不明 の IIOP の LocateReply メッセージの数。
IIOP ロケート レスポンス (オブジェクトが移動)	Silk Performer が受信した、ステータスが オブジェクトが移動 の IIOP の LocateReply メッセージの数。
IIOP メッセージ エラー	Silk Performer が受信した IIOP エラー メッセージの数。
受信した IIOP フラグメント	Silk Performer が受信したフラグメント化された IIOP メッセージの数。
IIOP ロケート リクエスト タイムアウト	保留されたロケート レスポンスを待っている間に、データ受信の時間制限 (負荷テスト スクリプトで指定) を超過した回数。時間制限を超過した場合は常に、ロケート レスポンスは受信されず、Silk Performer は IIOP の CancelRequest メッセージを送信して、中止された操作についてサーバーに通知します。

## データベース固有の測定値


データベースに特有な測定値について説明します。

測定値	説明
SQL コマンド	実行のために DBMS に送信された SQL コマンドの数。

測定値	説明
開いているデータベース ログイン	現在開いているデータベース セッションを測定します。OCI、ODBC、および ODBC の高水準インターフェイスに関連します (この測定値は、レポート ファイルでは使用できません)。
開いているデータベース カーソル	現在開いているデータベース カーソル (ハンドル) を測定します。OCI、ODBC、および ODBC の高水準インターフェイスに関連します。ODBC の場合、この測定値が減らされるのは、SQL_DROP オプションを伴う OdbcClose が呼び出されたときのみです。(この測定値は、レポート ファイルでは使用できません。)

## トランザクションの測定値

トランザクションの測定値グループは、トランザクション レベルのレスポンス時間データです。負荷テスト スクリプトで定義されているトランザクションごとに、この測定値グループが 1 つ自動的に作成され、トランザクション名が測定値用のキーとして使われます。カウントされた値を含む測定値のみが表示されます。つまり、トランザクションが 1 つも失敗していない場合、Trans.failed は表示されません。

 **注:** トランザクション TInit および TEnd は、負荷テスト測定値に対してカウントされません。開始および終了トランザクションは、テストしているアプリケーションに関連する測定値を含んでいないはずだからです。

対応する測定値タイプを次の表で説明します。

測定値	説明
トランザクション OK	正常なトランザクションのレスポンス時間 (秒)。トランザクション内で実行された API 関数呼び出しがエラー メッセージを返していない場合に、そのトランザクション時間がこの測定値タイプとして測定されます。
トランザクション (ビジー) OK	正常なトランザクションのビジー時間 (秒)。トランザクション ビジー時間は、トランザクション レスポンス時間から思考時間、待機時間、関数変換に要する時間を省いた時間です。トランザクション内で実行された API 関数呼び出しがエラー メッセージを返していない場合に、そのトランザクション時間がこの測定値タイプとして測定されます。
トランザクション失敗	失敗したトランザクションのレスポンス時間 (秒)。トランザクション内で重大度が エラー 以上のエラーが、1 回以上発生した場合に、そのトランザクション時間がこの測定値タイプとして測定されます。
トランザクション (ビジー) 失敗	失敗したトランザクションのビジー時間 (秒)。トランザクション ビジー時間は、トランザクション レスポンス時間から思考時間、待機時間、関数変換に要する時間を省いた時間です。トランザクション内で重大度が エラー 以上のエラーが、1 回以上発生した場合に、そのトランザクション時間がこの測定値タイプとして測定されます。
トランザクション キャンセル	キャンセルされたトランザクションのレスポンス時間 (秒)。スクリプト内の return 文が 戻り値 1 で発行された場合に、そのトランザクション時間がこの測定値タイプとして測定されます。

## Web フォーム測定値

Web フォーム測定値グループは、負荷テスト スクリプトの dclform セクションで宣言されているフォームに関する測定値タイプです。Web フォーム測定値グループは、スクリプト化された関数呼び出しによって送信される POST、GET、および HEAD コマンドでのフォーム送信の、レスポンス時間およびスループットを示します。dclform セクションで宣言されているフォームごとに、フォーム名をキーとして、測定値グループが自動的に作成されます。対応する測定値タイプを次の表で説明します。



測定値	説明
サーバー ビジー時間	測定 (秒) は、ユーザーがリクエスト全体を送信し終わった後に開始され、ユーザーがレスポンスの最初の部分を受信したときに終了します。
ラウンドトリップ時間	リクエストを送信してから適切なレスポンスを受信するまでの時間 (秒)。接続が継続的でない場合は、接続およびシャットダウンの時間も含まれます (キープアライブ オプションが有効な場合、または HTTP/1.1 が使用されている場合は、接続が継続されます)。
ヒット数	成功したフォーム送信の数。
送信されたリクエスト データ	フォームの送信中に Web サーバーに送信されたデータ量 (KB)。HTTP ヘッダーおよびボディ コンテンツの情報も含まれます。カウントは、送信された Web サーバー リクエストの数を示します。
受信したレスポンス データ	フォーム送信のレスポンスとして Web サーバーから受信したデータ量 (KB)。HTTP レスポンスのヘッダーおよびボディ コンテンツの情報も含まれます。カウントは、受信した Web サーバー レスポンスの数を示します。

## Web ページ測定値

Web ページ測定値グループは、Web ページに関する測定値タイプです。負荷テスト中にダウンロードされた Web ページごとに、Web ページ名をキーとして、測定値グループが自動的に作成されます。対応する測定値タイプを次の表で説明します。


測定値	説明
ページ時間	Web ページ全体をサーバーから取得するのにかかった時間 (秒)。すべての HTML ドキュメント、および、すべての埋め込みドキュメント (画像、ビデオ、および音声ファイルなど) の取得にかかった時間です。
ドキュメント ダウンロード時間	Web ページ内のすべての HTML ドキュメントを取得するのにかかった時間 (秒)。
サーバー ビジー時間	サーバーで Web ページ リクエストを処理するのにかかった時間 (秒)。この測定は、仮想ユーザーがリクエストの最後の 1 バイトを送信し終わった後に開始され、ユーザーがレスポンスの最初の 1 バイトを受信したときに終了します。
ページ データ	リクエストと、すべての埋め込みオブジェクトを含む Web ページ全体の取得に関係する、すべてのリクエスト データおよびレスポンス データの合計 (KB)。
埋め込みオブジェクト データ	リクエストと、Web ページに埋め込まれたオブジェクトの取得に関係する、すべてのリクエスト データおよびレスポンス データの合計 (KB)。

## ブラウザ駆動型の測定値

対応する測定値タイプを次の表で説明します。


測定値	説明
DOM インタラクティブ (DOM Interactive)	リクエストからブラウザがすべての HTML 要素を解析し、DOM を構築するまでの期間を表します。

測定値	説明
DOM 完了 (DOM Complete)	リクエストからブラウザがすべてのリソース (画像、スタイルシート、スクリプトなど) をダウンロードし、処理を完了するまでの期間を表します。
ロード終了 (Load End)	リクエストからブラウザが onLoad 関数の実行を完了するまでの期間を表します。すべてのページのロード処理の最後のステップとして、ブラウザは onLoad 関数をトリガーする onLoad イベントを送信します。onLoad 関数が実行されると、さらにアプリケーションのロジックが実行される場合があります。
描画開始 (First Paint)	リクエストからページの表示が始まるまでの期間を表します。この測定値は、Internet Explorer の場合にのみ使用できます。
操作開始時間 (Time to Interact)	リクエストからすべての TTI 関連要素がページ上で利用可能になるまでの期間を表します。この時点で、ユーザーは Web ページを操作できるようになります。
アクション時間 (Action Time)	リクエストからブラウザがすべてのリソースをダウンロードし、処理を完了するまでの期間を表します。この期間の終了は、定義された同期モードによって異なります。同期モードが HTML の場合、onLoad 関数の完了時にアクション時間が終了します。同期モードが AJAX の場合、非同期アプリケーション ロジック フェーズ中にアクション時間が終了します。

 **重要:** TTI 機能を使用すると、ここに挙げた測定値以外のものは、利用できなくなったり、正しくない値が表示されることがあります。

## CORBA 固有の測定値

IIOP 測定値グループは、負荷テスト スクリプトの中で呼び出される IIOP 操作に関する測定値タイプです。API 関数 `IiopRequest` が発行した、IIOP 操作呼び出しのレスポンス時間とスループット比率を示します。`IiopObjectSetMeasures` 関数を使って、特定の CORBA オブジェクトの操作に関連付ける、測定値グループ名の接頭辞を指定できます。指定しない場合、操作の名前が対応する測定値グループの名前として使用されます。対応する測定値タイプを次の表で説明します。


測定値	説明
ラウンドトリップ時間	<code>IiopRequest</code> 関数呼び出し処理開始から、CORBA オブジェクトの応答処理終了までの測定時間 (秒)。
サーバー ビジー時間	測定 (秒) は、ユーザーが CORBA 操作呼び出しの最後の 1 バイトを送信した後に開始され、ユーザーがレスポンスの最初の 1 バイトを受信したときに終了します。   <b>注:</b> アプリケーションおよびリクエストの内容によっては、サーバー ビジー時間は最後の 1 バイトが送信される前に開始され、ユーザーが最初の 1 バイトを受信した後も継続されることがあります。そのため、サーバー ビジー時間は、多くの場合近似値にすぎません。

ブロッキングされていない IIOP リクエストについて、これらの測定値は、同じ方法で測定されます。このため、測定時間すべての合計が、負荷テストの時間全体より長くなる場合があります。

CORBA オブジェクトにリクエストをリダイレクトされた場合、`IiopRequest` 関数はそのリクエストを別の場所に再発行しなければなりません。その結果、リクエスト メッセージとレスポンス メッセージのペアが 2 つになり、両方が別々に測定されます。`IiopRequest` 関数の測定時間の場合と同様に、両方の (別々に測定された) リクエストとレスポンスのペアの合計が表示されます。

## SQL データベース固有の測定値

対応する測定値タイプを次の表で説明します。

測定値	説明
Parse	<p>SQL コマンドの準備 (Oracle の用語では "解析") にかかった時間 (秒)。SQL コマンドの準備には、通常、SQL コマンドをサーバーに送信するときのサーバー ラウンドトリップ時間が含まれます。SQL コマンドは、サーバーで解析され、その後でその SQL コマンドが呼び出されたとき再利用できる内部形式に変換されます。</p> <p> <b>注:</b> ORA_DEFERRED オプションを用いて OraParse 関数を使用した場合、SQL コマンドの解析は実行ステップまで遅延されます。そのため、解析ステップで、サーバー ラウンドトリップ時間が生じます。この場合、この測定値タイプは無意味です。</p>
Exec	<p>SQL コマンドの実行にかかった時間 (秒)。SQL コマンドの実行には、SQL コマンドの実行を可能にするために、コマンドを送信して結果が返されるまでのサーバー ラウンドトリップが、少なくとも 1 回含まれます。1 つの SQL コマンドを処理するのに、サーバーでは多くのデータ処理が必要なので、SQL コマンドの実行は、通常、データベースサーバーにかかる負担の大部分を占めます。SQL 測定値グループには、データの取得に関する、別の測定値タイプがあります。データの取得は、実行ステップの一部にするか、別のステップにするか設定できます。後者の場合、結果は SQL 測定値グループに含まれません。</p>
ExecDirect	SQL コマンドの準備および実行にかかった時間 (秒)。

## ミドルウェア固有の測定値

TUXEDO 測定値グループは、負荷テスト スクリプトで実行される TUXEDO サービス呼び出しに関する測定値タイプです。関数 (Tux\_tpcall、Tux\_tpacall、Tux\_tpsend、Tux\_tprecv、Tux\_tpgetrply、Tux\_tpenqueue、Tux\_tpdequeue) が発行した、TUXEDO サービス呼び出しのレスポンス時間およびスループットを示します。これらの TUXEDO 関数で 사용되는 TUXEDO サービスごとに、サービス名をキーとして測定値グループが自動的に作成されます。対応する測定値タイプを次の表で説明します。

測定値	説明
レスポンス時間	<p>TUXEDO サービス呼び出しのレスポンス時間 (秒)。同期サービス呼び出しの場合、レスポンス時間には、サーバーでその呼び出しを処理するのにかかる時間が含まれます。非同期サービス呼び出しの場合、Silk Performer は、リクエストの送信からレスポンスの受信までの間に経過した時間に、リクエスト待ちの時間を加えて報告します。非同期サービス呼び出しの場合は、サーバーでサービスを実行するのにかかる時間は測定されません。送信呼び出しおよびリクエスト呼び出しの場合、Silk Performer は、開かれている接続にデータを送信する時間と、開かれている接続からデータを受信する時間を測定します。キューイング通信の場合、サービス呼び出しをキューに送信するのにかかった時間、およびサービス呼び出しの結果を取得するのにかかった時間の両方が測定されます。</p>
送信されたリクエスト データ	<p>特定のサービスを送信するために TUXEDO サーバーに送信されたデータの量 (KB)。Silk Performer は、ネットワークに転送されたバイト数ではなく、サービス呼び出しによって使用されたアプリケーション データバッファのサイズを測定します。</p>
受信したレスポンス データ	<p>特定のサービスが TUXEDO サーバーから受信したデータの量 (KB 単位)。Silk Performer は、ネットワークに転送されたバイト数ではなく、</p>

測定値	説明
	サービス呼び出しによって使用されたアプリケーション データ バッファのサイズを測定します。

## Citrix 固有の測定値


Citrix の同期点測定値グループは、既に実行済みの同期関数の数を示します。この測定値の主な目的は、トランザクションの進行状況を視覚化することです。

測定値	説明
Citrix の同期点	実行済みの Citrix 同期関数 (CitrixWaitForXXX) の数。

## カスタム タイマ測定値

カスタム タイマ測定値グループは、負荷テスト スクリプト内で MeasureStart 関数および MeasureStop 関数を使って定義される一連のタイマです。

測定値	説明
レスポンス時間	MeasureStart 関数呼び出しから MeasureStop 関数呼び出しまでの測定時間 (秒)。
ビジー時間	MeasureStart 関数呼び出しから MeasureStop 関数呼び出しまでを測定したビジー時間 (秒)。 ビジー時間は、レスポンス時間から思考時間、待機時間、関数変換に要する時間を省いた時間です。

 **注:** MeasurePause 関数または MeasureResume 関数を使って、タイマの一時停止または再開ができます。

## カスタム カウンタ測定値

カスタム カウンタ測定値グループは、負荷テスト スクリプト内で MeasureInc 関数を使って定義される、一連のカウンタです。

測定値	説明
カスタム カウンタ	MeasureInc 関数で設定された値をカウントします。

## ストリーミングの測定値

対応する測定値タイプを次の表で説明します。

測定値	説明
実ビデオ再生時間	すべてのダウンロードしたメディア セグメント (完全なストリーム) の再生時間 (秒)。
総ダウンロード時間	すべてのメディア セグメントのダウンロード時間 (秒)。
セグメント数	すべてのダウンロードしたメディア セグメントの数。
クリティカル D/P 比セグメント数	0.9 以上の D/P 比 (ダウンロード時間/再生時間比率) のメディア セグメントの数。

測定値	説明
ストリーム D/P 比	再生時間に対するダウンロード時間の比率 (D/P 比)。値が 0.9 を超える場合は、ラグの発生や、ちらつき、バッファリングが発生する可能性を示します。
ストリーム サイズ	すべてのダウンロードしたメディア セグメント (完全なストリーム) のサイズ (キロバイト)。
最初のセグメントにかかる時間	ページの読み込みから最初のメディア セグメントのダウンロードの完了までの経過時間。

## コマンド ライン パラメータ

Performance Explorer には、コマンド ライン インターフェイスから実行できるパラメータがいくつか用意されています。

### 使用できるコマンド

次の表に、コマンド ライン インターフェイスで使用できる Performance Explorer コマンドの一覧を示します。

コマンド	説明
/TSD:<ファイル>	指定した .tsd ファイルを開きます。
/ADDLOADTEST:<ファイル>	指定した .tsd ファイルを含む負荷テスト結果を開きます。
/OVT:<テンプレート>	指定したテンプレートを、概要レポートを手動生成する際にあらかじめ選択されている概要レポート テンプレートとして設定します。
/MONITORDIR:<ディレクトリ>	監視出力ファイル用の出力ディレクトリを設定します。
/MONCREATE:<ディレクトリ>	監視モードで始めます (BDL スクリプトの生成)。
/ACTION:STDGRAPH	一般タイプの定義済みのグラフを開きます。指定した .tsd ファイル (/TSD:<ファイル>) を使用します。
/ACTION:SELGRAPH	ウィザードを開き、グラフのテンプレートを選択します。指定した .tsd ファイル (/TSD:<ファイル>) を使用します。
/ACTION:MONSERVER	サーバー監視ウィザードを開始します。
/ACTION:STARTALLMONITORS	すべての監視グラフを開始します。
/ACTION:STOPALLMONITORS	すべての監視グラフを停止します。
/ACTION:OVERVIEWREPORT [:<テンプレート>]	指定した .tsd ファイル (/TSD:<ファイル>) に対する概要レポートを生成します。テンプレート ファイルを適用できます。マージした .tsd ファイル (m@) を指定してください。
/ACTION:WORDOVERVIEWREPORT	指定した .tsd ファイル (/TSD:<ファイル>) に対する Word 概要レポートを生成します。マージした .tsd ファイル (m@) を指定してください。パラメータ /WORDTEMPLATE と /WORDTARGET は省略可能です。

コマンド	説明
/WORDTARGET:<ファイル>	す。Performance Explorer を GUI なしモードで実行します。 生成する Word 文書 (.docx) の絶対パス。このパラメータは、パラメータ / ACTION:WORDOVERVIEWREPORT より前に指定する必要があります (以下の例を参照)。
/WORDTEMPLATE:<ファイル>	生成する Word 概要レポートのテンプレートとして使用される Word 文書 (.docx) の絶対パス。このパラメータは、パラメータ / ACTION:WORDOVERVIEWREPORT より前に指定する必要があります (以下の例を参照)。
/EXPORTOVR:<ファイル>	最近生成した概要レポート ビューを <ファイル> にエクスポートします。HTML の場合は file.htm を、Web アーカイブへのエクスポートの場合は file.mht を使用します。
/NOGUI	Performance Explorer を表示状態か非表示状態 (GUI なしモード) のどちらで実行するかを指定します。 / NOGUI は、DOS ベースのコマンドラインで /NOEXIT が使用されている場合にのみ適切に機能します。
/SINGLEINSTANCE[:NOSTARTUP]	既存のインスタンスを検索します。NOSTARTUP も指定すると、新しいインスタンスは作成されません。
/PEW:<ワークスペース ファイル>	指定したワークスペースを開くか作成します。 / SINGLEINSTANCE も指定すると、既存のインスタンスが指定のワークスペース ファイルを使用している場合、そのインスタンスのみが選択されます。指定しない場合、新しいインスタンスが作成されます。
/COMMANDFILE:<XML ファイル>	指定した XML ファイルからコマンドをロードします。
/EXIT	.tsd ファイルを保存した後、ワークスペースに対する変更を保存せずに Performance Explorer を終了します。
/NOEXIT	Performance Explorer をコマンド ラインの実行後開いたままにします。
/?	Performance Explorer で使用できるコマンド ライン パラメータを表示します。

### 使用例

上記のコマンドのいくつかの使用方法を以下に示します。

最初の例に示すように、パラメータ /TSD を PerfExp.exe コマンドに指定する場合、Performance Explorer が開かれ、指定した .tsd ファイルが表示されます。さらに、パラメータ /ACTION:STDGRAPH と /ACTION:STARTALLMONITORS を指定すると、定義済みのグラフ (一般タイプ) が開かれ、すべての監視が自動的に開始されます。

```
PerfExp.exe /TSD:"c:%test.tsd" /ACTION:STDGRAPH /
ACTION:STARTALLMONITORS
```

この 2 番目の例では、Performance Explorer を開始し、指定した .pew ファイルを開きます (/PEW)。さらに、監視ディレクトリが設定され (/MONITORDIR)、.pew ファイルに含まれるすべての監視が自動的に開始されます (/ ACTION:STARTALLMONITORS)。これらすべてのアクションは、ユーザー インターフ

エイスを表示せずに実行されます (/NOGUI)。最後に、パラメータ /SINGLEINSTANCE によって、Performance Explorer の既存のインスタンスが検索され、パラメータ /NOEXIT によって、コマンド ラインの実行後に Performance Explorer が開いたままになります。

```
PerfExp.exe /PEW:"c:¥test.pew" /MONITORDIR:"c:¥temp¥perf" /  
ACTION:STARTALLMONITORS /NOGUI /SINGLEINSTANCE /NOEXIT
```

3 番目の例は、次のような動作になります。Performance Explorer を開始し、指定した .pew ファイルを開きます (/PEW)。起動中に Performance Explorer は既存のインスタンスを検索します。インスタンスが見つかったら、新しいインスタンスは開始されません (/SINGLEINSTANCE)。監視ディレクトリが設定され (/MONITORDIR)、すべての監視が停止します (/ACTION:STOPALLMONITORS)。ユーザー インターフェイスは表示されません (/NOGUI)。最後に、.tsd ファイルが保存され、ワークスペースへの変更は保存されずに Performance Explorer が閉じます (/EXIT)。

```
PerfExp.exe /PEW:"c:¥test.pew" /MONITORDIR:"c:¥temp¥perf" /  
ACTION:STOPALLMONITORS /NOGUI /SINGLEINSTANCE:NOSTARTUP /EXIT
```

4 番目の例は、Word 概要レポートを生成します。まず、ユーザー インターフェイスを表示せずに Performance Explorer が開始され (パラメータ /ACTION:WORDOVERVIEWREPORT により)、指定した .tsd ファイルが開きます (/TSD)。マージした .tsd ファイル (m@) を指定してください。それから、生成される Word 文書のパスと名前 (/WORDTARGET)、および適用する Word テンプレートのパスと名前が指定されます (/WORDTEMPLATE)。最後に、パラメータ /ACTION:WORDOVERVIEWREPORT によりレポートが生成されます。

```
PerfExp.exe /TSD:"c:¥test.tsd" /WORDTARGET:"c:¥report.docx" /  
WORDTEMPLATE:"c:¥template.docx" /ACTION:WORDOVERVIEWREPORT
```

## コマンド ファイル

コマンドラインでコマンドを直接送信するだけでなく、XML 形式のコマンド ファイルを送信することもできます。コマンドは、次の構文を使って指定します。

```
Perfexp.exe /COMMANDFILE:<CmdFile>
```

コマンド ファイルは、バッチ操作用に指定します。

コマンド ファイルとして使用する XML ファイルの構造は、PerfExpCommandFile というルート ノードと、OverviewReport と Command という省略可能な 2 つのサブノードから成ります。

### PerfExpCommandFile

ルート ノード PerfExpCommandFile では、以下を指定できます。

- NoGui="True" / NoGui="False"  
Performance Explorer を表示状態か非表示状態のどちらで起動するかを指定します。
- LogFile="<filename>"  
ログ ファイルの名前と場所を指定します。

ルート ノードの下には、さまざまな Command ノードと OverviewReport ノードを指定できます。

### OverviewReport

OverviewReport ノードでは、Performance Explorer で使用する TSD ファイルを指定します。以下の属性を使用できます。

- File

使用する .tsd ファイルを指定します。省略すると、Performance Explorer で最後にロードしたファイルを使用します。

- Export

概要レポートを保存する出力ファイルを指定します。次の形式を使用できます。

- HTML (.htm)
- Web アーカイブ (.mht)

- Template

オプションで概要レポートに使用するテンプレート ファイル (.ovt) を指定します。

## コマンド

Command ノードのテキストは、コマンドライン パラメータとして解釈されます。ノードは順に実行されます。

次のコードは、XML コマンド ファイルを示しています。

```
<?xml version="1.0" encoding="UTF-8" ?>
<PerfExpCommandFile NoGui="True"
  LogFile="c:¥TEMP¥Results¥PerfExp.log">
  <OverviewReport
    File="c:¥TEMP¥Results¥m@host@Simple.tsd"
    Export="c:¥TEMP¥Results¥Simple.mht"
  />
  <Command>PEW:"c:¥test.pew"</Command>
  <Command>SINGLEINSTANCE</Command>
  <Command>ACTION:STARTALLMONITORS</Command>
  <Command>EXIT</Command>
</PerfExpCommandFile>
```

## ヘルプ コマンド

/? ヘルプ コマンドは次の構文で使用します。

```
Perfexp.exe /?
```

Performance Explorer が開き、コマンドライン ヘルプが表示されます。

## 表示をカスタマイズする

Performance Explorer には、グラフ、グラフの時系列、およびレポートの表示をカスタマイズするためのさまざまなオプションがあります。さまざまな表示設定のデフォルト値を設定したり、個々のグラフ、グラフの時系列やレポートの特定のオプションを設定することができます。

## デフォルト オプション

グラフやグラフの時系列、レポートを表示するためのデフォルト オプションを変更することができます。

### デフォルトのグラフ オプションを設定する

1. Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。
2. **選択された時系列を強調表示する** チェック ボックスをオンにすると、現在選択している時系列を太線で強調して表示できます。



**注:** このオプションは、グラフで作業しているときには便利ですが、グラフをエクスポートまたは印刷する前には、このチェック ボックスをオフにしてください。



- 3. 時系列を重ね合わせる** チェック ボックスをオンにすると、グラフに表示されるすべての時系列が同じ時間に開始すると想定します。  
開始時間が異なるテストの時系列を含むグラフを表示する場合に、このオプションを使用します。
- 4. 背景色** ボックスの一覧から、グラフの効果を高めるために使用する色を選択します。  
カスタム色を使用するには、リスト ボックスの右側の参照 (...) ボタンをクリックします。  
Performance Explorer により、背景イメージとして 1 つ目の色から 2 つ目の色へと縦方向に遷移する色が表示されます。
- 5. テキストおよびグリッドの色** ボックスの一覧から、グラフの軸情報とグリッドに使用する色を選択します。  
カスタム色を使用するには、リスト ボックスの右側の参照 (...) ボタンをクリックします。
- 6. 倍率** 領域で、次の機能を有効化できます。
  - **0 から 100 までの値に拡大縮小する** を有効化すると、実際の時系列データ値ではなく、拡大縮小された値を表示できます。
  - **類似した時系列に同じ倍率を適用する** を有効化すると、同じ種類の時系列に対して同じ倍率を設定できます。
- 7. OK** をクリックします。

## デフォルトの時系列データ オプションを設定する

- Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。
- 時系列** タブをクリックします。
- デフォルトで NULL 値を表示するには、**NULL 値を表示する** チェック ボックスをオンにします。  
**時系列** タブの **表示** グループで **NULL 値** をクリックすると、個々の測定値に対して NULL 値の有効/無効を設定できます。**オプション** ダイアログのチェック ボックスでは、すべての測定値に対する NULL 値の有効/無効が設定されます。
- 要約とカウンタ値の表示間隔** リスト ボックスから、時系列データの表示に使用する時間単位を選択します。
  - **間隔** を選択すると、.tsd ファイルに設定されている計算間隔でデータが表示されます。
  - **秒** を選択すると、データが 1 秒間隔に拡大縮小して表示されます。  
デフォルトの計算間隔を変更するには、**プロファイル - <プロファイル名> - 結果** ダイアログの **時系列** タブを使用します。
- デフォルトの線幅** リストから、新しい測定値をグラフに追加するとき使用する線幅を選択します。
- 各時系列の測定値ごとにグラフ マーカーを表示し、正確な測定値を示すには、**グラフ マーカーを表示する** チェック ボックスをオンにします。
- 数値の桁数** フィールドに、グラフ マーカーに表示する数字の桁数を入力します。  
グラフ マーカーを使用すると、グラフ内の各測定値に対して正確な測定した値を表示できます。
- OK** をクリックします。

## デフォルトのグラフ エクスポート オプションを設定する

- Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。
- HTML ビュー** タブをクリックします。
- グラフ画像の解像度** 領域の、いずれかのオプションをクリックして、グラフを HTML ドキュメントとしてエクスポートするときに生成される画像の解像度を指定します。  
画像ファイルのサイズは、直接解像度に比例します。そのため、画像の解像度が高いほど、解像度の低い画像より Web に転送する時間も長くなります。

4. **画像のタイプ** 領域を使用して、HTML ドキュメントのグラフの画像形式を選択します。

- **JPEG** をクリックすると、HTML ドキュメントのグラフが JPEG ファイルになります。
- **BMP** (Internet Explorer を使用する場合のみ) をクリックすると、HTML ドキュメントのグラフが BMP ファイルになります。

5. **OK** をクリックします。

## デフォルトのローソク グラフ オプションを設定する

1. Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。

2. **ローソク** タブをクリックします。

3. **枠線の色** ボックスの一覧から、現在のグラフのローソクの枠線に使用する色を選択します。  
カスタム色を使用するには、リスト ボックスの右側の参照 (...) ボタンをクリックします。

4. **ローソクの幅** ボックスに、現在のグラフのローソクの幅を入力します。

5. **マーカーの色** ボックスの一覧から、マーカーに使用する色を選択します。

カスタム色を使用するには、リスト ボックスの右側の参照 (...) ボタンをクリックします。

6. **マーカーの高さ** ボックスに、マーカーの高さを指定します。

7. **OK** をクリックします。

## デフォルトの監視オプションを設定する

1. Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。

2. **監視** タブをクリックします。

3. **間隔** フィールドに、グラフを更新する間隔を設定します。



**注:** 監視間隔を、実行中にデフォルトの最小間隔より小さな値に変更すると、間隔は、保存したワークスペースを再度開くときに最小監視間隔にリセットされます。

4. **ページ サイズ** フィールドに、各グラフの水平軸上に表示する、測定値の数を設定します。

5. **履歴サイズ** フィールドに、グラフ全体の水平軸上に保存する、測定値の数を設定します。

6. **背景色** ボックスの一覧から、グラフの効果を高めるために使用する色を選択します。

カスタム色を使用するには、リスト ボックスの右側の参照 (...) ボタンをクリックします。

Performance Explorer により、背景イメージとして 1 つ目の色から 2 つ目の色へと縦方向に遷移する色が表示されます。

7. **テキストおよびグリッドの色** ボックスの一覧から、グラフの軸情報とグリッドに使用する色を選択します。

カスタム色を使用するには、リスト ボックスの右側の参照 (...) ボタンをクリックします。

8. **OK** をクリックします。

## デフォルトの TSD 出力オプションを設定する

1. Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。

2. **TSD 出力プログラム** タブをクリックします。

3. **間隔** フィールドで、時系列データの出力に使用される、時間間隔と時間単位を選択します。

4. **OK** をクリックします。

## デフォルトのレポート作成オプションを設定する


1. Performance Explorer のアプリケーション ボタン (左上) をクリックし、**オプション** をクリックします。

2. **レポート作成** タブをクリックします。
3. テスト結果の検討を開始するときに、概要レポートを自動的に表示するには、**概要レポートを自動的に生成する** チェックボックスをオンにします。  
概要レポートには、アクティブであった仮想ユーザーの数、実行されたトランザクションの数、発生したエラーの数などの、テストの最も重要な結果が含まれます。
4. グラフに NULL 値を表示するには、**グラフに NULL 値を表示する** チェックボックスをオンにします。  
このチェックボックスをオフにした場合、Performance Explorer は、各 NULL 値を、直前の値とその次の値を結ぶ直線に置き換えます。
5. 概要レポートの作成用に保存しておいたテンプレートを使用するには、**概要レポートの新規作成時にテンプレートを使用する** チェックボックスをオンにしてから、参照 (...) をクリックして、テンプレートを選択します。選択されたテンプレートは、すべてのプロジェクトで、新しい概要レポートを作成するのに使用されます。
6. **OK** をクリックします。

## グラフ表示をカスタマイズする

Performance Explorer は、グラフを標準形式で表示しますが、カスタマイズできる部分が多くあります。倍率、グラフの重ね合わせ、グラフの強調表示、グラフのエクスポートなどを行えます。これらの表示オプションは、コンテキストに依存して表示される、**グラフ** タブにあります。グラフを作成または選択したときに、このタブが表示されます。

1. グラフを作成し、ツリーからグラフに測定値をドラッグします。
2. **グラフ** タブの **グラフ** グループには、次のオプションがあります。
  - **倍率**：倍率機能 (スケーリング) を使用すると、グラフを一様に表示することができます。つまり、大きい値を持つ測定値と小さい値を持つ測定値が適切に表示され、有意義なグラフが描画されます。スケーリングを行わないと、小さい値を持つ測定値は平坦なグラフとして表示され、大きい値を持つ測定値は変動の激しいグラフとして表示されてしまいます。個々のグラフに対して特定の倍率を設定するには、**時系列** タブの **時系列** グループで **倍率** リストから倍率を選択します。
  - **グラフの重ね合わせ**：グラフを重ね合わせると、すべてのグラフが実際の開始時間を 無視して共通の開始時間にシフトされます。開始時間が異なるテストの測定値を含むグラフを表示する場合に、このオプションを使用します。
  - **対数 (Y 軸)**：Y 軸を線形軸から対数軸に変更します。
  - **強調表示**：選択したグラフを太線で強調します。

 **ヒント**: グラフの強調表示は、Performance Explorer でグラフを表示して作業しているときには有効です。しかし、グラフをエクスポートしたり印刷する前に、このオプションをオフにするのを忘れないでください。

  - **類似した測定値の拡大縮小**：Performance Explorer は、類似した測定値 (時系列) に同じ倍率を適用します。類似した測定値とは、同じタイプの測定値を言います (測定値リストの **タイプ** 列を参照)。この機能を使用するには、次の手順に従います。あるタイプの測定値を空のグラフにドラッグします。測定値を右クリックして、自動以外の倍率を選択します。**グラフ** タブで **類似した測定値の拡大縮小** をクリックします。同じタイプの測定値をさらにグラフにドラッグします。  
Performance Explorer は、最初の測定値に対して手動で指定した倍率を自動的に適用します。
  - **プロパティ**：プロパティで、グラフの **タイトル** と **説明** を入力できます。これらは、エクスポートしたり印刷したときに、グラフの上部に表示されます。
  - **エクスポート**：作業中のグラフを .csv ファイル、または Excel 文書にエクスポートします。**CSV ファイルとしてエクスポート** をクリックすると、.csv、.htm、.mht、.png などの種々のファイルの種類から選択できます。
3. **表示** グループには、次のオプションがあります。
  - **時系列ポイント**：すべてのグラフの時系列ポイントを表示します。
  - **縮小**：拡大表示をリセットします。
  - **拡大表示**：拡大表示は、デフォルトで有効になっています。X 軸と Y 軸の拡大縮小率を保ったまま、グラフを拡大表示できます。拡大表示を無効化すると、X 軸のみ拡大されます。

- **測定値リスト**：測定値リストの表示/非表示を切り替えます。測定値リストは、グラフの凡例でもあります。ただし、各測定値について、他にも多くの情報が提供されます。

#### 4. 生成 グループには、次のオプションがあります。

- **監視出力**：監視グラフのリアルタイム測定値を使って新しい監視出力を作成します。
- **仮想ユーザー ベースのグラフ**：作業中の時間ベースのグラフと同じ測定値を使用して新しい仮想ユーザー ベースのグラフを作成します。時間ベースのグラフは、測定値を Y 軸に、時間を X 軸に表示します。仮想ユーザー ベースのグラフは、仮想ユーザーの数を X 軸に表示します。

#### 5. データ グループには、次のオプションがあります。

- **監視グラフのクリア**：作業中の監視グラフのすべてのグラフをクリアします。
- **ブラウザで表示**：作業中のグラフを HTML ドキュメントとしてブラウザで表示します。

## グラフの線表示をカスタマイズする

Performance Explorer は、グラフの線を標準形式で表示しますが、カスタマイズできる部分が多くあります。色、線の幅、線のスタイルを変更できます。倍率オプションやマーカーの挿入などを変更できます。これらの表示オプションは、コンテキストに依存して表示される、**時系列** タブにあります。グラフを作成または選択したときに、このタブが表示されます。

#### 1. グラフを作成し、ツリーから測定値をグラフにドラッグし、グラフの線または測定値リストから測定値を選択します。

#### 2. 時系列 タブの **線のスタイル** グループには、次のオプションがあります。

- **スタイル**：さまざまな線のスタイルから選択します。
- **色**：色を選択します。
- **幅**：幅を選択します。

#### 3. 時系列 グループには、次のオプションがあります。

- **倍率**：倍率機能 (スケーリング) を使用すると、グラフを一様に表示することができます。つまり、大きい値を持つ測定値と小さい値を持つ測定値が適切に表示され、有意義なグラフが描画されます。スケーリングを行わないと、小さい値を持つ測定値は平坦なグラフとして表示され、大きい値を持つ測定値は変動の激しいグラフとして表示されてしまいます。グラフのすべての線に対して倍率を有効にするには、**グラフ** タブの **グラフ** グループで **倍率** をクリックします。
- **表示形式**：選択したグラフの表示形式を選択します。線、ローソク、ローソク足、領域 から選択できます。
- **ソース タイプ**：次のソース タイプのいずれかを選択します：平均、カウント、合計、最大、最小、標準偏差。

#### 4. 表示 グループには、次のオプションがあります。

- **マーカー**：マーカーは測定した値をグラフに表示します。
- **NULL 値**：選択したグラフに NULL 値を表示します。
- **測定値の限界**：選択したグラフに境界を表示します。負荷テストを開始する前に、**しきい値の自動生成** ダイアログでこれらの境界を設定できます。
- **パフォーマンス レベルの条件**：選択したグラフにパフォーマンス レベルを表示します。パフォーマンス レベルは、**レポート** タブで定義および割り当てることができます。
- **正規化値**：選択したグラフに正規化値を表示します。正規化した値は、秒ごとの値です。
- **標準偏差領域**：選択したグラフに標準偏差を表示します。
- **表示**：このチェック ボックスをクリックして、選択したグラフの表示/非表示を切り替えます。測定値リストのチェック ボックスを使用して、グラフの表示/非表示を切り替えることもできます。

## 測定値の限界を表示する

負荷テストを開始する前に、下限と上限を **しきい値の自動生成** ダイアログで、またはスクリプト内で直接定義する必要があります。限界値は、概要レポートに表示されます。

測定値の限界をグラフに表示するには：

1. グラフを作成または開きます。
2. 限界を定義した測定値をグラフにドラッグします。
3. 測定値リスト内の測定値をクリックしてアクティブにします。
4. **時系列** タブの **表示** グループで、**測定値の限界** をオンにします。

グラフに赤色と黄色のカラー領域が表示されます。赤色のカラー領域は、上限を超えた領域です。黄色のカラー領域は、上限と下限の間の領域です。



**注：** カラー領域は、測定値が設定されたしきい値を超えた場合のみ表示されます。Performance Explorer には、一度に 1 つのグラフィカルな測定値の限界だけが表示されます。測定値の限界を表示する際は、グラフのスケールが自動的に調整されます。

Performance Explorer には、次の測定値の **平均**、**最大**、および **最小** の各値の限界がグラフィカルに表示されます。

- ページおよびアクション タイマ - アクション時間
- ページおよびアクション タイマ - ページ時間
- タイマ - レスポンス時間
- トランザクション - トランザクション(ビジー) OK

## パフォーマンス レベルの条件を表示する

グラフにパフォーマンス レベルを表示するには、あらかじめパフォーマンス レベルを定義して割り当てておく必要があります。この作業は、負荷テストの実行前（能動的アプローチ）、または実行後（受動的アプローチ）に行うことができます。

パフォーマンス レベルの条件をグラフに表示するには：

1. グラフを作成または開きます。
2. パフォーマンス レベルを割り当てた測定値をグラフにドラッグします。
3. 測定値リスト内の測定値をクリックしてアクティブにします。
4. **時系列** タブの **表示** グループで、**パフォーマンス レベルの条件** をオンにします。

割り当てられたパフォーマンス レベルの条件に対するカラーの境界線と、それを説明する小さな凡例がグラフに表示されます。例：現在のパフォーマンス レベルに  $P90 < 0.2$  のような条件が含まれている場合、グラフの 90% が表示された境界線以下であれば、条件は満たされます。



**注：** Performance Explorer には、一度に 1 つの測定値に対するパフォーマンス レベルの条件だけが表示されます。

# 索引

## A

Apache サーバー 48

## B

BDL 監視プロジェクト

Performance Explorer で 46

新しいプロジェクトを作成する 38

概要 35

基本ユース ケース 38

高度なユース ケース 42

作成 35

思考時間と待機 47

初期化 47

スクリプトのパッケージ化 41

スクリプトを作成する 39, 44

その他のベスト プラクティス 47

チュートリアル 38

プロジェクト属性の計画 38, 42

プロジェクトの使用 42

ループ 47

BEA WebLogic

監視する 66

## C

CPU 使用率

エージェントの健全度 76

CSV

CSV 形式で結果をエクスポートする 16

時系列データへの変換 74

ファイルのエクスポートを自動化する 16

リモート サーバー上でパフォーマンス カウンタ ログ  
を生成する 74

リモート サーバーのパフォーマンス データを取得す  
る 75

リモート サーバーのパフォーマンス ログ データを変  
換する 75

CSV から時系列データ形式へ変換する 75

## G

GUI

Performance Explorer のツアー 8

## H

HTML

レポートをエクスポートする 18

HTML 概要レポート

作成 19

テンプレート 20

テンプレートの作成 19

## I

IBM

DB2

監視する 50

スナップショット監視 50

パフォーマンス指標を取得する 53

WebSphere

パフォーマンス測定値 53

IIOP 67

IIS のパフォーマンス監視

トラブルシューティング 57

利用可能なパフォーマンス測定値 56

## J

JMX

Bean をフィルタリングする 26

監視の前提条件 23

クエリ 26

詳細クエリ 26

接続 24

JMX 管理サーバ 67

## L

Linux コマンド 30

## M

Management Information Base (MIB) 32

MBean を指定する 25

MemFree 30

MIB 32

MIB ファイル 70

Microsoft SQL Server のパフォーマンス監視

利用可能なパフォーマンス測定値 58

## O

OID 31

Oracle

監視する 61, 64

Oracle Forms

監視する 60

クライアント側パフォーマンス測定値 61

Oracle データベース監視

Oracle カウンタの追加 62

Oracle クライアントと Oracle カウンタをインストー  
ルする 62

Oracle のパフォーマンスを監視する (Perfmon) 62

既知の問題 64

パフォーマンス カウンタ 63

## P

PerfMon 68

Performance Data Collection Engine 38

Performance Explorer

概要 4

結果の分析 4

用語 4

リアルタイム監視 4

## R

- REXEC
  - 概要 27
  - 監視する 68
- REXEC クエリ 28

## S

- SAP
  - 監視する 65
- Silk Central
  - 統合 88
- Silverstream 49
- SNMP
  - 監視する 66, 68
  - 検索する 69
  - データソースを追加する 69
- SNMP サポート
  - データソースの定義 32

## U

- UNIX ベースのクエリ 28

## W

- WebLogic (JMX) を監視する 67
- WebLogic (JMX) の監視
  - IIOP 67
  - JMX 管理サーバ 67
- Web ブラウザ
  - 結果を表示する 17
- welcomePE 4
- Windows マシン
  - 監視の要件 22
- Word 概要レポート
  - テンプレート 18
  - 表示 18

## え

- エージェント
  - エージェントの健全度
    - 応答性 76
  - 健全度 76
  - ファイル 5
- エクスポートする
  - グラフ 17
- エラー
  - 分析する 11
- エラー分析グラフ 12

## お

- 応答性
  - エージェントの健全度 76

## か

- 概要レポート
  - HTML 19
  - Word 18

- カスタマイズ
  - グラフ 91, 92
- 監視
  - IBM WebSphere Application Server 54
  - Windows マシン 22
  - インターフェース 68
  - 監視グラフ 72
  - 監視出力 72
  - グラフ 72
  - リアルタイム 22
    - リアルタイム監視
  - 概要 22

- 監視出力
  - 監視を作成する 72
- 監視する
  - デフォルトのオプションを設定する 90
- 監視プロジェクト
  - インフラストラクチャ監視 37

## <

- クエリ
  - JMX 26
  - REXEC 28
  - UNIX 28
  - 実行する 27
- クライアント側測定値
  - 構成する 71
  - 削除する 72
- グラフ
  - Web ブラウザで表示する 17
  - エクスポートする 17
  - エラー分析 12
  - 監視を作成する 72
  - 作成 11
  - デフォルト オプション 88
  - デフォルトのオプションを設定する 88, 89
  - 比較する 13
  - 表示のカスタマイズ 91, 92
  - ローソク オプションを設定する 90

## け

- 結果
  - エクスポートする 16
  - 処理 14
  - 分析 11
- 結果を追加する
  - Silk Central から 10
  - 概要 10
  - 結果フォルダから 10
  - 単一の結果ファイル 10

## こ

- コマンドライン インターフェイス
  - 概要 85
  - コマンド 85
  - コマンド ファイル 87
- 根本原因の分析 12

## さ

サーバー パフォーマンス  
監視する 31

## し

時系列  
データ オプションを設定する 89  
データ出力  
デフォルトのオプションを設定する 90  
ファイルの種類 5  
時系列データ  
CSV からの変換 74  
システムの選択  
ダイアログ ボックス 26  
実行 27

## せ

製品  
起動する 7  
製品を起動する 7  
セキュア シェル 33, 34  
接続が失われた 15  
[接続パラメータ] ダイアログ 31  
接続を失ったコントローラ 15

## そ

属性プール 25  
測定値  
Citrix 84  
CORBA 82  
IIOP 79  
SQL データベース 83  
TUXEDO 83  
Web 78  
Web フォーム 80  
Web ページ 81  
インターネット 77  
概要 75  
カスタム カウンタ 84  
カスタム タイマ 84  
全般 76  
関連させる 12  
データベース 79  
トランザクション 80  
ブラウザ駆動型 81  
ミドルウェア 83  
要約 75  
測定値の限界  
表示 92

## た

ダイアログ ボックス  
システムの選択 26  
単独実行 27

## ち

中間ファイル 5

## つ

ツアー  
ユーザー インターフェイス、Performance Explorer  
8  
追加する  
属性 25

## て

データ ソース  
JMX 24  
PerfMon 68  
UNIX 27  
ウィザード 48  
カスタム 68  
自動検出 70  
スキャンする 70  
定義済み 48  
定義する 48  
データ ソース ツリー メニュー 9  
テクノロジー  
監視 22  
テスト結果  
検討する 9  
テンプレート  
Word 概要レポートの作成 18  
割り当てる 20

## は

パフォーマンス  
監視する 31  
パフォーマンス レベル  
グラフに表示する 93  
パフォーマンス測定値  
WebSphere 53

## ひ

比較レポート  
カスタム 21  
カスタムの作成 22  
種類 20  
地域 21  
地域の作成 21  
ユーザー タイプ 21  
ユーザー タイプの作成 21

## ふ

ファイル  
時系列データへの CSV 変換 74  
負荷テストの結果  
マージ 15  
複数実行 27



## ほ

- ホスト
  - 資格情報の編集 73
  - 複製 73
  - 変更 73
- 保存したクエリ 27

## ま

- マージされたファイル 5

## め

- メモリ使用率
  - エージェントの健全度 76

## ゆ

- ユーザー インターフェイス
  - Performance Explorer の概要 8
- ユーザー ファイル 5

## り

- リアルタイム
  - ファイル 5
- リサンプル

- 結果ファイル 14
- 負荷テストの結果 14
- リモート サーバー
  - パフォーマンス カウンタ ログを生成する 74
  - パフォーマンス データを取得する 75
  - パフォーマンス ログ データを変換する 75
- リモート実行 27

## れ

- レポート
  - Performance Explorer 11
  - Web ブラウザで表示する 17
  - カスタム比較 21, 22
  - コマンドラインでエクスポートする 18
  - 作成 11
  - 種類 18
  - 地域比較 21
  - デフォルト オプション 88
  - デフォルトのオプションを設定する 90
  - テンプレート 20
  - 比較 20
  - ユーザー タイプ比較 21

## わ

- ワークフロー バー
  - Performance Explorer 8