

Silk Performance Explorer 15.0

[ヘルプ](#)

Micro Focus
575 Anton Blvd., Suite 510
Costa Mesa, CA 92626

Copyright © Micro Focus 2014. All rights reserved. SilkPerformer は Borland Software Corporation に由来する成果物を含んでいます, Copyright © 2014 Borland Software Corporation (a Micro Focus company).

MICRO FOCUS, Micro Focus ロゴ、及びその他は Micro Focus IP Development Limited またはその米国、英国、その他の国に存在する子会社・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。

2013-12-16

目次

Performance Explorer 15.0	5
入門	5
Performance Explorer	5
UI のツアー	6
リアルタイム監視	8
結果の分析	8
Silk Central	9
アプリケーションの外見を定義する	9
データソース	9
定義済みのデータソース	10
データソースの自動検出	15
カスタムデータソース	17
CSV ファイルから TSD ファイルへのデータ変換	23
JMX 監視	23
Easy JMX クエリ	25
サーバーの監視	27
自動監視	27
Silk Central パフォーマンスを監視する	27
IBM DB2 のパフォーマンス監視	28
IBM WebSphere Application Server の監視	32
Internet Information Server (IIS) のパフォーマンス監視	33
Microsoft SQL Server のパフォーマンス監視	36
Oracle Forms のパフォーマンス監視	38
Oracle のパフォーマンスを監視する	40
SAP のパフォーマンス監視	43
Secure Shell (セキュア シェル) を介したパフォーマンス監視	44
SNMP サポート	45
WebLogic (SNMP) の監視	47
WebLogic (JMX) を監視する	49
リモートマシンの監視	50
リモートマシン監視の要件	50
Windows NT でリモート監視を有効にする	50
Windows XP でリモート監視を有効にする	51
Windows コンピュータで Diskperf を実行する	51
リモートコンピュータで Diskperf を実行する	52
Perflib サブキーに対する読み取りアクセス権の付与	52
リモートマシンのトラブルシューティング	53
BDL 監視プロジェクトを生成する	54
Silk Performer BDL 監視プロジェクト	54
BDL 監視プロジェクトを作成する	55
チュートリアル	57
ベストプラクティス	67
データの収集	68
監視オプションを設定する	68
クライアント側測定値	69
監視タイプ	70
ホストデータ	71
測定値	72
デフォルト オプション	80
グラフ表示をカスタマイズする	83

結果の分析	84
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	85
結果の表示	85
概要レポート	87
時系列データ (TSD) 結果ファイルを指定する	91
特定の結果	92
カスタム結果	92
複数の負荷テストから得られた結果の表示	93
地域比較レポート	94
負荷テスト結果とサーバー監視データの表示	95
Web ブラウザでの結果の表示	96
時系列設定をカスタマイズする	97
根本原因の分析	98
ファイルのマージ	99
CSV ファイルのエクスポート	102
グラフを比較する	104
エラー分析	104
Silk Central と Performance Explorer のインテグレーション	105
Silk Central から結果を追加する	105
オンデマンドで利用できるダウンロード	106
相互負荷テストレポート	106
コマンドライン パラメータ	106
コマンドラインから Performance Explorer を起動する	106
使用できるコマンド	107
コマンド ファイル	108

Performance Explorer 15.0

Performance Explorer 15.0 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、過去の負荷テストの結果を分析できます。多様なグラフィカル ツールにより、テスト中にターゲット サーバーのリアルタイム監視を行うことと、テスト完了後に結果を徹底的に分析することの両方が可能です。テスト結果の検討は、ドラッグ機能を備えたクリック方式のユーザー インターフェイスを使って、ワークフロー バーから簡単に行えます。

入門

このセクションでは、Silk Performance Explorer を使用してサーバーを監視したりテスト結果を分析するのに必要な情報が記載されています。

Performance Explorer

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、過去の負荷テストの結果を分析できます。多様なグラフィカル ツールにより、テスト中にターゲット サーバーのリアルタイム監視を行うことと、テスト完了後に結果を徹底的に分析することの両方が可能です。テスト結果の検討は、ドラッグ機能を備えたクリック方式のユーザー インターフェイスを使って、ワークフロー バーから簡単に行えます。

Performance Explorer の概要

Performance Explorer リアルタイム監視では、実況グラフにより、最も関連のあるターゲット サーバーのパフォーマンス情報をカスタマイズして表示することができます。Web サーバー、アプリケーションサーバー、およびデータベース サーバーを幅広く監視できます。同時に複数のグラフを開くことができるので、1 つの画面で、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の両方を同時に監視できます。ドラッグ機能を備えたメニュー ツリー エディタにより、任意のデータソースから得られる要素をグラフ上で組み合わせることができます。

テスト後、ターゲット サーバーのパフォーマンスは、クライアント側、サーバー側両方からグラフにすることができます。レスポンス時間の測定値は、クライアント側の視点を示し、スループット データは、サーバー側の視点を示します。グラフは自由にカスタマイズでき、テスト中に採取した測定値を、いくつでも、必要に応じて含めることができます。1 つ以上のテストから得た情報を使用して、複数のグラフを同時に開くことができるので、対比/比較の操作がしやすくなります。Web、データベース、I/O など、最も典型的なテストのシナリオにあわせたテンプレートが用意されています。これらの既定のグラフに、必要なデータを流し込むこともすぐ簡単にできます。ドラッグ機能により、グラフ要素を任意のデータソースから得られる要素と組み合わせることができます。クライアント レスポンス時間とサーバー 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。

メニュー ツリー エディタの高度なドラッグ機能を使用すると、任意のデータ ソースから得られる情報要素を組み合わせ、選択した任意の数のグラフに表現することができます。さまざまなソースから得られる測定値の追加や削除を行って、ニーズに合った恒久的または一時的なグラフやレポートを作成することもできます。変更した設定をプロジェクトごとに保存できるので、Performance Explorer を常に好みのビューで開くことができます。

Performance Explorer には、ベースライン レポート ファイル (baseline.brp) から得たユーザー グループ データと、時系列テスト結果情報を組み合わせた包括的な XML/XSL ベースの概要レポートも用意されています。この概要レポートは、用意されているテキストを編集したり、関心のあるグラフを挿入したり

することにより、ニーズに合わせてカスタマイズできます。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 の起動

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。

以下の手順の 1 つを実行します。

- **スタート > プログラム > Silk > Silk Performer <バージョン> > 分析ツール > Silk Performance Explorer** を選択します。
- Silk Performer から、**結果 > サーバーの監視** を選択します。
- Silk Performer から、**結果 > 時系列の検討** を選択します。

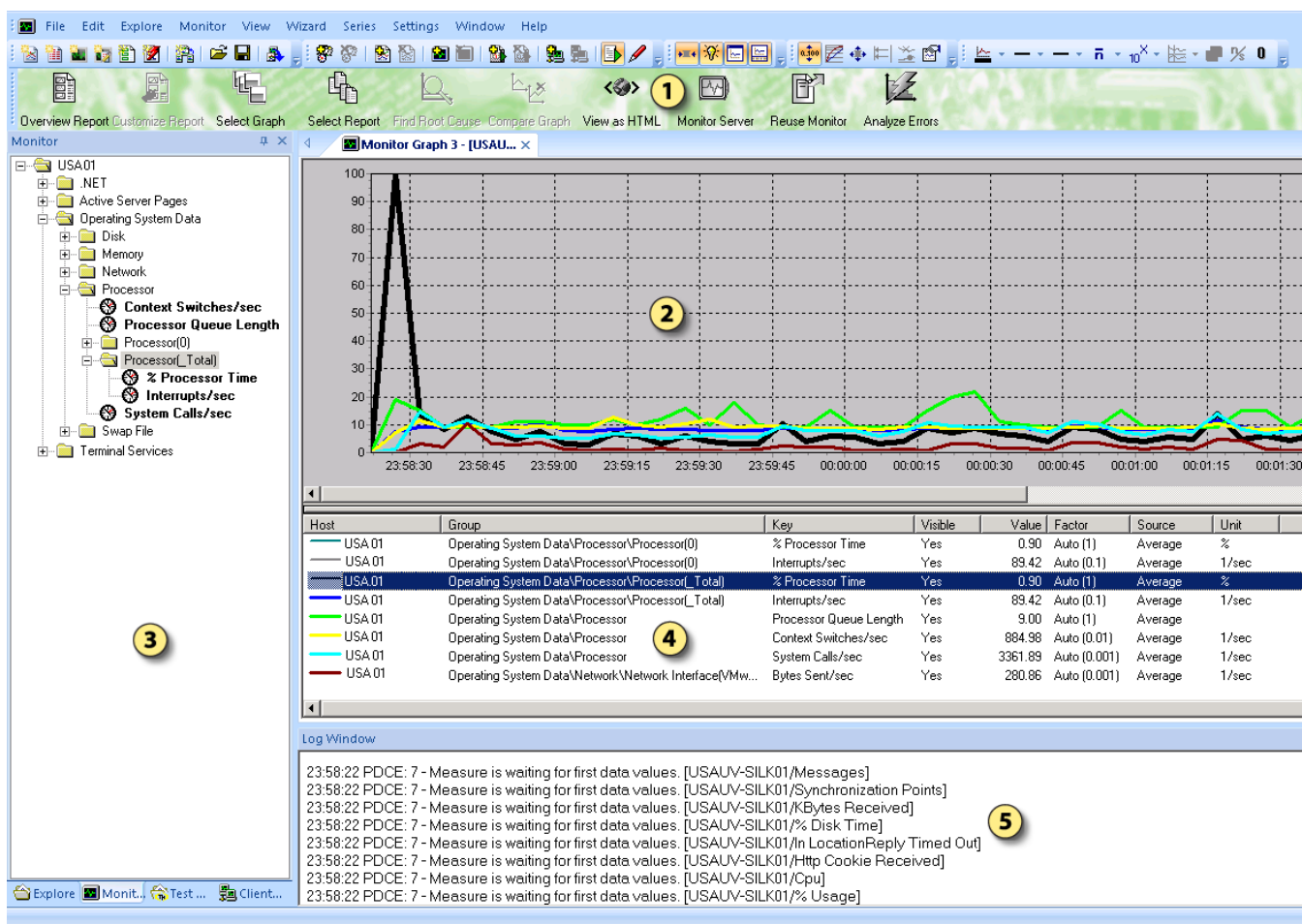


注: コマンドライン インターフェイスから Performance Explorer を起動することもできます。

新しいテンプレートを編集するか、Performance Explorer 結果メニューから Silk Performer 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

以下では、Performance Explorer のインターフェイスの主要セクションを取り上げて説明します。



- ① ワークフロー バー
- ② グラフ/レポートビュー
- ③ ツリーメニュー
- ④ 監視リスト
- ⑤ ログウィンドウ

ワークフロー バー

ワークフロー バーは、サーバー パフォーマンスの監視に関する主要な操作を行いやすくするためのものです。ワークフロー バーのボタンを使用すると、次のコマンドを実行できます。

- **概要レポート**：概要レポートを生成します。このレポートには、一般情報、ユーザー タイプ、グラフ、データ テーブルなど、負荷テストの重要な結果がすべて含まれます。
- **レポートのカスタマイズ**：概要レポート テンプレートに表示されるエントリをカスタマイズできます。
- **グラフの選択**：時系列データを表示するためのグラフ テンプレートを選択できます (TSD ファイル)。
- **レポートの選択**：時系列データを表示するためのレポート テンプレートを選択できます (TSD ファイル)。
- **根本原因の特定**：クライアント側のパフォーマンスの問題を、サーバー側の測定値と関連させることができます。
- **グラフの比較**：複数の負荷テスト間で同じ測定値タイプを調べることができます。
- **HTML として表示**：Web ブラウザで表示するために、レポート出力を HTML で生成します。
- **サーバーの監視**：分析のためのデータ ソースを指定したり、使用可能なデータ ソースを自動検出することができます。
- **監視の再利用**：Silk Performer を使用してパフォーマンス モニターを編集したり、Silk Central で使用するためにモニターをエクスポートしたりすることができます。

- **エラーの分析**：負荷テストのすべてのエラーがグラフィックで表示され、詳細な情報まで掘り下げることが可能です。

グラフ/レポート ビュー

ここでは、サーバー監視データと負荷テスト結果が、グラフおよびレポート内に表示されます。

Performance Explorer は、グラフおよびレポート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いです。測定値の表示に使用される色、線の幅、および線のスタイルを変更できます。倍率オプションを変更したり、マーカーの挿入または削除もできます。

ツリー メニュー

ツリーメニューには、複数のビューがあります。ビュータブを使用して、分析するテスト結果の参照、データソースの追加、Silk Central からのテスト結果のダウンロード、使用可能なパフォーマンス モニタの参照を行うことができます。使用可能なモニタは、幅広いサーバー パフォーマンス指標とオペレーティングシステム データをカバーしています。

モニタ リスト

このビューには、アクティブなリアルタイム パフォーマンス モニタとオフライン時系列データ (TSD) のパフォーマンス測定値がすべて一覧表示されます。モニタおよび測定値ごとの主要プロパティと統計が、表形式で表示されます。

ログ ウィンドウ

このビューには、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。

リアルタイム監視

Performance Explorer は、サーバー側とクライアント側のリアルタイム監視をサポートします。

サーバーの監視

Performance Explorer をリアルタイム監視に使用する場合は、1 つ以上の実況グラフが自動的に表示され、関連するターゲットサーバー パフォーマンス統計のうち、デフォルト選択のデータを表示します。

Performance Explorer には、一般に使用されているオペレーティングシステム上で動作する 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、クライアントなど広範囲にわたる自動監視機能が組み込まれています。

クライアント側の監視

リアルタイムのクライアント側測定値も、他の監視測定値と同様に使用します (たとえば、組み込みグラフや監視レポートなど)。



注: インスタンスによっては、負荷テストやクライアント側のリアルタイム測定値の監視の最中に、仮想ユーザーがエラーになったり、起動しなかったりする場合があります。「SYSTEM:231 All pipe instances are busy」(パイプ インスタンスがすべてビジーです) というエラーが表示されます。このエラーの原因は、リアルタイム監視を妨げるリアルタイム ウィルス防御ソフトです。名前付きパイプはファイルとして認識されるため、ウィルス防御ソフトのスキャン対象となります。

結果の分析

監視レポートを作成して、テストデータを分析します。Performance Explorer は、1 つ以上の時系列データ (TSD) ファイルに書き込んで保存することができます。TSD ファイルを使用すれば、テスト中のサーバーのパフォーマンス測定値をクライアントとサーバーの両方の観点から表示するグラフをいつでも生

成することができます。レスポンス時間の測定値は、クライアント側から見たパフォーマンスを明らかにするのに対して、スループットデータは、サーバー側から見たパフォーマンスを明らかにします。

監視レポートは、保存してプロジェクトに追加したり、後で使用できるようアーカイブしておくこともできます。

また、結果の検討中にグラフをカスタマイズすることもできます。グラフには、テスト測定値をいくつでも、必要に応じて表示することができます。複数のグラフを開き、記録した任意のテストから測定値を追加できます。グラフと測定値を使って、さまざまなテストの結果を選択してビジュアルに比較したり、テスト結果とリアルタイム監視データを比較したりできます。

包括的な HTML ベースの概要レポートでは、ベースライン レポート ファイル (baseline.brp) から得たユーザー グループ データと、時系列のテスト結果情報を組み合わせます。デフォルト テキストを編集し、関心のあるグラフを挿入することにより、概要レポートはニーズに合わせてカスタマイズできます。変更内容をテンプレートとして保存すれば、将来レポートを作成する際にカスタム設定を再利用できます。

Silk Central

Performance Explorer は、Silk Central と完全に統合されています。Silk Performer を使って実行した Silk Central のテストの結果は、Performance Explorer にダウンロードして分析できます。

アプリケーションの外見を定義する

各種のカラー スキームに基づいて全体的な UI の外見を変更できます。

1. **表示 > アプリケーションの外見** を選択します。
2. 次のとおり、対象のアプリケーション スタイルを選択します。
 - 青スタイル
 - 黒スタイル
 - 銀色スタイル
 - アクア スタイル

UI は選択したスタイルに基づいて描画しなおされます。

データ ソース

このセクションでは、Performance Explorer 用のデータ ソースを定義する方法について説明します。

サーバーやアプリケーションをパフォーマンス データを収集するためのデータ ソースとして定義します。データ ソースの定義方法は以下のいずれかです。

- 定義済みデータ ソースのセットからサーバーやアプリケーションを選択する。


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。

どちらの方法でも、**データ ソース ウィザード** を開くことが必要です。それには、**監視** メニューを使用するか **監視** ペインを右クリックします。


定義済みのデータ ソース


このページでは、Performance Explorer に使用できる定義済みのデータ ソースの一覧を示し、Performance Explorer が監視できるサーバー タイプごとに固有のインストールと構成要件の概要を説明します。

システム	インストールと構成の要件
.NET	PerfMon を使用した .NET CLR と ASP.NET
Active Server Pages	Performance Explorer では、カスタム構成なしで Active Server Pages を監視できません。表示できるカウンタは、インストールされている IIS のバージョンによって変わります。 <ul style="list-style-type: none">• IIS 5.1/6.0 (PerfMon 使用)
Apache	<p>(rexec を使用して) Apache Web Server の監視がサポートされるのは、HP-UX、Linux、Solaris、OSF1 の各 OS です。</p> <p>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。それには、mod_status モジュールを組み込む必要があります (デフォルトで実行)。さらに詳しい情報を取得するには、コンパイル オプション ExtendedStatus On を選択します。ステータス レポートを有効にするブラウザを foo.com ドメインからに限定するには、構成ファイル access.conf にコードを追加します。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、通常、/usr/local/apache/etc、または /usr/local/apache/conf (Solaris)、/etc/httpd/conf (Linux) にあります。デフォルトでは、次のコードは、構成ファイルの 1 つの中でコメントアウトされています。</p> <pre><Location /server-status> SetHandler server-status order deny, allow deny from all allow from.foo.com </Location></pre> <p>コメントアウトしない場合は、それを Apache のメイン構成ファイル httpd.conf に追加します。Allow 文が正しいことを確認します。サーバーを再起動します。サーバーのパフォーマンス モジュールを確認するには、ブラウザに次の URL を入力します。</p> <pre>http://hostname/server-status?auto</pre> <p>次のメッセージと同様のレスポンスが返れば、監視機能は有効です。</p> <pre>Total Accesses: 210 Total kBytes: 94 CPULoad: .000278279 Uptime: 366539 ReqPerSec: .000572927 BytesPerSec: .262608 BytesPerReq: 458.362 BusyServers: 1 IdleServers: 6 Scoreboard: _____W.....</pre> <p> 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 コマンドではなく、次の例に示すような WGet コマンドを使用します。</p>

システム	インストールと構成の要件
	<pre>wget -q -O - http://hostname/server-status grep "requests/sec" cut -c5- cut -d¥ -f1</pre>
アプリケーション In-Depth	<p>AppDynamics Diagnostics : 組み込み</p> <p>dynaTrace Diagnostics: 組み込み</p>
ASP .NET	PerfMon を使用
BEA WebLogic 8.x、9.x、10.x、11.x	<p>Performance Explorer では、WebLogic 8.x サーバー、9.x サーバー、10.x サーバー、および 11.x サーバーを監視できます。BEA WebLogic 8.x および 9.0 から 9.1 を監視するには、JNDI プロトコルを使用して、Management EJB (MEJB) 経由で MBean Server に接続します。バージョン 9.2 以上では、IIOP を通して JSR-160 が使用されます。デフォルトの接続パラメータは、Performance Explorer であらかじめ構成されています。</p> <p>WebLogic Server 11.x のサポートには、次のものが含まれます。</p> <ul style="list-style-type: none"> • WebLogic Server 11g (10.3.1) • WebLogic Server 11gR1 PS1 (10.3.2) • WebLogic Server 11gR1 PS2 (10.3.3) • WebLogic Server 11gR1 PS3 (10.3.4) <p>正しいクラスパスを構築するため、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 (その場所へのアクセス権) を Performance Explorer に設定する必要があります。</p> <p>JMX ブラウザから、監視対象の MBean 属性を選択できます。簡易 JMX プロファイルを使用して、対象となる MBean を事前に選択できます。</p> <p>SNMP も使用します。</p>
Borland Application Server 6.6	<p>Performance Explorer では、IIOP 上の Corbaloc プロトコルを使用して、JSR-160 経由で Borland Application Server 6.6 を監視できます。アプリケーション サーバーは、IIOP 経由で MBean Server にアクセスできるように構成する必要があります。</p> <p>正しいクラスパスを構築するため、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 (その場所へのアクセス権) を Performance Explorer に設定する必要があります。</p> <p>JMX ブラウザから、監視対象の MBean 属性を選択できます。簡易 JMX プロファイルを使用して、対象となる MBean を事前に選択できます。</p> <p>一部の MBean では、Borland Application Server にパッチ 7697 を適用する必要があります。</p>
カスタム データ	<ul style="list-style-type: none"> • JMX データ • NT パフォーマンス モニタ データ • Rexec データ • SNMP データ
F5 Labs 3DNS	SNMP を使用
Helix Server	PerMon を使用
IBM Universal Database DB2	スナップショットを使用
IBM UniversalDatabase DB2 7.1	PerfMon を使用

システム	インストールと構成の要件
IBM WebSphere Application Server 6.1	<p>Performance Explorer では、アプリケーション サーバーのカスタム構成なしで、WebSphere Server を監視できます。</p> <p>JNDI プロトコルを使用して、Management EJB (MEJB) 経由で MBean Server に接続します。デフォルトの接続パラメータは、Performance Explorer であらかじめ構成されています。</p> <p>正しいクラスパスを構築するため、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 (その場所へのアクセス権) を Performance Explorer に設定する必要があります。Performance Explorer では、監視を行うために、同じバージョンの IBM JVM を使用する必要があります。</p> <p>JMX ブラウザから、監視対象の MBean 属性を選択できます。</p> <p>WebSphere の JMX 監視を行うには、IBM JDK を指定する必要があります。</p>
IBM WebSphere Application Server 7.0、8.0、8.5	<p>Performance Explorer では、アプリケーション サーバーのカスタム構成なしで、WebSphere Server を監視できます。これは、管理セキュリティとインバウンド CSIv2 トランスポート層が有効となった WebSphere Server 7.0 には適用されません。その場合は、SSL が必要です。詳細については下記の表を参照してください。</p> <p>JSR-160/Plain IIOP プロトコルは、Management EJB (MEJB) 経由で接続 MBean Server に接続するために使用されます。デフォルトの接続パラメータは、Performance Explorer であらかじめ構成されています。</p> <p>正しいクラスパスを構築するため、または追加の JVM パラメータを正しく設定するため、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 (その場所へのアクセス権) を Performance Explorer に設定する必要があります。</p> <p>WebSphere の JMX 監視を行うには、同じバージョンの 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。</p> <p>JMX ブラウザから、監視する MBean 属性を選択できます。</p>
Internet Information Server 5.1、6.0、7.0、7.5	<p>Performance Explorer では、カスタム構成なしで IIS Web サーバーを監視できます。</p> <p>SNMP は、カスタム監視構成を使用する場合に限りサポートされます。IIS をインストールする前に、Windows NT SNMP サービスをインストールする必要があります。</p>
JBoss 4、5、6、7	<p>Performance Explorer では、アプリケーション サーバーのカスタム構成なしで、JBoss 4、5、6、7 を監視できます。JNDI プロトコルを使用して、Management EJB (MEJB) 経由で MBean Server に接続します。デフォルトの接続パラメータは、Performance Explorer であらかじめ構成されています。正しいクラスパスを構築するため、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 (その場所へのアクセス権) を Performance Explorer に設定する必要があります。JMX ブラウザから、監視対象の MBean 属性を選択できます。2 つの簡易 JMX プロファイルを使用して、対象となる MBeans を事前に選択できます。</p> <p>JBoss 5、6、または 7 サーバーにリモートからアクセスするには、JBoss がデフォルトの場所 [127.0.0.1] ではなく、サーバーの外部 IP アドレスを指している必要があります。JBoss が外部 IP アドレスを指すようにするには、以下の処理を実行します。</p> <p>JBoss 5 JBoss アプリケーション サーバーの run.conf.bat に以下の設定を追加します。JAVA_OPTS=%JAVA_OPTS% -Djboss.bind.address=<使用するサーバーの外部 IP></p> <p>JBoss 6 コマンド run.bat -b 0.0.0.0 を使用して、コマンド ラインから JBoss アプリケーション サーバーを開始します。</p>

システム	インストールと構成の要件
<p>Micro Focus Enterprise Server</p>	<p>JBoss 7 コマンド Standalone.bat -b 0.0.0.0 -Djboss.bind.address.management=<使用するサーバーの外部 IP> を使用して、コマンドラインから JBoss アプリケーション サーバーを開始します。</p> <p>Performance Explorer では、アプリケーション サーバーのカスタム構成なしで、Micro Focus Enterprise Server を監視できます。</p> <p> 注: 監視は、Windows プラットフォームでのみサポートされます。</p> <p>Micro Focus Enterprise Server の監視プロジェクトをセットアップするには、データソース ウィザード から Micro Focus > Micro Focus Enterprise Server > System (perfmon) を選択します。</p> <p>接続パラメータ ダイアログで 属性設定 をクリックして、属性設定 ダイアログを開きます。</p> <ul style="list-style-type: none"> • #Perfmon.User - Value フィールドに、監視するサーバーの Windows ユーザー名を入力します。 • #Perfmon.Password - Value フィールドに、対応するパスワードを入力します。 • #Perfmon.Domain - Value フィールドに、監視するサーバーのドメインを入力します。 • #Region Name - Value フィールドに、領域名またはサーバー名を入力します。
<p>Microsoft Proxy Server 2.0</p>	<p>Microsoft Proxy Server は、パフォーマンス モニタにより監視されます。</p> <p>特別な構成は必要ありません。</p>
<p>オペレーティング システム データ</p>	<p>オペレーティング システム データは、次のプラットフォームで監視できます。</p> <ul style="list-style-type: none"> • Windows <ul style="list-style-type: none"> • Windows XP/2003 • Windows 7/Vista/2008 • Linux <ul style="list-style-type: none"> • システム (rexec:vmstat) • システム (rstat) • システム (ucd snmp) • Solaris <ul style="list-style-type: none"> • システム (rexec:mpstat) • システム (rexec:vmstat/iostat/df) • システム (rstat) • HP-UX <ul style="list-style-type: none"> • システム (rexec:vmstat/iostat) • システム (rstat) • AIX (rstat を使用) • Unix (rstat を使用) • OSF1 <ul style="list-style-type: none"> • システム (rexec:vmstat/iostat) • システム (rstat) <p>ホストで使用可能な場合、プロセッサ利用状況、ディスク使用率、スワップ ファイル利用状況、およびネットワーク パケットなどの情報を監視できます。</p>

システム	インストールと構成の要件
	<p>Microsoft システムでは、オペレーティング システム データはパフォーマンス モニタにより監視されます。UNIX システムでは、Remote Execution Daemon を実行する必要があります。このようなデーモンは、UNIX システムではデフォルトで実行されます。</p> <p> 注:</p> <p>Rexec:vmstat を使用して値を取得するとき、Performance Explorer により他の測定値に反映される場合があります。これは、特定のサーバーの VMSTAT が想定しない列に値を返すことにより起きることがあります。</p> <p>Performance Explorer が VMSTAT から値を受信することを想定している列は、realtime.ini (C:\Program Files\Silk\Silk Performer15.0\Include) で定義されます。必要な測定値を正確に取得するために、このファイルにある列の値を編集できます。</p> <p>たとえば、サーバーが System\CPU Usage System の値を列 13 に返し、一方 Operating System Data\Linux\System (rexec:vmstat) では列 14 を System\CPU Usage System に使用します。この動作を訂正するには、realtime.ini を編集する必要があります。</p>
Oracle	SNMP と v\$sysstat を使用
Oracle 9、10、11	PerfMon を使用
	<p>Performance Explorer は、カスタム構成を使用しないで Oracle 9、10、および 11 サーバーを監視できます。ただし、Windows パフォーマンス モニタ用の Oracle カウンタは、デフォルトの Oracle インストール オプションではインストールされません。カスタムのインストール オプションで、使用可能にする必要があります。Oracle サーバーの監視に関する詳細情報は、Performance Explorer ヘルプの「Oracle のパフォーマンスを監視する」のセクションを参照してください。</p>
Oracle Application Server 10.1.3	<p>Performance Explorer では、アプリケーション サーバーのカスタム構成なしで、Oracle Application Server 10.1.3 を監視できます。</p>
	<p>JNDI プロトコルを使用して、Management EJB (MEJB) 経由で MBean Server に接続します。デフォルトの接続パラメータは、Performance Explorer であらかじめ構成されています。</p>
	<p>正しいクラスパスを構築するため、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 (その場所へのアクセス権) を Performance Explorer に設定する必要があります。</p>
	<p>JMX ブラウザから、監視対象の MBean 属性を選択できます。簡易 JMX プロファイルを使用して、対象となる MBean を事前に選択できます。</p>
Oracle Forms Application Server	
PeopleSoft	<ul style="list-style-type: none"> • アプリケーション サーバー (BEA Tuxedo) • データベース システム <ul style="list-style-type: none"> • IBM Universal Database DB2 (スナップショットを使用) • Oracle (v\$sysstat を使用) • オペレーティング システム <ul style="list-style-type: none"> • AIX (ssh を使用) • HPUX (ssh を使用) • Solaris (ssh を使用) • プロセス スケジューラ

システム	インストールと構成の要件
	<ul style="list-style-type: none"> • Oracle • SQL Server 2000 • Web サーバー • Apache • BEA WebLogic (SNMP を使用)
SAP と SAP V7	SAPGUI Monitoring、SAPGUI OS-Monitoring
Secure Shell	
Silk Central 2008、2008 R2、2009、2009 R2、2010、2010 R2、2011	アプリケーション サーバーとフロントエンド サーバー
Silk Performer 15.0	エージェントとコントローラ
SQL Server 7.0、2000、2005、2008、2008 R2	PerfMon を使用 SQL Server の最上位バージョンは、パフォーマンス モニタを使って監視できます。特別な構成は必要ありませんが、パフォーマンス モニタのすべてのカウンタが SQL Server が動作しているサーバーで有効であることを確認します。
SUN JVM 1.5、1.6、および 1.7	EJMX と JMX MBeanServer を使用 Performance Explorer では、RMI プロトコルを使用して、JSR-160 経由で SUN JVM 1.5、1.6、および 1.7 を監視できます。プラットフォーム MBean Server を使用するアプリケーションまたはアプリケーション サーバーはすべて、この方法で監視できます。カスタム構成は必要ありません。 JMX ブラウザから、監視対象の MBean 属性を選択できます。簡易 JMX プロファイルを使用して、対象となる MBean を事前に選択できます。
ターミナル サービス	PerfMon を使用
あらかじめ設定されている VMware ESX Server	SNMP を使用
Windows Media サービス	PerfmMon を使用

セキュリティ設定 WebSphere のバージョン	無効な管理セキュリティ	有効な管理セキュリティ インバウンド CSIv2 トランスポート層 = TCP/IP	有効な管理セキュリティ インバウンド CSIv2 トランスポート層 = SSL サポート	有効な管理セキュリティ インバウンド CSIv2 トランスポート層 = SSL 必須
7.0.0.27	+	+	+	- *
8.0.0.5	+	+	+	+
8.5.0.1	+	+	+	+

*) 正常に接続しましたが、一部の MBean にアクセスできません。

データ ソースの自動検出

データ ソースの検出では、お使いのシステムに問い合わせ使用可能なデータ ソースを調べ、Performance Explorer を使って監視できるものを提案します。検出の目的は、考えられるすべてのデー

タソースを探すことではありません。自動検出は、定義済みのデータソースによく似た構成に基づいて行われます。定義済みのカウンタを調べて、それが使用できるかどうかを確認します。たとえば、データソース検出では、システムを調べて、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。

その結果、監視可能なさまざまなデータソースの一覧が得られるため、システムを手動で調べてパフォーマンスデータソースを探す必要はありません。

データソース スキャナの目的は、監視対象となるシステムのパフォーマンスデータソースに高いレベルでアクセスできるようにすることです。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特性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

使用可能なデータソースがあるかシステムに問い合わせると、手動でパフォーマンスデータソースを探してシステムを参照する必要がなくなります。

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. **データソースを検出させる** 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。 **接続パラメータ** ページが開きます。
3. **ホスト名** テキスト ボックスで、調査す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。


データソースの検出は、特定の構成に基づいて行われます。さまざまなプラットフォーム上で次のデータソースを確認します。

- .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

データ ソース スキャナがシステムを調査する一方で、検出がどの程度進捗しているかのフィードバックを返します。上の進捗バーが全体的な進捗を示します。下の進捗バーは、どのプロトコルが現在システムの調査に使用されているかを示します。

データ ソース スキャナは時折ダイアログ ボックスを表示し、ユーザー ID とパスワードをリクエストしてきます。これらのダイアログ ボックスは、たとえば、特定のホストでリモート実行デーモンが見つかった場合に表示されます。

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ともチェックします。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。 **表示する測定値の選択** ページが開きます。

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。

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 WinNT/Windows 2000 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使用する 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. **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. **Custom Data** フォルダを展開して、**NT Performance Monitor Data** 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** 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もあります。

7. 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。

8. **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。

9. **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。

10 **次へ** をクリックします。 **NT パフォーマンス カウンタの参照** ダイアログ ボックスが開きます。

11 省略可能: 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。

- a) **Performance object** リスト ボックスから、監視するオブジェクトを選択します。
デフォルトでは、**Processor** が選択されます。
- b) **Select counters from list** オプション ボタンをクリックして、カウンタ タイプを選択します。
デフォルトでは、**% Processor Time** が選択されます。
- c) **Select instances from list** オプション ボタンをクリックして、インスタンスの数を選択します。
デフォルトでは、**_Total** が選択されます。
- d) **Add** をクリックして、選択したカウンタを測定値に追加します。 **カウンタ使用** ページが開きます。
- e) **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。
- f) **次へ** をクリックして、**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。

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. **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. **Custom Data** フォルダを展開して、 **Rexec Data** 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** ボックスで、監視対象となるマシンを指定します。
7. 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。

8. [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。
9. **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。
10. 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SEMB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 rexec コマンド (ps -ef egrep -c ".*" など) を指定します。
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。

オプション 説明

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 XXXXXXXXXXX XXXXXXXXXXXX
          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。

11追加 をクリックします。

12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。

13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プレースホルダです。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。

```
XXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX
      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
	8					
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	cat /proc/meminfo
単独実行	オン
正規表現	^MemFree:.*\$
フィールド インデックス	2
区切り文字	空白を区切り文字とする をオンにします。
スキップする行数	0

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 バ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

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。

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. **Custom Data** フォルダを展開して、**Snmp Data** 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、SNMP エージェントを実行しているホストを指定します。
7. 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。

8. 適切な [ポート]、[コミュニティ]、および [バージョン] を指定します。
9. **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。
- 10 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。
 - a) **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。
 - b) 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。
 - c) **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。
- 11 **追加** をクリックして、SNMP 測定値を追加します。
- 12 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。
- 13 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. **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. **Custom Data** フォルダを展開して、**Snmp Data** 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、SNMP エージェントを実行しているホストを指定します。
7. 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。

8. 適切な [ポート]、[コミュニティ]、および [バージョン] を指定します。
9. **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。
- 10 **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 の新規コンパイル...** を選択します。 **MIB の新規コンパイル** ダイアログ ボックスが開きます。
- 11 コンパイルする MIB ファイルの名前を **MIB ソース ファイル** テキスト ボックスに入力します。
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、Silk Performer インストール ディレクトリの ...¥INCLUDE ¥VENDORMIBS ディレクトリにあります。
- 12 **コンパイル** 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 ボックスに追加されます。

CSV ファイルから TSD ファイルへのデータ変換

Silk Performer には、CSV (コンマ区切り値) ファイルを TSD (時系列データ) ファイルに変換できるコマンドライン ツール (csv2tsd.exe) が用意されています。

TSD 形式への変換は、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 形式で保存した場合は、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。

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. **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. 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 アプリケーション 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 アプリケーション 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。

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 ボックスで、JMX サーバーを指定します。
7. 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。
8. [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。
9. 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。
- 10 **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。
- 11 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。
- 12 **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。
- 13 **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。
- 14 **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。
- 15 **OK** をクリックします。 **接続パラメータ** ページが開きます。
- 16 **次へ** をクリックしてから、**終了** をクリックします。 Performance Explorer が JMX サーバーに接続されます。
- 17 **JMX Data Source Browser** に表示された Bean を調べます。
- 18 **終了** 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 Performer15.0\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。

- [JMX Data Source Browser] を開きます。
- 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。
- 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。
- 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。
- 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。
- 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。
- OK** をクリックします。
- Query** メニュー ツリーのルート要素をクリックします。
- Advanced Query** タブの下のペインに、クエリの説明を入力します。
- 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。
- 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 .jqd です。




注: C:¥Program Files¥Silk¥Silk Performer15.0¥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。

- [JMX Data Source Browser] を開きます。
- Advanced Query** タブをクリックします。
- 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 Performer15.0\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。

サーバーの監視

このセクションでは、負荷テスト時にサーバーを監視するように Performance Explorer をセットアップする方法について説明します。

 **注:** サーバーやアプリケーションを監視するには、まず、データ ソースを定義する必要があります。

サーバーのパフォーマンスは、テストの進行に合わせてグラフィカルな実況表示で見ることができます。サーバー監視によりサーバー側の結果が生成されますが、これらは、後で確認や比較を行えるようにアーカイブにすることができます。監視は、サーバーのボトルネックを明らかにし、その場所を特定して解決する働きもするので、オペレーティング システムとアプリケーション サーバーのパフォーマンスを調べることができます。

自動監視

負荷テストの開始時に Silk Performer プロファイルで **自動的に監視を開始する** が有効になっている場合、Silk Performer は、プロファイルに割り当てられている監視テンプレートを使って Performance Explorer を起動します。

すべての監視レポートが、生成された TSD ファイルの書き込みと保存を負荷テスト ディレクトリに対して開始します。これらの監視ファイルの書き込みは、負荷テストが完了すると自動的に停止します。

生成される TSD ファイルの名前は、r@NAMEOFMONITOR@STARTTIME.tsd という形式になります。ここで、NAMEOFMONITOR は監視レポートの表題です。負荷テストを実行した後は、監視レポートごとに TSD ファイルがワークスペースに定義されています。

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. **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. 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 Performer15.0\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witch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. **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. **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. **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. **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** ウィンドウが表示されます。
- 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 (¥¥<server name>¥c¥¥IBM¥WebSphere¥AppServer¥java など) を開きます。



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
セキュリティ設定		
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	
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** 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、ファイル sas.client.props への URL パスにする必要があります。デフォルトの sas.client.props ファイルはフォルダ %INSTALL_DIR%/Include/jmx-config/WebSphere%MAJOR_VERSION% にあります。例: WebSphere 7 の場合、URL パスは C:\Program Files\IBM\SilkPerformer\10.0\Include\jmx-config\WebSphere7 です。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 Files\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 フォルダにあるファイル ssl.client.props への URL パスにする必要があります。例: file:/C:\Program Files\IBM\WebSphere\AppServer1\profiles\AppSrv01\properties\ssl.client.props。

ファイル ssl.client.props 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。ファイル sas.client.props および ssl.client.props 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終了 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

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. 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。
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、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 **終了** 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. **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. 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 **終了** 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 Performer15.0¥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. **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。

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、DB 管理者権限を有して目標のデータベース サーバーを認識する必要があります。
- 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 Performer15.0\Monitors にあります。

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. **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. **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. **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。

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 Performer15.0\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 シェルデー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. **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. **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	文字列	ps -ef egrep -c ¥".*¥"
Command1.Active	論理値	true
Command1.Column	数	1
Command2	文字列	mpstat head -n 2 tail -1
Command2.Active	論理値	true
Command2.Column	数	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。

監視プロジェクトの変更方法については、『Silk Performer Advanced Concepts』を参照してください。

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ンスがある場合) 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 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= l 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;
```

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. **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. **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. **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. 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 ¥wls_server_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。

リモート マシンの監視

このセクションでは、リモート 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。

リモート マシン監視の要件

Windows 7 および Windows Vista の対象マシン

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 2000 の対象マシン

- ユーザーは Administrators グループのメンバであること
- ユーザーは Users グループのメンバで、ネットワーク経由でコンピュータへアクセス 権限があること

Windows XP の対象マシン

- ユーザーは Administrators グループのメンバであること
- ユーザーは Users グループのメンバで、次のネットワーク アクセス設定のいずれかにパス Software ¥Microsoft が追加されていること:[ローカル ポリシー > セキュリティ オプション] 設定の下の [リモートからアクセスできるレジストリのパス]

Windows 2003 の対象マシン

- ユーザーは Administrators グループのメンバであること
- ユーザーは Network Configuration Operators グループと Performance Monitor Users グループのメンバであること
- ユーザーは Users グループのメンバで、次の [ネットワーク アクセス] 設定のいずれかにパス Software¥Microsoft が追加されていること
 - [リモートからアクセスできるレジストリのパス] 設定
 - [ローカル ポリシー > セキュリティ オプション] 設定の下の [リモートからアクセスできるレジストリのパスおよびサブパス]

これは、信頼される側のさまざまなドメインにあるマシンとそれらのドメインからのユーザーを用いたテストでうまく機能します。

詳細の参照先

URL からの監視に関する詳細については、次の URL を参照してください。

http://patrolexpress.bmc.com/help/en_US/gal_webhelp/trouble_windows.htm


Windows NT でリモート監視を有効にする

次のファイルに対して、少なくとも読み取りアクセス権を許可する必要があります。

- "%windir%\system32\PERFCxxx.DAT

- "%windir%\system32\PERFHxxx.DAT

ここで xxx はシステムの基本言語 ID です。たとえば、英語では 009 を使います。

 **注意:** レジストリ エディタの使い方を間違えると、システム全体に及ぶ深刻な問題を引き起こし、Windows の再インストールが必要になる場合があります。このツールは、各自の責任においてご使用ください。

1. REGEDT32 を使用して、ユーザー アカウントに HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib とそのキーのすべてのサブキーに対して、少なくとも読み取りアクセス権を許可します。
2. REGEDT32 を使用して、ユーザーに HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg に対して少なくとも読み取りアクセス権を許可します。

Windows XP でリモート監視を有効にする

Windows XP OS を実行しているコンピュータの監視を試みると、エラー メッセージが表示されることがあります。この場合は、リモート監視を有効にします。

1. 対象となる Windows XP コンピュータで Windows エクスプローラを開きます。
2. ツール > フォルダ オプション を選択します。
3. 表示 タブをクリックします。
4. 簡易ファイルの共有を使用する チェック ボックスをオフにします。
5. OK をクリックします。
6. 対象となる Windows XP コンピュータの特権を以下で確認します。
 - システム ルートが NTFS パーティション上にある場合、次のファイルに対して少なくとも読み取りアクセス権が必要です。
 - %SystemRoot%\System32\Perfc009.dat
 - %SystemRoot%\System32\Perfh009.dat
 - リモート コンピュータ上の次のレジストリ キーに対して、少なくとも読み取りアクセス権が必要です。
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib
7. Windows XP コンピュータを再起動します。

リモート コンピュータに対して管理者権限を持つユーザーは、これでリモート カウンタを参照できるようになります。

Windows コンピュータで Diskperf を実行する

監視対象のコンピュータ上で Disk Performance Statistics Driver (Diskperf) が実行されたことがない限り、Windows パラメータ セットは、そのコンピュータに関する情報を収集できません。


デフォルトでは、Diskperf は I/O マネージャ ディスク ドライバ スタック内で Disk Performance Statistics Driver をフォルトトレラント ドライバ Ftdisk.sys の上位にインストールします。デバイス コントロール パネルを使用して、Ftdisk が構成内で起動されているかどうかを判断します。

1. コマンド プロンプト ウィンドウを開きます。
2. コマンド プロンプトで、下に示したコマンドの 1 つを入力し、Enter キーを押します。

オプション **説明**

diskperf -y 標準ディスク構成のカウンタを有効化します。

オプション	説明
diskperf	ディスク カウンタが有効であるかどうかを示します。
diskperf -yv	ミラー セットまたはストライプ セット、および、その他の隣接しないパーティション セットのディスク カウントを有効化し、I/O スタックのフォルトトレラントドライバの下位にパフォーマンス統計ドライバをインストールします。
diskperf -n	ディスク カウンタを無効にします。

 **注:** Diskperf を実行するには、対象コンピュータの Administrators ローカル グループのメンバーである必要があります。Ftdisk を含むディスク構成内の物理ディスクを監視するには、diskperf -yv コマンドを実行します。

リモート コンピュータで Diskperf を実行する

監視対象のコンピュータ上で Disk Performance Statistics Driver (Diskperf) が実行されたことがない限り、Windows パラメータ セットは、そのコンピュータに関する情報を収集できません。リモート コンピュータを使用している場合は、必ず情報を収集できるように次の手順を実行します。

1. コマンド プロンプト ウィンドウを開きます。
2. コマンド名を入力し、その後ろにコンピュータ名を続けます。
たとえば、diskperf¥¥thecomputername と入力します。
3. リモート コンピュータを再起動します。

Perflib サブキーに対する読み取りアクセス権の付与

パフォーマンス モニタ (PerfMon) アカウントが監視対象コンピュータのレジストリに接続した後、監視対象コンピュータ上のそのアカウントには、PerfMon で使用するサブキーへの読み取りアクセス権が必要です。コンピュータでは、Perflib¥LanguageID サブキーの ACL (アクセス制御リスト) を使用して、レジストリ内の PerfMon データにアクセスできるユーザーを決定します。ここで、<LanguageID> は、Windows NT でインストールされている言語 (国語) を表す数値コードです。Perflib¥LanguageID サブキーは、以下のレジストリパスにあります。

```
HKEY_LOCAL_MACHINE¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Perflib
¥<LanguageID>
```

Perflib サブキーに読み取りアクセス権を付与する

デフォルトでは、Everyone グループに Perflib¥LanguageID サブキーへの読み取りアクセス権がありません。監視対象コンピュータでのデフォルト権限が変わった場合は、Perflib¥LanguageID サブキーの読み取り権限をアカウントに付与しなければならない場合があります。

1. 監視対象のコンピュータで Regedt32 を起動します。
2. <言語 ID> サブキーをクリックします。
3. **セキュリティ > アクセス許可** を選択します。サブキーの ACL が **<レジストリ キー> のアクセス許可** ダイアログ ボックスに表示されます。
4. **追加** をクリックして、ACL にアカウントを追加し、読み取り権限を許可します。
5. **セキュリティ > アクセス許可** を選択して、言語 ID サブキーへのパスにある各サブキーの ACL をチェックします。

アカウントが、どの ACL においてもすべてのサブキーに対して読み取りアクセス権 (許可) を持っていることを確認します。

リモート マシン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。

- コンピュータ名が見つかりません
- Diskperf がインストールされていません
- Perflib サブキーに対する読み取りアクセス権がありません
- Windows XP マシンに接続できません
- Error=AGENT_ERROR_TCP_SEND_REQUEST_FAILED

コンピュータ名が見つかりません

考えられる原因	解決策
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、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。

Diskperf がインストールされていません

考えられる原因	解決策
監視対象のコンピュータ上で Disk Performance Statistics Driver (Diskperf) が実行されたことがない限り、Windows パラメータ セットは、そのコンピュータに関する情報を収集できません。	監視対象の Windows コンピュータで Diskperf を実行します。

Perflib サブキーに対する読み取りアクセス権がありません

考えられる原因	解決策
デフォルトでは、Everyone グループに Perflib ¥LanguageID サブキーへの読み取りアクセス権があります。監視対象コンピュータでのデフォルト権限が変わった場合は、Perflib ¥LanguageID サブキーの読み取り権限をアカウントに付与しなければならない場合があります。	Perflib¥<LanguageID> サブキーの読み取り権限をアカウントに付与します。

Windows XP マシンに接続できません


考えられる原因	解決策
このエラーは、Windows XP オペレーティング システム を実行しているコンピュータの監視時に発生する場合があります。	リモート監視を有効にします。

Error=AGENT_ERROR_TCP_SEND_REQUEST_FAILED

考えられる原因	解決策
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	以下のプロキシ設定を確認します。 <ul style="list-style-type: none">• 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

考えられる原因	解決策
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。	<ul style="list-style-type: none"> RSM コンピュータからプロキシ サーバーに ping できること 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

SYSTEM: 5 - アクセスが拒否されました

考えられる原因	解決策
CUSTOM / PERFMON オプションを使用して Windows 2008 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 - access denied エラーが生成されます。	<p>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 データソースウィザード 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。</p> <p> ヒント: 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、別のユーザーとして実行 を選択します。</p>

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。

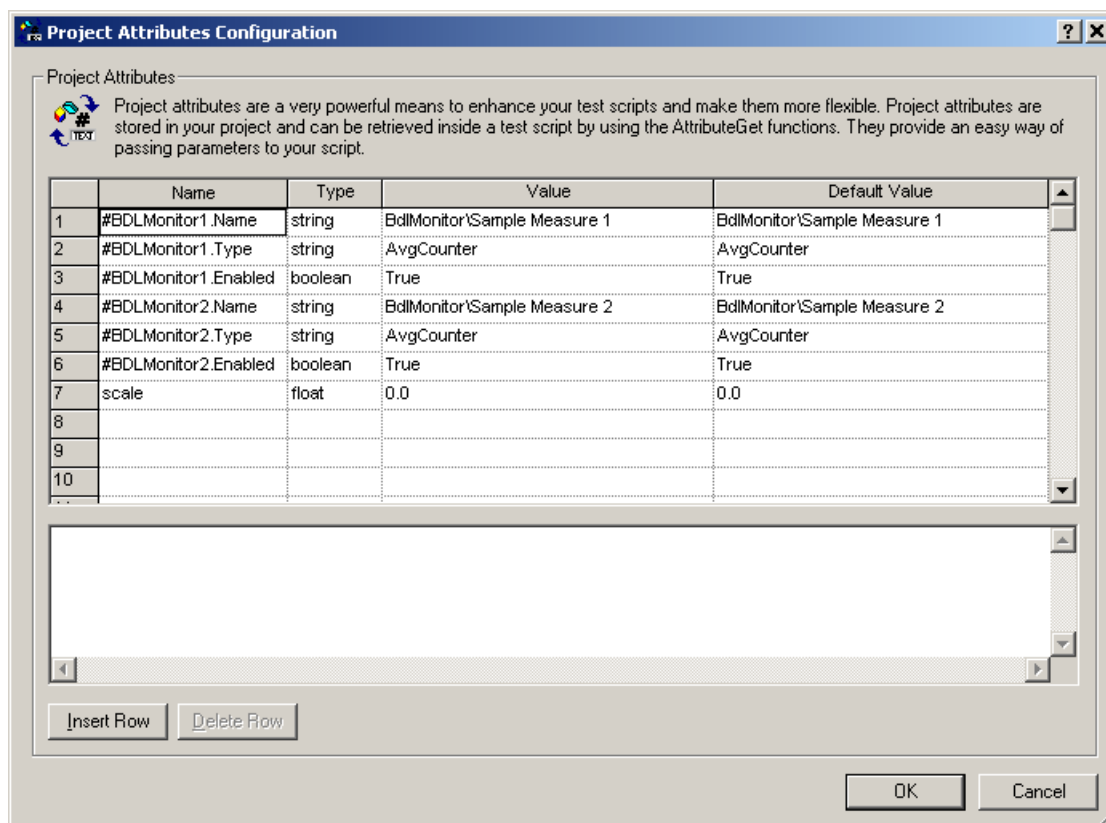
bdlMonitor.bdh で定義されるラッパー関数は、BDL リアルタイム監視に必要な追加機能を配布するために使用されます。次の関数の使用の詳細については、補足チュートリアルを参照してください。

- 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、**データ ファイル** ノードの**プロジェクト** ツリーで利用できます。 .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。

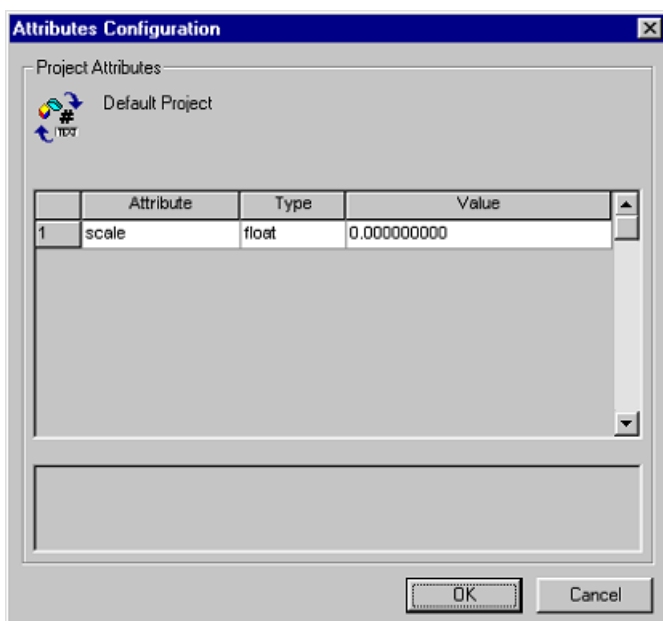
ファイル > プロジェクトのエクスポート を選択します。 Zip を使用して 1 つのアーカイブ ファイルに圧縮する チェック ボックスをオンにします。

エクスポートの場所に、C:\Program Files\Silk\Silk Performer\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

- Performance Explorer を開始して、監視 > データ ソースの追加 を選択します。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。
- 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。
- 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)。
- 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。
- 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。



- 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。

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。

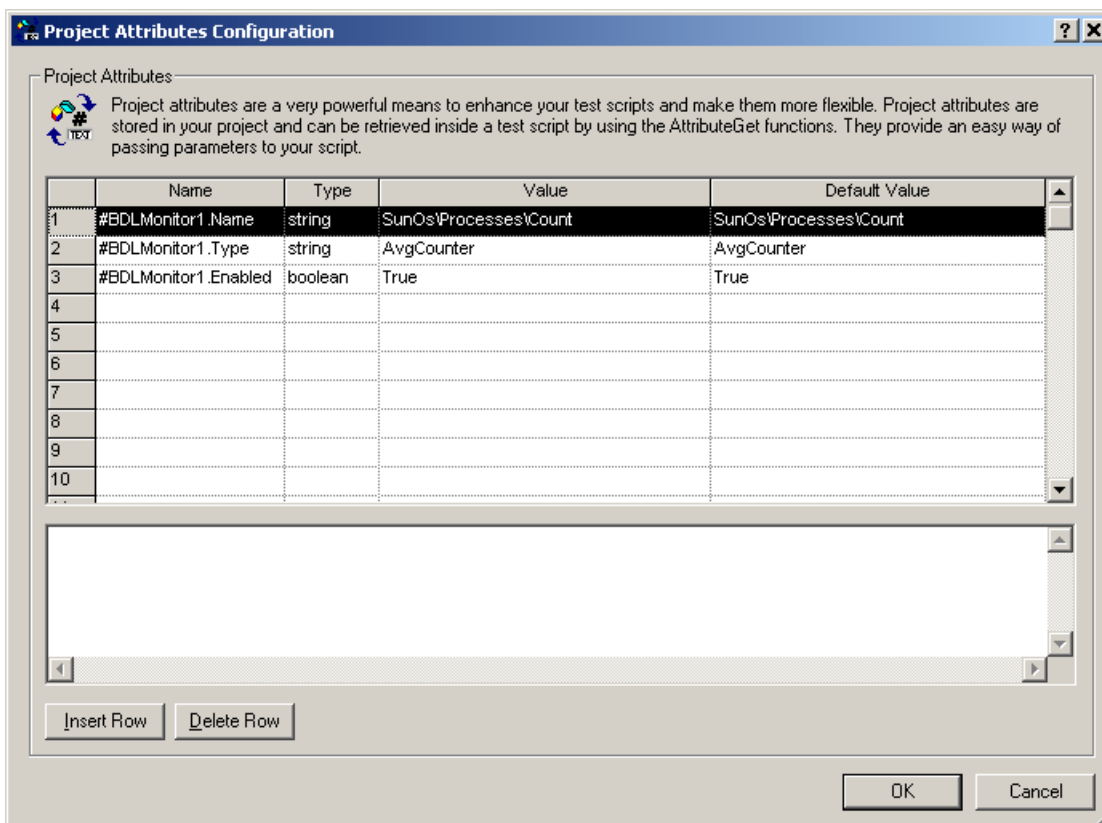
OS

```
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
    WebTcpipConnect(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 を印刷するかログ 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>
```

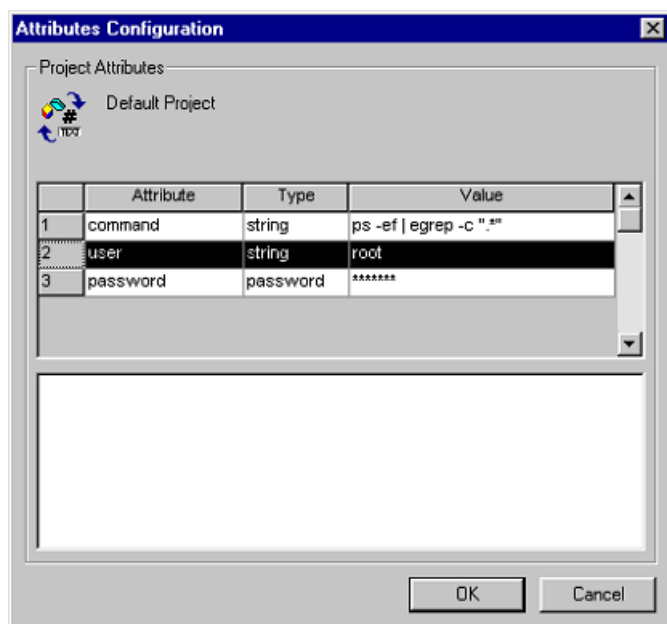
タイプの設定では、「¥」で区切られた階層を指定してください。この階層は Performance Explorer に反映されます。

Performance Explorer 内部での監視プロジェクトの使用

1. Performance Explorer を起動して、**監視 > データソースの追加** に進みます。 **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

名前	タイプ	値
#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	/
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 に反映されません。

Performance Explorer での監視プロジェクトの使用

1. Performance Explorer を開始して、**監視 > データソースの追加** に進みます。**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。

データの収集

このセクションでは、プロジェクトに監視テンプレートを割り当てる方法と、グラフやレポートに含めることができるデータの種類について説明します。

監視オプションを設定する

Silk Performer では、プロジェクト用の監視テンプレートを定義し、負荷テストを開始したときに Performance Explorer を自動的に起動するプロジェクトを構成できます。

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。



ヒント: あるいは、メニューバーから **設定 > アクティブ プロファイル** を選択することもできます。

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。左側のショートカット リストに **再生** カテゴリが表示されます。

3. ショートカット リストから **結果** アイコンをクリックします。
4. **監視** タブをクリックします。
5. テストを開始したら Silk Performer の監視機能が自動的に起動されるようにするには、**監視オプション** 領域の **自動的に監視を開始する** チェック ボックスをオンにします。
6. プロジェクトに最適な監視テンプレートを自動的に使用するには、**デフォルト監視テンプレートを使用する** をクリックします。
たとえば、Web プロジェクトを作成している場合、Web 負荷テストに役立つ測定値をテンプレートで指定します。

7. カスタム監視テンプレートを使用するには、**カスタム監視テンプレートを使用する** をクリックし、次のいずれかのステップを実行します。

- サーバー監視に使用するカスタム テンプレート ファイル (.pew) の名前を入力します。 Silk Performer により、標準監視テンプレートのコピーが作成されます。
- 名前フィールドのフォルダ アイコンをクリックして、既存の監視テンプレートを選択します。

8. 省略可能： **カスタム監視テンプレートの編集** をクリックして、任意の監視パフォーマンス データを追加または削除します。

このボタンをクリックすると、Silk Performance Explorer が開きます。以下の手順を実行します。

- a) 任意の監視パフォーマンス データを追加や削除します。
- b) 変更をテンプレートに適用するには、Silk Performance Explorer ワークスペースを保存します。

9. Windows のパフォーマンス モニタに表示される追加パフォーマンス測定値のデータを計算するには、**パフォーマンス モニタのインテグレーション** 領域の **オンライン パフォーマンス データを計算する** チェック ボックスをオンにします。

このデータを使用すると、同時使用ユーザー、トランザクションのスループット、送信/受信されたデータ、および実行された SQL 文を表示できます。

10 設定を保存するには、**OK** をクリックします。

テストを実行すると、Silk Performer は、テスト対象のサーバーの種類に関連するサーバー パフォーマンス データを表示します。

クライアント側測定値

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。

Performance Explorer の構成

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. **監視 > クライアント側データ ソースの追加** 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. **ホスト名** ボックスに、コントローラ マシンの名前を指定します。
3. **エイリアス** ボックスに、クライアント側測定値を提供するコントローラ マシンのエイリアスを指定します。
4. **終了** をクリックします。データ ソースの接続エントリがメニュー ツリーに表示されます。このデータ ソースで利用可能なクライアント側測定値 (ページ タイマなど) が、メニュー 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 データの測定値を提供するように指示します。Performance Explorer の **クライアント側測定値** ページにある緑色のデータソース接続アイコンは、データ ソースに正常に接続していることを示します。赤色のデータソース接続アイコンは、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

Performance Explorer のメニュー 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. **クライアント側測定値** ペインをクリックします。
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。ノードを削除するかどうか確認ダイアログ ボックスが表示されます。

3. はい をクリックします。

監視タイプ

監視の目的と利用可能なタイプについて説明します。

監視タイプの概要

監視は、システムのパフォーマンス カウンタを収集し、指定した間隔でその測定値をポーリングします。監視にパフォーマンス カウンタを追加するには、パフォーマンス カウンタをデータ ソースとして定義しなければなりません。

監視タイプには次のものがあります。

- グラフ - 測定したカウンタをグラフに表示します。
- レポート - 測定したカウンタを TSD ファイルに出力します。

ポーリングの間隔


エージェントとサーバーの使用率が高いと、パフォーマンス測定値のポーリングは CPU リソースの負荷になります。使用率の高いシステムで間隔を狭めると、テスト結果に影響を及ぼす可能性があります。

 **注:** 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。

サーバー側とクライアント側の測定値の比較

負荷テストを実行した後でサーバー側の測定値をクライアント側の測定値と比較するには、監視レポートを定義する必要があります。

監視レポートは、時系列データ ファイル (TSD) 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。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。

 **注:** 監視グラフに表示されるデータは、TSD ファイルに出力することができないため、後で概要レポートに含めることはできません。

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. **ファイル > 新規作成 > 監視グラフ** をクリックします。
2. 複数のグラフを追加するには、**監視** メニュー ツリーの測定値を右クリックして、**ビューに追加** をクリックします。
代替方法: **監視** メニュー ツリーから監視グラフに、測定値をドラッグ アンド ドロップします。
3. 監視グラフ内のホスト名を右クリックして、**プロパティ** をクリックします。**プロパティ** ダイアログ ボックスが開きます。
4. **監視** タブをクリックします。
監視グラフには、新しい値を取得する更新頻度を定義する間隔があります。
5. **間隔** ボックスに、どれぐらいの頻度で保存するかを指定します。
6. **ページ サイズ** ボックスに、各ページに表示する間隔の数を指定します。
7. **履歴サイズ** ボックスに、グラフの履歴に保存する間隔の数を指定します。
8. **OK** をクリックします。

監視レポートを作成する

監視レポートを定義すると、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。監視レポートは、時系列データ ファイル (TSD) 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. **ファイル > 新規作成 > 監視レポート** をクリックします。



注: SAPGUI データ ソースについては、監視間隔を 20 秒以上にすることをお勧めします。これは、監視レポートの **TSD 出力プログラム** タブの **オプション** ダイアログで指定できます。

2. レポートに複数の測定値を追加するには、**監視** メニュー ツリーの測定値を右クリックして、**ビューに追加** をクリックします。

代替方法: **監視** メニュー ツリーからレポートに、測定値をドラッグします。

3. **監視 > 監視データの書き出し** をクリックします。

代替方法: ツールバーの **監視データの書き出し** をクリックします。

名前を付けて保存 ダイアログ ボックスが開きます。

4. **ファイル名** ボックスに、ファイル名を入力します。

5. **ファイルの種類** ボックスに **監視の時系列ファイル (r@*.tsd)** が指定されていることを確認します。

6. **保存** をクリックします。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レポートが含まれている場合、すべての監視レポートは、テストが開始したときに自動的に書き出しを始め、負荷テストが完了すると停止します。

すべての監視レポートは負荷テストの結果ディレクトリに保存されます。

ホスト データ

Performance Explorer では、ホスト データの名前の変更、複製、および更新を行うためのウィザードが提供されています。

測定値の複製または名前の変更を行う

測定値の複製や名前の変更を行い、異なるホストで同じ測定値を監視します。

複製機能では、測定値を一方のホストから他方へとコピーします。名前の変更とは、測定値のホスト名を変更することです。

1. **監視** ペインのホスト名を右クリックしてから、**ホスト名の変更** または **ホストの複製** をクリックします。**ホスト複製ウィザード** の **接続パラメータ** ページが表示されます ([ホスト名の変更] をクリックした場合は [ホスト名変更ウィザード])。

2. **ホスト名** ボックスに、複製または名前を変更するデータ ソースの名前を入力します。

3. 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り詳しく説明するような同義語とします。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。




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。

4. **次へ** をクリックします。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。

5. 省略可能:測定値を無視する場合は、**このグループの測定値を無視する** チェック ボックスをオンにします。
6. **終了** 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。

既存の監視とワークスペースのユーザー名とパスワードを変更する

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にとっては不便です。そのような環境に対応して、Performance Explorer では既存の監視とワークスペースのユーザー名とパスワードを変更できます。

1. **監視** ペインで資格情報を編集するホスト名を右クリックしてから、**ホストの資格情報の編集** をクリックします。**ホストの資格情報の編集** ウィザードが開きます。
2. 資格情報を変更したいホスト名とエイリアスであるか確認してから、**次へ** をクリックします。**接続パラメータ** ページに **属性の変更** ボタンが表示されます。
3. **属性の変更** をクリックします。**属性の設定** ダイアログ ボックスが開きます。
4. ユーザー名を変更するには、**ユーザー** 行の **値** セルを変更します。
5. パスワードを変更するには、**パスワード** 行の **値** セルを変更します。
6. **OK** をクリックします。
7. **終了** をクリックします。

測定値

測定値 (または測定値タイプ) とは、負荷テスト中に収集されるパフォーマンス測定値を意味する総称的な用語です。測定値にはタイマとカウンタがあります。タイマは、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。

測定値はグループに分類できます。各測定値グループには、特定の用途または特定のアプリケーション タイプに関連付けられた測定値タイプが、あらかじめいくつか含まれています。

Performance Explorer では以下の測定値タイプを提供しています。

- 要約測定値
- トランザクション測定値
- Web フォーム測定値
- Web ページ測定値
- CORBA 固有の測定値
- SQL データベース固有の測定値
- ミドルウェア固有の測定値
- Citrix 固有の測定値
- カスタム タイマ測定値
- カスタム カウンタ測定値

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タイプや、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。要約測定値グループは、カウンタのみで構成されており、タイマは含まれません。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。


全般測定値

概要レポートの先頭に表示される測定値タイプを示します。

測定値	説明
アクティブなユーザー	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 ユーザーと見なされるのは、開始済みで、現在、以下のいずれかの状態にある場合です。実行中、準備中、データベース待ち、ドキュメント ダウンロード中、リソース待ち、待ち合わせ、思考時間、およびシャットダウン中。
トランザクション	正常に終了したかどうかに関わらず、終了した Silk Performer トランザクションの数。 概要レポートには、以下の内容が表示されます。 <ul style="list-style-type: none">1 秒間に終了したトランザクションの数を示すグラフ1 秒間に終了したトランザクション数の平均終了したトランザクションの合計数
エラー	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。重大度が 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。

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	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。
メモリ	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。  注: 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。
応答性	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。 SAP GUI またはブラウザ駆動の負荷テスト (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。

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	ブラウザレベルの関数が送信した 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

測定値	説明
失敗したリクエスト	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。
送信されたリクエスト データ	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。
受信したレスポンス	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 データ	ブラウザレベルの関数が受信した 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	ブラウザレベル、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。
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、ブラウザレベル、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、ブラウザレベルの関数は 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	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、ブラウザレベルの関数でのみ使用されません。

測定値	説明
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 の関数のみです。
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 リクエスト数。この測定値を増加させるのは、Web の関数のみです。
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 の関数のみです。
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 の関数です。
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 の関数です。
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 の関数です。
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 の関数です。
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 の関数です。
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。
送信された HTTP クッキー	ブラウザレベルの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	ブラウザレベルの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	Java スクリプト シミュレーションが有効の場合に、ページレベルの API またはブラウザレベルの API がリクエストした HTML ページの数。

IIOP 関連の測定値タイプ

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 ロケット リクエスト	Silk Performer が送信した IIOP ロケット 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。

測定値	説明
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タイプ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。
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)。トランザクションのビジー時間は、思考時間以外のトランザクション レスポンス時間です。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)。トランザクションのビジー時間は、思考時間以外のトランザクション レスポンス時間です。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)。

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。  注: 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
解析	<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> 注: 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>
実行	<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ため、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>
実行 (直接)	<p>SQL コマンドの準備および実行にかかった時間 (秒)。</p>

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。Tux_tpcall、Tux_tpacall、Tux_tpsend、Tux_tprecv、Tux_tpgetreply、Tux_tpenqueue、および Tux_tpdequeue 関数が発行した、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 関数で使用される 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</p>

測定値	説明
	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。

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)。

 **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せん。

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。

デフォルト オプション


グラフやレポートのデフォルト 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 メニューバーから、**設定 > オプション** をクリックします。

Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。

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. **倍率オプション** 領域を使用して、実際の時系列データ値を表示するか、0 から 100 までの値に拡大縮小するか指定します。
 - **表示値を 0 から 100 までの値に拡大縮小する** オプションをクリックすると、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。
 - **実際の値を表示する** オプションをクリックすると、拡大縮小された値ではなく、実際の時系列データ値を表示できます。
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

1. Performance Explorer メニュー バーから、**設定 > オプション** をクリックします。
Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。
2. **時系列** タブをクリックします。
3. デフォルトで NULL 値を表示するには、**デフォルト設定** 領域の **NULL 値を表示する** チェック ボックスをオンにします。
ツールバーの **null 値の表示** アイコンをクリックして、時系列ごとに個別に NULL 値を表示するかどうか指定できます。
4. **要約とカウンタ値の表示間隔** ボックスの一覧から、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 ボックスの **時系列** タブを使用します。
5. **デフォルトの線幅** リスト ボックスから、新しい時系列をグラフに追加するとき使用する線幅を選択します。
6. 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。
7. **数値の桁数** テキスト ボックスに、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、グラフ内の各時系列の正確な測定値を表示できます。
8. **OK** をクリックします。

結果グラフをエクスポートするときのデフォルトのオプションを設定する


1. Performance Explorer メニュー バーから、**設定 > オプション** をクリックします。
Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。
2. **HTML ビュー** タブをクリックします。
3. **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** オプションをクリックすると、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) オプションをクリックすると、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 メニュー バーから、**設定 > オプション** をクリックします。
Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。
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. **マーカースの色** ボックスの一覧から、マーカース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. **マーカースの高さ** ボックスに、マーカース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 メニュー バーから、**設定 > オプション** をクリックします。
Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。
2. **監視** タブをクリックします。
3. **監視オプション** 領域の **間隔** ボックスに、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** ボックスに、各グラフの水平軸上に現在表示する、測定ポイントの数を設定します。
5. **履歴サイズ** ボックスに、グラフ全体の水平軸上に保存する、測定ポイントの数を設定します。
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。

デフォルトの時系列データ出力オプションを設定する

1. Performance Explorer メニュー バーから、**設定 > オプション** をクリックします。
Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。
2. **TSD 出力プログラム** タブをクリックします。
3. **間隔** ボックスで、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 メニュー バーから、**設定 > オプション** をクリックします。
Silk Performance Explorer のオプション ダイアログ ボックスの **グラフ** タブが開きます。
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、グラフおよびレポート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数あります。測定値の表示に使用される色、線の幅、および線のスタイルを変更できます。倍率オプションを変更したり、マーカーの挿入または削除もできます。

グラフを右クリックして、**表示可能** チェックボックスをオフにすると、測定値は非表示になります。

 **注:** 非表示の測定値は、エクスポート後、概要レポートに表示されません。

これらの変更はデフォルト表示設定で行います。これらの変更を行った後に開かれたすべてのビューは、新しいデフォルト設定が適用されます。選択したビューをカスタマイズして、すぐに結果を表示することもできます。

1. カスタマイズするグラフを右クリックして、**プロパティ** をクリックします。
代替方法: ツールバーの**プロパティの表示** をクリックします。
プロパティ ダイアログ ボックスの **全般** タブが開きます。
2. **タイトル** ボックスに、現在のグラフの名前を入力または変更します。
3. **説明** ボックスに、現在のグラフの説明を入力します。
4. **グラフ** タブをクリックして、現在のグラフの表示オプションを編集または変更します。
5. **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。

6. **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。
7. **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。
8. **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。
9. **倍率オプション** 領域を使用して、実際の時系列データ値を表示するか、0 から 100 までの値に拡大縮小するか指定します。
 - **表示値を 0 から 100 までの値に拡大縮小する** オプションをクリックすると、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。

- **実際の値を表示する** オプションをクリックすると、拡大縮小された値ではなく、実際の時系列データ値を表示できます。

10OK をクリックして、グラフに変更を適用します。

11 グラフ内の時系列データはすべて、右クリックすると、その外観を変更できます。右クリックすると、**時系列の色、線の幅、および線のスタイル** のリストボックスが開きます。

12 グラフ内のある領域を拡張するには、拡張する領域の左端をクリックして、その領域の右端にドラッグします。ドラッグすると領域が強調表示され、マウスボタンを放すと拡張されます。

13 領域の拡張を元に戻すには、グラフで逆方向にドラッグするか、ツールバーの **縮小** をクリックします。

14 ツールバーの **ソースタイプ** ボックスの一覧から、対象とする期間の測定値情報を表示する方法を選択します。Performance Explorer は、以下の項目を計算して表示します。

- 値の合計
- 値の平均
- カウント
- 最小値または最大値
- 平均からの標準偏差

15 ツールバーの **時系列の倍率** ボックスを使って、選択された要素がスケーリングされる倍率を変更します。

16 ツールバーの **時系列タイプ** ボックスを使って、グラフ要素の表示方法を以下の方法から選択します。

- 単純な [線] (デフォルト)
- [ローソク] グラフ。各測定値が中央にボタンを持つ垂直マーカで表示されます。中央のボタンは、期間中に測定された平均値を表します。マーカの上端および下端は、平均からの最大および最小の偏差を表します。
- [ローソク足] グラフ。2つの値が追加され、マーカの上下に線として表示されます。これらは、期間中に測定された最大値および最小値を示します。
- [領域] 時系列タイプ。グラフの線より下側の領域全体が色で塗りつぶされます。

17 ツールバーの **時系列マーカの表示** をクリックして、実際の測定値をグラフに表示または非表示にします。

18 ツールバーの **null 値の表示** をクリックして、測定値がゼロの値を表示または非表示にします。

19 ツールバーの **強調表示の表示/非表示** をクリックして、選択されたグラフ要素を太線で強調表示します。

20 ツールバーの **凡例の表示/非表示** をクリックして、測定値タイプの凡例が表示されるグラフウィンドウの下部を表示または非表示にします。

21 ツールバーの **フレームの表示/非表示** をクリックして、タイトルバーを表示または非表示にします。

22 ツールバーの **倍率の表示/非表示** をクリックして、実際の値を表示するか、垂直軸の 0 から 100 の間に値をスケーリングして表示するか選択します。



注: 明確に表示される倍率が、最初に選択されます。

23 ツールバーの **対数 Y 軸** をクリックして、グラフの垂直軸を標準スケーリングにするか、または対数スケーリングにするか選択します。

24 ツールバーの **重ね合わせの表示/非表示** をクリックして、表示されているすべての測定値の開始位置を水平軸上のゼロに一致させます。

これは、開始時間が異なる測定値を比較する場合に便利です。

結果の分析

Performance Explorer では、負荷テスト結果の表示、レポート生成、分析を目的としたさまざまな機能をサポートしています。選択した測定値を、さまざまなグラフィック形式および表形式で表示できます。そのようなユーザー定義のテスト結果には、要素を必要な数だけ含めることができます。また、異なる負荷

テストの結果を単一のグラフ内で比較することさえできます。さらに、過去のサーバー監視の再現および検討をサポートする機能があります。

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 で使用される負荷テストプロジェクトごとに、Performance Explorer を使用して概要レポート テンプレートを定義できます。



注: Performance Explorer のコマンドライン インターフェイスには、Performance Explorer 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。

1. Performance Explorer 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. 概要レポートを Performance Explorer で生成するための概要レポート テンプレートをこのテストプロジェクトに定義するには、**時系列** ページの **概要レポート テンプレート** エリアにある **参照 (...)** ボタンをクリックします。
 - a) **テンプレート ファイルの選択** ダイアログ ボックスで、プロジェクトに適用する概要レポート テンプレート (OVT) ファイルを選択します。
 - b) **開く** をクリックします。

ここで指定したファイルは、このテスト プロジェクトのあらゆる概要レポートを生成するための概要レポート テンプレートとして使用されることとなります。新しい概要レポートを手動で作成する際にも、このテンプレート ファイル名が **概要レポート テンプレート** ワークフロー ダイアログ ボックスであらかじめ選択されています。

6. 設定を保存するには、**OK** をクリックします。

結果の表示

負荷テスト結果の分析は、通常、大きく 4 つのステップで構成されます。

1. テスト結果の初期評価のために **概要レポート** を調査する。
2. 一般的な測定値タイプ (アクティブな仮想ユーザーの数、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) を検査する。
3. テストしているアプリケーションのタイプにおいて特に重要な測定値を確認する。Performance Explorer には、ほとんどすべてのタイプのアプリケーションに対して、最適な測定値タイプを含む事前定義済みのテンプレートがあります。
4. さらに詳細な測定値を選択して検討する。測定値は、いくつでも選択してグラフおよびレポートで組み合わせることができます。

テスト結果を表示する

Performance Explorer には、分析用に負荷テスト結果を選択する場合に役立つウィザードがあります。

1. **検討** タブをクリックします。
2. 次のいずれか 1 つのステップを行います：
 - **検討 > 負荷テストの追加** を選択します。
 - **結果** メニュー ツリーを右クリックし、**負荷テストの追加** を選択します。

プロジェクトの選択 ダイアログ ボックスが表示され、<マイ ドキュメント>¥Silk Performer15.0¥Projects で使用可能なすべてのプロジェクトが一覧表示されます。

3. 負荷テスト結果を表示するプロジェクトを選択して、**次へ** をクリックします。**負荷テストの選択** ダイアログ ボックスが開き、選択プロジェクトのすべての負荷テストが一覧表示されます。
4. 分析する負荷テストを選択して、**完了** をクリックします。選択した負荷テストの概要レポートが開きます。

Performance Explorer の **検討** ページのメニュー ツリーに、負荷テスト結果が表示されます。これは、ファイルではなく結果構造に基づいて表示されます。

メニューツリーには、以下の構造で結果が表示されます。

- Project_1
 - Loadtest_1
 - クライアントの測定値
 - 健全度管理
 - 要約 (一般)
 - ...
 - トランザクション
 - エージェント
 - Agent_1
 - 健全度管理
 - 要約 (一般)
 - ...
 - トランザクション
 - ...
 - Agent_n
 - ユーザー タイプ
 - UserType_1
 - 健全度管理
 - 要約 (一般)
 - トランザクション
 - ...
 - UserType_n
 - 地域
 - Region_1
 - 健全度管理
 - 要約 (一般)
 - トランザクション
 - ...
 - Region_n
 - ファイル
 - File_1 (たとえば、再マージされたファイル 1)
 - 健全度管理
 - 要約 (一般)
 - ...
 - トランザクション

- ...
 - File_n (たとえば、再マージされたファイル n)
- サーバーの測定値
 - Server_1
 - ...
 - Server_n
- ...
- Loadtest_n
- ...
- Project_n

概要レポート

概要レポートには、負荷テスト結果についての重要な情報が表示されます。

概要レポートの構造

概要レポートには、最も重要なテスト結果が表形式およびグラフィック形式で表示されます。概要レポートは、以下のセクションで構成されます。

- 全般情報
 - プロジェクトの全般設定
 - プロジェクト属性
- 要約テーブル
- 順位
- ユーザー タイプ
- カスタム グラフ
- カスタム テーブル
- 詳細グラフ

セクション	説明
全般情報	<p>全般情報には、管理情報が表形式で、重要な負荷テスト結果がグラフィック形式で表示されます。</p> <p>管理情報には以下の情報が含まれます。</p> <ul style="list-style-type: none"> • プロジェクト名 • プロジェクトの説明 • テスト番号 • 負荷テストの説明 • 負荷テストの日時 • エージェントコンピュータの数 • 動作した仮想ユーザーの数 <p>グラフには、以下の情報が表示されます。</p> <ul style="list-style-type: none"> • アクティブな仮想ユーザーの数 • トランザクションのレスポンス時間測定値 • 時間内に発生したエラーの数 <p>トランザクションのレスポンス時間は、以下のトランザクションに関して測定されます。</p> <ul style="list-style-type: none"> • 正常に実行されたトランザクション • 失敗したトランザクション • キャンセルされたトランザクション

セクション	説明
要約テーブル	<p>追加のグラフには、負荷テストプロジェクトのタイプに関連する測定値の要約が表示されます。たとえば、Web アプリケーションのテストの場合は、Web ページのレスポンス時間測定値がグラフに表示されます。</p> <p>このセクションには、測定値の要約が表形式で表示されます。これらは、すべての仮想ユーザーに関する測定値の総計です。最初のテーブルには、実行されたトランザクションの数や発生したエラーの数などの、全般情報が表示されます。それ以降のテーブルはすべて、テストしているアプリケーションのタイプに関係のある要約情報を提供します。</p>
順位	<p>このセクションは、測定値グループごとに最悪の測定値 10 個の順位を表形式で示します。たとえば、Web アプリケーションのテストの場合は、最も遅い Web ページ、最も大規模な Web ページ、最もサーバー リソースを消費する Web ページ、および最もネットワーク リソースを消費する Web ページが表示されます。</p> <p>ブラウザ駆動型負荷テストの場合、最も低速なアクション時間が表示されます。</p> <p>また、データベース アプリケーションのテストの場合は、最も遅い SQL コマンドと最もリソースを消費する SQL コマンドが表示されます。</p>
ユーザー タイプ	<p>ユーザー グループごとに、詳細な測定値が表形式で表示されます。以下の測定値が含まれます。</p> <ul style="list-style-type: none"> • トランザクション レスポンス時間 • 個々のタイム • カウンタ • テストしているアプリケーションのタイプ (Web、CORBA、COM、TUXEDO のいずれか) に関連するレスポンス時間とスループットの測定値 <p>さらに、エラーや警告もユーザー グループごとに一覧表示されます。</p> <p>必要に応じて、パーセンタイル計算で得られた測定値のパーセンタイル グラフを表示できます。パーセンタイル グラフを使用すると、特定の制限時間内に実行されたアクションの割合を分析できます。パーセンタイル グラフを用いて、システムを対象に定義されたパフォーマンス基準を満たすトランザクション、ページロード時間、カスタム タイマの各割合を判断できます。さらに、グラフの形状を考察することによって、実行時間の分布を分析できます。</p> <p>特定の測定値のパーセンタイル グラフを表示するには、MeasureCalculatePercentiles() 関数を使用します。</p>
カスタム グラフ	<p>このセクションには、手動で追加されたグラフが表示されます。このセクションに対して、いつでもグラフの追加や削除を行うことができます。変更内容をテンプレートとして保存して、今後の要約レポートで使用することができます。</p>
カスタム テーブル	<p>このセクションには、手動で追加されたテーブルが表示されます。このセクションに対して、いつでもテーブルの追加や削除を行うことができます。変更内容をテンプレートとして保存して、今後の要約レポートで使用することができます。</p>

セクション	説明
詳細グラフ	このセクションには、レポートに含まれているグラフの拡大版が表示されます。グラフの縮小版をクリックすると、そのグラフの拡大版にジャンプします。そのグラフを再びクリックすると、縮小版に戻ります。

概要レポートを表示する

Performance Explorer は、負荷テストが完了すると自動的に概要レポートを作成します。

デフォルトでは、概要レポートは自動的に生成されます。レポートの自動生成を無効にしている場合は、ワークフロー バーの **概要レポート** をクリックします。

概要レポートをカスタマイズする

Performance Explorer では、カスタマイズしたグラフやレポートを概要レポートに追加できます。また、グラフやレポートをいつでも概要レポートから削除できますし、変更内容をテンプレートとして保存して、今後生成する概要レポートに適用することもできます。

デフォルトのグラフを概要レポートに追加する

この作業を行う前に、概要レポートを生成して、表示します。

1. Performance Explorer 結果ウィンドウで、グラフを追加する概要レポートをクリックします。
2. ワークフロー バーの **レポートのカスタマイズ** をクリックします。 **カスタム概要レポートウィザード** ダイアログ ボックスが開きます。
3. 概要レポートに表示するすべてのエントリのチェック ボックスをオンにし、**次へ** をクリックします。デフォルトでは、利用できるすべてのエントリが表示されます。
4. **新規ビューの追加** チェック ボックスをオンにして、**次へ** をクリックします。
5. 概要レポートに挿入するグラフのタイプをクリックします。
6. **終了** をクリックします。選択したグラフ タイプの新規ビューが Performance Explorer に表示されます。また、Performance Explorer により、グラフが概要レポートの [カスタム グラフ] セクションに挿入されます。



注: グラフが含まれているビューに対して新しい測定値の追加や既存の測定値の削除を行うと、概要レポート内の対応するグラフが自動的に更新されます。

カスタム グラフを概要レポートに追加する

この作業を始める前に、概要レポートと、概要レポートに追加するグラフを生成します。

1. Performance Explorer 結果ウィンドウで、グラフを追加する概要レポートをクリックします。
2. ワークフロー バーの **レポートのカスタマイズ** をクリックします。 **カスタム概要レポートウィザード** ダイアログ ボックスが開きます。
3. 概要レポートに表示するすべてのエントリのチェック ボックスをオンにし、**次へ** をクリックします。デフォルトでは、利用できるすべてのエントリが表示されます。
4. 作成したグラフのチェック ボックスをオンにします。
5. **新規ビューの追加** チェック ボックスをオンにして、**次へ** をクリックします。
6. グラフに使用するテンプレートのタイプを選択して、**完了** をクリックします。Performance Explorer により、選択したグラフが概要レポートの [カスタム グラフ] セクションに挿入されます。



注: グラフが含まれているビューに対して新しい測定値の追加や既存の測定値の削除を行うと、概要レポート内の対応するグラフが自動的に更新されます。

デフォルトのレポートを概要レポートに追加する

1. Performance Explorer 結果ウィンドウで、レポートを追加する概要レポートをクリックします。
2. ワークフロー バーの **レポートのカスタマイズ** をクリックします。 **カスタム概要レポートウィザード** ダイアログ ボックスが開きます。
3. 概要レポートに表示するすべてのエントリのチェック ボックスをオンにし、**次へ** をクリックします。デフォルトでは、利用できるすべてのエントリが表示されます。
4. **新規ビューの追加** チェック ボックスをオンにして、**次へ** をクリックします。
5. 使用するレポート テンプレートを選択して、**完了** をクリックします。Performance Explorer により、選択したレポートのタイプで新しいビューが表示されます。また、Performance Explorer により、このレポートが概要レポートのカスタム テーブル セクションに挿入されます。



注: レポートが含まれているビューに対して新しい測定値の追加や既存の測定値の削除を行うと、概要レポート内の対応するレポートが自動的に更新されます。

カスタム レポートを概要レポートに追加する

この作業を始める前に、概要レポートと、概要レポートに追加するレポートを生成します。

1. Performance Explorer 結果ウィンドウで、レポートを追加する概要レポートをクリックします。
2. ワークフロー バーの **レポートのカスタマイズ** をクリックします。 **カスタム概要レポートウィザード** ダイアログ ボックスが開きます。
3. 概要レポートに表示するすべてのエントリのチェック ボックスをオンにし、**次へ** をクリックします。デフォルトでは、利用できるすべてのエントリが表示されます。
4. 作成したレポートのチェック ボックスをオンにします。
5. **新規ビューの追加** チェック ボックスをオンにして、**次へ** をクリックします。
6. 使用するレポート テンプレートを選択して、**完了** をクリックします。Performance Explorer により、選択したレポートのタイプで新しいビューが表示されます。また、Performance Explorer により、このレポートが概要レポートのカスタム テーブル セクションに挿入されます。



注: レポートが含まれているビューに対して新しい測定値の追加や既存の測定値の削除を行うと、概要レポート内の対応するレポートが自動的に更新されます。

グラフを概要レポートから削除する

1. Performance Explorer の結果ウィンドウで、グラフを削除する概要レポートをクリックします。
2. ワークフロー バーの **レポートのカスタマイズ** をクリックします。 **カスタム概要レポートウィザード** ダイアログ ボックスが開きます。
3. **次へ** をクリックします。
4. 概要レポートから削除するグラフのチェック ボックスをオフにして、**完了** をクリックします。Performance Explorer により、選択したレポートが、概要レポートの [カスタム グラフ] セクションから削除されます。

レポートを概要レポートから削除する

1. Performance Explorer 結果ウィンドウで、レポートを削除する概要レポートをクリックします。
2. ワークフロー バーの **レポートのカスタマイズ** をクリックします。 **カスタム概要レポートウィザード** ダイアログ ボックスが開きます。
3. **次へ** をクリックします。
4. 概要レポートから削除するレポートのチェック ボックスをオフにして、**完了** をクリックします。Performance Explorer により、選択したレポートが、概要レポートの [カスタム テーブル] セクションから削除されます。

テキスト領域を変更する

Performance Explorer の概要レポートで、多くのテキスト領域が事前に定義されています。必要に応じてこれらのテキスト領域を変更できます。

1. 概要レポートで、**テキストを編集するには、ここをクリックしてください。** をクリックして、対応するテキストを変更します。
2. 今後のレポート用に、レポートをテンプレートとして保存します。

テンプレートを保存する

あとから概要レポートに使用するためにテンプレートを保存します。


1. ニーズに応じて概要レポートをカスタマイズします。
2. **ファイル > テンプレートとして保存** をクリックします。
3. **ファイル名** テキスト ボックスに、新しいテンプレートの名前を入力します。
テンプレートは <my documents>¥Silk Performer15.0¥Projects に保存され、すべてのプロジェクトで使用できます。
4. **ファイルの種類** テキスト ボックスで、**概要テンプレート (.ovt)** が選択されていることを確認し、**保存** をクリックします。

デフォルト テンプレートを設定する

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ために使用するテンプレートを指定します。

1. **設定 > オプション** を選択し、今後、概要レポートを作成するときに、このテンプレートを使用するように指定します。**オプション** ダイアログ ボックスが開きます。
2. **レポート作成** タブをクリックします。
3. **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ます。
4. **テンプレート** テキスト ボックスで、使用するテンプレートの名前と場所を入力します。
代替方法:参照 (...) をクリックして、**テンプレート ファイルの選択** ダイアログ ボックスからテンプレートを選択します。
5. **OK** をクリックします。

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ために使用されます。

 **注:** デフォルトの概要テンプレートが Performance Explorer の /OVT:<テンプレート> コマンドライン オプションを使って指定された場合、または対応する Silk Performer プロジェクトでテンプレートが指定されている場合、この設定は上書きされます。

グラフまたはレポートの名前や説明を変更する

グラフまたはレポートの名前や説明を変更し、表示情報をわかりやすくします。

1. グラフまたはレポートを右クリックして、**プロパティ** をクリックします。**プロパティ** ダイアログ ボックスが開きます。
2. **タイトル** ボックスにタイトルを入力します。
3. **説明** ボックスに説明を入力します。
4. **OK** をクリックします。

時系列データ (TSD) 結果ファイルを指定する

次のタスクを完了する前に、使用する時系列データ (TSD) ファイルを指定します。

- 特定の結果を表示する

- カスタム レポートやグラフを生成する
- 複数のファイルから結果を結合する

1. 検討 > 結果の追加 を選択します。

代替方法:メニュー ツリーでプロジェクト名を右クリックし、**結果の追加** を選択します。

結果ファイルの選択 ダイアログ ボックスが開きます。

2. グラフを表示する TSD ファイルを指定して、**開く** をクリックします。Performance Explorer により、選択されたファイル内のすべての測定値を含むメニュー ツリーが表示されます。
3. Silk Performer によりメッセージが表示された場合は、負荷テストに関係のあるファイルをすべてロードするかどうかを指定します。

特定の結果

負荷テスト中に記録された一般的な測定値を調査した後で、より特殊な測定値 (テスト中のサーバーまたはアプリケーションのタイプに典型的なもの) を調べることができます。たとえば、Web アプリケーションの負荷テストを行う場合は、サーバーへの正常な接続の数および HTTP ヒットと同時接続の数を調べます。データベース アプリケーションの場合は、データベースへのログオン、アクティブなデータベース カーソル、および実行された SQL 文の数が特に重要になります。

グラフに特定の結果を表示する

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。

1. ワークフロー バーの **グラフの選択** をクリックします。 **テンプレート** ダイアログ ボックスが開きます。
2. **テンプレート** 領域で、表示する測定値に対応するテンプレートをクリックします。
デフォルトのグラフを表示するには、**一般** をクリックします。
3. **終了** をクリックします。 選択した測定値を表示するグラフが開きます。

レポートに特定の結果を表示する

この作業を行う前に、レポートのデータ 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。

1. ワークフロー バーの **レポートの選択** をクリックします。 **テンプレート** ダイアログ ボックスが開きます。
2. **テンプレート** 領域で、表示する測定値に対応するテンプレートをクリックします。
デフォルトのレポートを表示するには、**一般** をクリックします。
3. **終了** をクリックします。 選択した測定値に関する情報を表示するレポートが開きます。

カスタム結果

アプリケーションのパフォーマンスに関する特定の情報を詳細に表示するために、グラフやレポートに表示する測定値を指定してカスタム グラフやカスタム レポートを作成します。

カスタム グラフを生成する

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。

1. **ファイル > 新規作成 > グラフ** をクリックします。新しいグラフが開きます。
2. メニュー ツリーで、グラフに追加する測定値タイプを含むフォルダを開きます。
3. 必要な測定値をグラフまでドラッグします。

4. このプロセスを必要な回数だけ繰り返して、グラフに要素を追加します。
5. 省略可能:測定値の限界を **しきい値の自動生成** ダイアログ ボックスで定義した場合、またはスクリプト内で直接定義した場合には、これらの測定値の限界をグラフで表示できます。 監視リスト内の測定値を右クリックし、 **測定値の限界の表示** をクリックします。

カスタム レポートを生成する

この作業を行う前に、レポートのデータ 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。

1. **ファイル > 新規作成 > レポート** をクリックします。 Performance Explorer により、選択されたファイル内のすべての測定値を含むメニュー ツリーが表示されます。
2. メニュー ツリーで、レポートに追加する測定値タイプを含むフォルダを開きます。
3. 必要な測定値をレポートまでドラッグします。
4. この操作を必要な回数だけ繰り返して、レポートに要素を追加します。


測定値の限界を表示する

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。


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 ボックスで、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。

グラフ内で測定値の限界を表示できるようにするには、次の手順を実行します。

1. **ファイル > 新規作成 > グラフ** をクリックします。 新しいグラフが開きます。
2. メニュー ツリーで、グラフに追加する測定値タイプを含むフォルダを開きます。
3. 測定値をグラフにドラッグします。

 **注:** グラフィカルな限界によって表示できる測定値に関しては、以下を参照してください。

4. 監視リスト内の測定値を右クリックし、 **測定値の限界の表示** を選択します。 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。 監視リストの **限界** 列には、グラフィカルな上限で表示できる測定値が表示されます。

現在、Performance Explorer には、次の測定値の **平均**、**最大**、および **最小** の各値の上限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

複数の負荷テストから得られた結果の表示

Performance Explorer では、任意の数の負荷テストの結果を組み合わせることで単一のグラフまたはレポートにすることができ、それによって、複数の負荷テストを比較できます。 たとえば、サーバーのパフォーマンスを微調整する前に行われた以前のテストの結果と、最近のテストの結果を一緒に表示できます。

複数の負荷テストから得られた結果をグラフにまとめる

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。

1. **ファイル > 新規作成 > グラフ** をクリックします。新しいグラフが開きます。
2. **検討 > 結果の追加** を選択します。
代替方法:メニュー ツリーでプロジェクト名を右クリックし、**結果の追加** を選択します。
結果ファイルの選択 ダイアログ ボックスが開きます。
3. カスタム グラフで使用するデータが含まれている記録済みの負荷テストを選択して、**開く** をクリックします。
Shift キーや Ctrl キーを押しながらクリックすると、リストから複数のファイルを選択できます。
選択された TSD ファイルが、メニュー ツリー内の使用可能なリソースに追加されます。
4. メニュー ツリーで、グラフで使用する負荷テストおよび測定値タイプを含むフォルダを展開します。
5. 必要な測定値をグラフまでドラッグします。

複数の負荷テストから得られた結果をレポートにまとめる

この作業を行う前に、レポート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。

1. **ファイル > 新規作成 > レポート** をクリックします。新しいレポートが開きます。
2. **検討 > 結果の追加** を選択します。
代替方法:メニュー ツリーでプロジェクト名を右クリックし、**結果の追加** を選択します。
結果ファイルの選択 ダイアログ ボックスが開きます。
3. カスタム レポートで使用するデータが含まれている記録済みの負荷テストを選択して、**開く** をクリックします。
Shift キーや Ctrl キーを押しながらクリックすると、リストから複数のファイルを選択できます。
選択された TSD ファイルが、メニュー ツリー内の使用可能なリソースに追加されます。
4. メニュー ツリーで、レポートで使用する負荷テストおよび測定値タイプを含むフォルダを展開します。
5. 必要な測定値をレポートまでドラッグします。

地域比較レポート

地域比較レポートでは、クラウド ベースの負荷テスト測定値が、地理的な地域全体で比較されます。1 列ごとに 1 地域で、異なる地域の結果が並べられます。

全体の結果と地域固有の結果の比較

地域比較レポートの設定には、非常に多くのオプションを使用できます。合計テスト結果には、すべての地域からの測定値結果が含まれますが、地域固有の測定値結果を並べて比較することができます。また、地域比較レポートは、個々の地域結果から選択した組み合わせのデータのみを含めるよう、設定することもできます。

(**検討** タブから個々の地域ノードをドラッグ アンド ドロップすることによって) 追加の地域結果を既存のレポートに追加することができ、また、(**クライアントの測定値** ノードをレポートにドラッグ アンド ドロップすることによって) 合計の地域結果を既存のレポートに追加することもできます。

ベースライン ペイン

(各地域比較レポートの下部にある) **ベースライン** ペインには、**地域** 列が含まれており、その列には、特定の地域の名前が含まれるか、または、行にすべてのテスト結果が含まれている場合は、**合計** が含まれます。**エージェント** 列には、特定の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。

いずれの地域も、他の地域の結果が測定されるベースラインとして定義できます。

地域比較レポートの生成

この作業を行う前に、複数の地域からのエージェントを使用する Silk Performer で、クラウド ベースの負荷テストを実行する必要があります。

1. クラウド ベースの負荷テストの後で、**負荷テストの要約** ページの **負荷テストの結果** グラフをクリックして、Performance Explorer を開きます。
時系列データが Performance Explorer にロードされ、**検討** ペインで使用できるようになります。

マージされた測定値/時系列データは、測定値/時系列データがプロジェクト エージェントに提供されるのと同じように、負荷テストに含まれる各クラウド地域 (たとえば、EU-Ireland、US-West) に使用できます。テストに含まれていたすべてのプロジェクト エージェント (Silk Performer **プロジェクト ツリーのエージェント** ノードの下に表示されているエージェント) からの結果が格納されている **プロジェクト エージェント** ノードを使用できるので、特定の地域ベースの結果と対照してプロジェクト エージェントの結果を評価できます。

2. テスト結果ノードと **クライアントの測定値** ノードを展開します。
3. **地域** ノードを右クリックして、**地域比較レポート** を選択します。
代替方法：ツールバーの **地域比較レポートの新規作成** ボタンをクリックします。
地域比較レポートが生成されます。このレポートは、相互負荷テスト レポートに似た形式で、1 地域に 1 列使用してすべての地域の測定値が並べられます。最大 6 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。
4. 省略可能：1 つの地域ノードを右クリックして、**地域比較レポート** を選択し、1 つの地域に関するレポートを生成することもできます。その後追加の地域ノードを **検討** タブからレポートにドラッグ アンド ドロップして、地域の結果を動的にマージし、並べて比較できます。
レポートに組み込むことができるのは、同じ負荷テストからの地域だけです。
5. 省略可能：1 地域の結果をレポートのベースラインとして定義し、他の地域の結果をそのベースライン地域と比較するには、(レポートの下部にあるベースライン ペインで) **地域エントリ** を右クリックし、**ベースラインとして設定** を選択します。

レポート順位では、ベースライン地域は常に 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。温度分布設定の詳細を表示するには、カーソルを温度分布グラフの上に置きます。

負荷テスト結果とサーバー監視データの表示

Performance Explorer では、サーバー監視データと負荷テスト結果を組み合わせて、単一のグラフまたはレポートにすることができます。それには、サーバーの負荷テストを行いながら、監視データを TSD ファイルに書き込みます。負荷テストが完了したら、負荷テスト結果を保存します。その後、監視データが含まれている TSD ファイルを、負荷テスト レポートが含まれているグラフまたはレポートにインポートします。

負荷テスト結果とサーバー監視データを単一のグラフに表示する

1. **ファイル > 新規作成 > グラフ** をクリックします。新しいグラフが開きます。
2. **検討 > 結果の追加** を選択します。
代替方法: メニュー ツリーでプロジェクト名を右クリックし、**結果の追加** を選択します。
結果ファイルの選択 ダイアログ ボックスが開きます。
3. カスタム グラフで使用するデータが含まれている記録済みの負荷テストを選択して、**開く** をクリックします。
Shift キーや Ctrl キーを押しながらクリックすると、リストから複数のファイルを選択できます。
選択された TSD ファイルが、メニュー ツリー内の使用可能なリソースに追加されます。

4. メニュー ツリーで、グラフで使用する負荷テストおよび測定値タイプを含むフォルダを展開します。
5. 必要な測定値をグラフまでドラッグします。

負荷テスト結果とサーバー監視データをレポートに表示する

1. **ファイル > 新規作成 > レポート** をクリックします。新しいレポートが開きます。
2. **検討 > 結果の追加** を選択します。
代替方法:メニュー ツリーでプロジェクト名を右クリックし、**結果の追加** を選択します。
結果ファイルの選択 ダイアログ ボックスが開きます。
3. カスタム レポートで使用するデータが含まれている記録済みの負荷テストを選択して、**開く** をクリックします。
Shift キーや Ctrl キーを押しながらクリックすると、リストから複数のファイルを選択できます。
選択された TSD ファイルが、メニュー ツリー内の使用可能なリソースに追加されます。
4. メニュー ツリーで、レポートで使用する負荷テストおよび測定値タイプを含むフォルダを展開します。
5. 必要な測定値をグラフまでドラッグします。
6. 必要な測定値をレポートまでドラッグします。

Web ブラウザでの結果の表示

Performance Explorer では、テスト結果、グラフ、およびレポートを HTML ドキュメントに変換できません。この機能により、グラフおよびレポート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 ブラウザに結果グラフを表示する

Web ブラウザに表示するグラフを作成するか、開きます。

1. ワークフロー バーの **HTML として表示** をクリックします。 **HTML として表示** ダイアログ ボックスが開きます。
2. **タイトル** ボックスにタイトルを入力します。
3. **説明** ボックスに説明を入力します。
4. **OK** をクリックします。 Web ブラウザが開いて、グラフが表示されます。 Web で公開する準備ができました。

Web ブラウザに結果レポートを表示する

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. ワークフロー バーの **HTML として表示** をクリックします。 **HTML として表示** ダイアログ ボックスが開きます。
2. **タイトル** ボックスにタイトルを入力します。
3. **説明** ボックスに説明を入力します。
4. **OK** をクリックします。 Web ブラウザが開いて、レポートが表示されます。 Web で公開する準備ができました。

HTML レポートをエクスポートする

Performance Explorer グラフを利用して、プロジェクトの状態およびエラー状況を視覚的に情報交換できます。Performance Explorer にアクセスできない場合、Performance Explorer では、簡単に電子メールを送信できる Web アーカイブ (MHT ファイル) として、すべての HTML レポートをエクスポートできます。

1. Performance Explorer にロードされたグラフのアーカイブを作成するには、次の手順のいずれかを実行します。

- **ファイル > ビューのエクスポート** を選択します。
- ツールバーの **ビューのエクスポート** をクリックします。


形式を指定してエクスポート ダイアログ ボックスが開きます。

2. アーカイブを保存するディレクトリを参照して指定します。
3. **ファイル名** テキスト ボックスに、アーカイブの名前を入力します。
4. **ファイルの種類** リスト ボックスから、**Web アーカイブ (.mht)** を選択します。
5. **保存** をクリックして、Web アーカイブ レポートを生成します。

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
```

時系列設定をカスタマイズする


TSD とリアルタイム時系列は、ユーザーのニーズに合わせてカスタマイズできます。


1. **設定 > オプション** を選択します。Silk Performance Explorer の**オプション** ダイアログ ボックスが開きます。
2. **時系列** タブをクリックします。
3. 省略可能:**NULL 値を表示する** チェック ボックスをオフにして、デフォルトで NULL 値を非表示にします。

 **ヒント:** 時系列ごとに個別に NULL 値を表示するかどうかを指定するには、ツールバーの **NULL 値の表示** をクリックします。

4. 時系列データを表示する時間の単位を **要約とカウンタ値の表示間隔** リスト ボックスから選択します。

- 間隔 を選択すると、データの表示間隔のデフォルト計算値 (5 秒ごと) を設定します。
- 秒 を選択すると、データを 1 秒間隔で表示します。

 **ヒント:** デフォルトの計算間隔を変更するには、**監視** タブの **間隔** テキスト ボックスを使用します。ツールバーの **時系列の正規化** をクリックすると、1 秒間隔で正規化したデータを時系列ごとに個別に表示できます。

 **注:** 時系列データを正規化できるのは、ソース タイプがカウンタまたは概要であるデータの場合だけです。

5. **デフォルトの線幅** リスト ボックスから、新しい時系列をグラフに追加するときに使用する線幅を選択します。
6. 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、**マーカー 領域の 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。
7. **数値の桁数** テキスト ボックスに、グラフ マーカーに表示する数字の桁数を指定します。
8. **OK** をクリックします。


根本原因の分析


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。自動結果相関は、逆方向にも使えます。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ができます。

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。

次のいずれかのオプションを選択します。

- **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 関連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。

ファイルのマージ

マージウィザードは **ウィザード** メニューから使用することができます。 **マージ** ウィザード と **カスタム マージ** ウィザード は似ており、どちらも、複数の時系列データ (TSD) ファイルを単一の TSD ファイルにマージします。

再マージ ウィザード では、間隔が元のファイルと異なる新しい TSD ファイルを作成します。

プロジェクトの TSD ファイルを単一の概要レポートにマージする

マージ ウィザード は、負荷テストから得られた複数の時系列データ (TSD) ファイルを、元のファイルに含まれるすべての測定値を含む単一の TSD ファイルにマージします。

1. **ウィザード** > **マージ ウィザード** を選択します。 **プロジェクトの選択** ダイアログ ボックスが開きます。
2. 時系列データをマージする対象の負荷テストを含むプロジェクトを選択し、 **次へ** をクリックします。 **負荷テストの選択** ダイアログ ボックスが開きます。
3. 単一の出力先ファイルにマージする負荷テストを選択して、 **次へ** をクリックします。 **ファイルの選択** ダイアログ ボックスが開きます。
4. マージする時系列データ ファイルを選択し、 **次へ** をクリックします。 **詳細マージ プロパティ** ダイアログ ボックスが開きます。
5. **出力間隔** ボックスに、時系列データの各マージ間隔の秒数を指定します。
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。
6. 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。
7. 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。
8. **次へ** または **完了** をクリックすると、マージ処理が開始されます。

マージが完了すると、概要レポートが生成されて表示されます。

ディレクトリの TSD ファイルを単一の概要レポートにマージする

カスタム マージ ウィザード は、負荷テストから得られた複数の時系列データ (TSD) ファイルを、元のファイルに含まれるすべての測定値を含む単一の TSD ファイルにマージします。

1. **ウィザード** > **カスタム マージ ウィザード** を選択します。 **ディレクトリの選択** ダイアログ ボックスが開きます。
2. **ソース ディレクトリ** テキスト ボックスで、マージした時系列データ ファイルを格納する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、 **フォルダの参照** ダイアログ ボックスを開き、場所を選択します。
3. **ターゲット ファイル** テキスト ボックスで、マージされた TSD ファイルをインクルードするファイル名を入力します。

4. **次へ** をクリックします。 **ファイルの選択** ダイアログ ボックスが開きます。
5. マージされる TSD ファイルのターゲット ファイル名と負荷テスト ディレクトリを選択します。
ファイル名は m@@<TARGETFILENAME>.tsd という形式です。
6. 単一の出力先ファイルにマージする TSD ファイルを負荷テスト ディレクトリから選択して、**次へ** をクリックします。 **詳細マージプロパティ** ダイアログ ボックスが開きます。
7. **出力間隔** ボックスに、時系列データの各マージ間隔の秒数を指定します。
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。
8. 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。
9. 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。
- 10 **次へ** または **完了** をクリックすると、マージ処理が開始されます。

マージが完了すると、概要レポートが生成されて表示されます。

Remerging a Single Time-Series Data File

The **Remerge** wizard allows you to remerge a single time-series data (TSD) file with advanced options, such as a different interval or time frame. Increasing the output interval is useful to make result files smaller in size and to smoothen the corresponding graphs.

1. Choose **Wizard > Remerge Wizard** . The **File Selection** dialog box opens.
2. Select the source and destination TSD file and click **Next**.
By default, the destination file possesses the same file name as the source with `_r` appended to the name.

The **Advanced Merging Properties** dialog box opens.
3. **出力間隔** ボックスに、時系列データの各マージ間隔の秒数を指定します。
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. 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。
6. **次へ** または **完了** をクリックすると、マージ処理が開始されます。

マージが完了すると、概要レポートが生成されて表示されます。

時系列ファイルの種類

時系列ファイルの種類	説明
マージされたファイル	マージされたファイルは m@... で始まり、すべて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。これらは負荷テスト コントローラで生成されます。
中間ファイル	中間ファイルは I@... で始まり、負荷テスト エージェントによって生成されます。これには、1 つのエージェント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すべての仮想ユーザーから得られたデータを含む) が含まれています。
エージェント ファイル	エージェント ファイルは a@... で始まり、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、これらのエージェント ファイルはエージェント上で生成された後、すべてのユーザー ファイルと共に中間ファイルにマージされます。
ユーザー ファイル	ユーザー ファイルは u@... で始まり、1 人の仮想ユーザー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、これらのユーザー ファイルはエージェント上で生成された後、その他のすべてのユーザー ファイルと共に中間ファイルにマージされます。
リアルタイム ファイル	リアルタイム ファイルは r@... で始まり、サーバー側監視によって生成された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	ユーザー タイプごとのエージェント ファイルは t@... で始まり、1 つのエージェントから得られた特定のユーザーの時系列データが含まれています。
ユーザー タイプごとのマージされたファイル	ユーザー タイプごとのマージされたファイルは k@... で始まり、すべてのエージェントから得られた特定のユーザーに対する時系列データが含まれています。これらは負荷テスト コントローラで生成されます。
領域ファイル	領域ファイルは n@... で始まり、特定の領域に対する時系列データが含まれています。



注:

```
I@ = Merge(a@, u@_1, ..., u@_n)
m@ = Merge(I@_1, ..., I@_m)
users: 1 - n
agents: 1 - m
```

コントローラがエージェントとの接続を失った場合にファイルをマージする

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 コントローラの中間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. 中間ファイルが格納されているディレクトリにすべてのファイルをコピーします。
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. **終了** 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。

Merging Load Test Results

To run a big load test, it can be necessary to start the load test from multiple controllers. Consequently, Silk Performer will generate multiple result folders (one folder per controller). Use the wizard to merge the result folders into one result folder. This allows you to view all metrics in a single comprehensive report in Performance Explorer.

 **注:** Results from different projects cannot be merged.

1. Choose **Wizard > Merge Load Test Results Wizard**.
2. Drag two or more load test result folders from a file explorer onto the table or click **Add** to browse for result folders.
3. Specify a master load test in the **Master** column. Some values from the result files cannot be merged (like the project description). In such a case, Silk Performer will use the value of the master load test.
4. Select a **Merge type**:
 - **absolute**: In a report, the results will display with their actual start time on the time line. For example: If you started a load test on Controller B 10 seconds after you started the load test on Controller A, the results will display on the time line accordingly.
 - **relative**: In a report, all results will display with the same start time on the time line, regardless of the actual start times of the load tests.
5. Specify an **Interval** for the merged time-series data files. The minimum interval (shown in the **Interval** list) is the lowest common multiple of the intervals from all time series data files that are located in the specified results folders.
6. Specify a **Target folder**. The merged results will be stored in this folder.
7. Click **Merge**.


When the results are merged successfully, Performance Explorer displays an overview report. This newly created report has the workload type **Merged Workload**, which contains all executed usergroups.

CSV ファイルのエクスポート

CSV 形式で結果をエクスポートし、表計算プログラムで (オフライン) 分析します。

CSV 形式で結果をエクスポートする

Performance Explorer の **エクスポート** ウィザード を使用すると、表計算プログラム (Microsoft Excel など) でオフライン分析するために、時系列データ ファイル (TSD) を簡単にカンマ区切りファイル (CSV) にエクスポートできます。

 **注:** Silk Performer では、**エクスポート** ウィザード の機能は、TSD File Export コマンドライン ツールを使って自動化できます。

1. **ウィザード > エクスポート ウィザード** を選択します。 **ファイルの選択** ページが開きます。
2. **ソース ファイル** テキスト ボックスの右にある参照 (...) ボタンをクリックして、エクスポートする TSD ファイルを選択します。

デフォルトでは、**エクスポート ウィザード** は TSD ファイルの名前を CSV ファイルの名前として使用します。

3. **エクスポート ファイル** テキスト ボックスの右にある参照 (...) ボタンをクリックして、作成する CSV ファイルをエクスポートする場所と名前を指定します。

デフォルトでは、**エクスポート ウィザード** は TSD ファイルの場所を CSV ファイルの場所として使用します。

4. **次へ** をクリックします。**時系列の選択** ダイアログが開きます。ここで、エクスポートする測定値を指定できます。

5. 出力ファイルにエクスポートする測定値を選択します。

少なくとも 1 つの測定値を選択する必要があります。すべての測定値を選択するには、**すべて選択** を、測定値の選択をすべて解除するには、**すべてクリア** を使用します。

6. **次へ** をクリックします。**オプションの選択** ページが開きます。

7. **エクスポート タイプ** 領域で、以下のいずれかの手順を実行します。

- **標準** をクリックして、CSV ファイルに追加す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** をクリックして、すべての生データを CSV ファイルに含めます。

8. **データ ソース** 領域で、エクスポートする要素タイプに対するチェックボックスをオンにします。

9. **エクスポート オプション** 領域で **ヘッダー情報を含める** チェックボックスをオンにして、CSV ファイルにヘッダー情報を含めます。

- 10 **小数点** リスト ボックスからオプションを選択して、小数点を表す方法を定義します。

- 11 **完了** をクリックして、指定に基づいて CSV ファイルを生成します。

- 12 出力ファイルを表示する場合は、**エクスポートが完了しました!** ダイアログ ボックスで **はい** をクリックします。表示しない場合は、**いいえ** をクリックして、出力ファイルを開かずに **エクスポート ウィザード** を閉じます。

たとえば、Microsoft Excel を起動する場合、**はい** をクリックします。

コマンドライン ツールで CSV ファイルのエクスポートを自動化する

TSD (時系列データ) エクスポート ファイル ツール は、TSD ファイルから CSV 形式のファイルへのエクスポートを自動化できるコマンドライン ツールです。このコマンドライン ツールには、**エクスポート ウィザード** と同じ機能があります (GUI の使いやすさは除く)。

1. **スタート > プログラム > Silk > Silk Performer <バージョン番号> > 分析ツール > TSD File Export Tool** を選択します。

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]
```

使用できるコマンド	説明
tsdFiles	エクスポートする TSD ファイルの名前を入力する。
[csvFile]	作成する CSV ファイルの名前を入力する。
[-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. Return または Enter キーを押して、出力ファイルを生成します。

例

```
Tsd2Csv "the.tsd" "the.csv" -Delimiter "," -DecimalPoint "."
```

グラフを比較する

対象とする負荷テストの測定値を識別した後に、同じ測定値タイプの別の負荷テストを調べると便利です。Performance Explorer には、ファイルを簡単に比較できる機能があります。選択されたグラフと比較する時系列データ (TSD) ファイルを選択すると、Performance Explorer は現在のグラフで表示されている測定値タイプのみを使用した新しいグラフを作成します。

1. 比較する測定値を含むグラフを開き、ワークフロー バーの **グラフの比較** をクリックします。 **TSD ファイルの比較** ダイアログ ボックスが開きます。
2. アクティブな結果と比較する、記録された負荷テストから、TSD ファイルを見つけ、**開く** をクリックします。利用できるすべての一致する要素を使った新しいグラフが開きます。
3. 省略可能:比較しやすいようにグラフのサイズや配置を変えます。

エラー分析

すべてのエラーをグラフィックで表示し、詳細情報を見つけるために詳しく調べます。

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ユーティリティがあります。概要レポートではエラーについての情報が一部表示されますが、[エラーの分析] 機能ではすべてのエラーがグラフィックで表示され、詳細な情報まで掘り下げることができます。

1. Silk Performer の 負荷テストが完了した後に、次の手順のいずれかを実行します。
 - ワークフロー バーの **エラーの分析** をクリックします。
 - **ウィザード > エラーの分析** を選択します。
 - **検討** タブをクリックして、結果のペインを右クリックして、**エラーの分析** を選択します。
- エラー分析のソース** ダイアログ ボックスが開きます。
2. **ファイル名** テキスト ボックスで、調べる負荷テスト結果に対する TSD ファイル名を入力します。
代替方法:TSD ファイルを探す場合、**参照 (...)** ボタンをクリックします。
 3. **マージされた時系列ファイル (m@*.tsd)** が **ファイルの種類** テキスト ボックスで指定されていることを確認します。
 4. **開く** をクリックします。
 5. **終了** をクリックします。エラー グラフが開きます。Performance Explorer の標準ズーム機能で、特定の必要な情報に掘り下げることができます。

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	エラー グループ 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	エラーの詳細 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 TrueLog の表示 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。

表示	説明
凡例	凡例 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 値 列が更新されます。

Silk Central と Performance Explorer のインテグレーション

Performance Explorer と Silk Central のインテグレーションの結果、Silk Central を通じて起動される Silk Performer テスト実行の詳細な分析に Performance Explorer を使用できるようになります。Performance Explorer は Silk Central から直接起動することができ、結果パッケージは Silk Central と Performance Explorer の間で容易に転送することができます。最新の Silk Performer 負荷テスト実行 (最近行った TryScript 実行を除く) の結果を Silk Central にアップロードして、Silk Central のテスト定義と関連付けることができます。

Performance Explorer では、関係のある Silk Central テスト結果の取得に役立つ **Silk Central の結果の追加** アイコンがツールバーにあり、テスト結果の統合と整理に役立つ **Silk Central** タブが結果ペインにあります。

Silk Central から結果を追加する

1. Performance Explorer で、**Silk Central** タブをクリックし、次に **Silk Central の結果を追加するには、ここをクリックします** リンクをクリックします。

代替方法： ツールバーの **Silk Central の結果の追加** をクリックします。

Silk Central **の結果を開く** ダイアログ ボックスが開きます。

2. **ホスト名** テキスト ボックスには、接続する Silk Central ホストを指定します。
3. **ポート** テキスト ボックスには、Silk Central への接続に使用するポートを指定します。
4. **ユーザー名** テキスト ボックスには、Silk Central へのアクセス権を持つユーザーを指定します。
5. **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。
6. **次へ** をクリックします。

Silk Performer から Silk Central へのアクセスが既に構成済みであれば、設定値が自動的に入ります。

7. Silk Performer プロジェクト リストから、Silk Central の結果を開くプロジェクトを選択します。
8. **次へ** をクリックします。
9. テスト定義メニュー ツリーから、Silk Performer の結果を開くテスト定義を選択します。
- 10 **次へ** をクリックします。 **実行の選択** ページが開きます。
- 11 結果を表示したいテスト定義の実行を選択します。

ここ [] 日間のテスト実行 ボックスを使用して、過去何日間の実行を **実行** リストの定義に含めるか指定できます。

- 12 **完了** をクリックして、選択した実行結果を Performance Explorer に取り込みます。Performance Explorer の **Silk Central** ページには、Silk Central GUI の結果ファイルへのパスが表示され、概要レポートが自動的に生成されます。

オンデマンドで利用できるダウンロード

Silk Central ページは、ファイルのダウンロードをオンデマンドで提供します。測定値結果ファイルは、表示のために Performance Explorer で開くよう明示的に要求するまで、Silk Central データベースに残っています。

1. Performance Explorer の **Silk Central** タブのファイル メニュー ツリーのノードを参照します。
2. 測定値をダブルクリックして、Performance Explorer で表示するために開きます。

相互負荷テスト レポート

相互負荷テスト レポートを使用すると、テスト実行の結果を最大 4 件まで並べて比較できます。相互負荷テスト レポートは、標準ベースライン レポートと同様の構造になっており、トランザクション、ページとアクション タイマ、Web フォーム、順位、エラーに関する要約レポートと統計データを提供します。

相互負荷テスト レポートを生成する

相互負荷テスト レポートを生成し、テスト実行の結果を 4 つまで並べて比較します。

表示順に関しては、ベースライン テストは常に 1 番目にリストされます。

1. Performance Explorer 内で、ツールバーの **相互負荷テスト レポートの新規作成** ボタンをクリックします。 **相互負荷テスト レポート** ウィンドウが開きます。

2. 最大 4 つのテストを開きます。

新しいレポートのタイトルには、それぞれ 1 から 4 の一意となる数字が割り当てられます。

開いたそれぞれの新しい負荷テスト実行が、**Silk Central** ページのその他のインポートした実行の横に並んで表示されます。

3. **検討** タブから、レポートに表示する実行を **相互負荷テスト レポート** ウィンドウにドラッグします。

代替方法: の Performance Explorer の **Silk Central** タブで、ファイル メニュー ツリーのテスト定義で右クリックし、**相互負荷テスト レポート** を選択します。選択したテスト定義のすべての実行が、相互負荷テスト レポートに自動的に追加されます。

b アイコンまたは **ベースライン** タグは、各レポートのベースライン実行を特定します。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。



注: 温度分布設定の詳細を表示するには、カーソルを温度分布グラフの上に置きます。



注: 他のテストが比較されるベースラインとして別の実行を定義するには、その実行を右クリックして、**ベースラインとして設定** を選択する必要があります。

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。

コマンドラインから Performance Explorer を起動する

次の構文を使って、コマンドライン インターフェイスから Performance Explorer を起動します。

```
perfexp [/TSD:<file>] [/ACTION:STDGRAPH]
[/ACTION:STARTALLMONITORS]
```

たとえば、次のように入力します。

```
Perfexp.exe /pew:"c:\test.pew" /monitordir:"c:\temp\perf"  
/action:startallmonitors /nogui /singleinstance /noexit
```

または、次のように入力します。

```
Perfexp.exe /pew:"c:\test.pew" /monitordir:"c:\temp\perf" /action:stopallmonitors  
/nogui /singleinstance[:nostartup] /exit
```

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コマンドの一覧を示します。

コマンド	説明
/TSD:<ファイル>	ファイルをワークスペースに追加します。
/OVT:<テンプレート>	<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	最後に指定された /TSD:<ファイル> を使用します。
/ACTION:SELGRAPH	最後に指定された /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起動します。
/ACTION:STARTALLMONITORS	すべての監視グラフを起動します。
/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>]	最後に指定された /TSD:<ファイル> の概要レポートを生成します。オプションでテンプレート ファイルを適用できます。
/EXPORTOVR:<ファイル>	最近生成した OVR 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>	指定され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:<コマンドファイル>	指定された 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。

コマンド	説明
/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 を開き、使用できるコマンドラインパラメータを表示します。

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。

索引

記号

BEA WebLogic
監視する 47, 48

B

BDL 監視プロジェクト
Performance Explorer での 67
新しいプロジェクトを作成する 58
概要 54
基本ユース ケース 58
高度なユース ケース 62, 63
作成 55
思考時間と待機 67
初期化 68
スクリプトのパッケージ化 61
スクリプトを作成する 59, 64
その他のベスト プラクティス 68
チュートリアル 57
プロジェクト属性の計画 58, 63
プロジェクトの使用 62
ループ 67

C

CPU 使用率
エージェントの健全度 73
CSV
CSV 形式で結果をエクスポートする 102
TSD への変換 23
ファイルのエクスポートを自動化する 103

D

Diskperf
エラー メッセージ 53
コマンド 51
実行する 51, 52

G

GUI
Performance Explorer のツアー 6

H

HTML
レポートをエクスポートする 96, 97

I

IBM
DB2
監視する 28
スナップショット監視 28

パフォーマンス指標を取得する 31
WebSphere
パフォーマンス測定値 32

IIOP 49
IIS のパフォーマンス監視
トラブルシューティング 36
利用可能なパフォーマンス測定値 34

J

JMX
Bean をフィルタリングする 25
監視の前提条件 23
クエリ 25
詳細クエリ 26
接続する 23
JMX 管理サーバ 49

L

Linux コマンド 21
load test results
merging 102

M

MBean を指定する 24
MemFree 21
Merge wizard
remerge 100
MIB ファイル 22
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 36

O

Oracle
監視する 40, 42
Oracle Forms
監視する 38, 39
クライアント側パフォーマンス測定値 39
Oracle データベース監視
Oracle カウンタの追加 40
Oracle クライアントと Oracle カウンタをインストールする 40
Oracle のパフォーマンスを監視する (Perfmon) 40
既知の問題 42
パフォーマンス カウンタ 41

P

Perflib
Perflib サブキーに対する読み取りアクセス権がありません 53
サブキー 52
PerfMon 17, 52
Performance Explorer 5

R

- Remerge wizard 100
- REXEC
 - 概要 18
 - 監視する 17
- REXEC クエリ 19

S

- SAP
 - 監視する 43
- Silk Central
 - インテグレーション 105
 - 結果を追加する 105
- SNMP
 - 監視する 17
 - 検索する 21
 - データソースを追加する 21
- SNMP サポート
 - データソースの定義 46

U

- UNIX ベースのクエリ 19

W

- WebLogic (JMX) を監視する 49
- WebLogic (JMX) の監視
 - IIOP 49
 - JMX 管理サーバ 49
- Web ブラウザ
 - 結果を表示する 96
- welcomePE 5
- Windows
 - エラー 51, 53
 - リモート監視を有効にする 50, 51

あ

- アプリケーションの外見 9

い

- インテグレーション
 - Silk Central 9

え

- エージェント
 - エージェントの健全度
 - 応答性 73
 - 健全度 73
 - ファイル 101
- エラー
 - SYSTEM:5 - アクセスが拒否されました 54
 - Windows 51, 53

- 解決ポータル 53
- プロキシ サーバー 53
- 分析する 104
- エラー分析グラフ 104

お

- 応答性
 - エージェントの健全度 73
- オンデマンドによるダウンロード 106

か

- 概要レポート
 - 概要 87
 - カスタマイズする 89
 - グラフ 90
 - グラフを追加する 89
 - 構造 87
 - 削除する 90
 - テキスト領域 91
 - デフォルトのグラフ 89
 - デフォルトのレポート 90
 - テンプレート 85, 91
 - 表示する 89
 - 分析する 8
 - レポートを追加する 90
- カスタムレポート
 - 生成する 93
- 監視
 - グラフ 70
 - 種類 70
- 監視する
 - IBM WebSphere Application Server 32
 - インターフェイス 17
 - サーバー 27
 - 自動 27
 - 設定 68
 - デフォルトのオプションを設定する 82
 - リモート マシン 50
- 監視レポート 8

<

- クエリ
 - JMX 26
 - REXEC 19
 - UNIX 19
 - 実行する 26
- クライアント側測定値
 - 構成する 69
 - 削除する 69
- グラフ
 - Web ブラウザで表示する 96
 - エラー分析 104
 - カスタム生成する 92
 - 監視する 70
 - 監視レポートを作成する 70
 - グラフ 8
 - 結果を表示する 92

- 結果をまとめる 94
- 説明を変更する 91
- デフォルト オプション 80
- デフォルトのオプションを設定する 80, 81
- 名前を変更する 91
- 比較する 104
- 表示をカスタマイズする 83
- 負荷テストを監視する 95
- プロパティ 83
- ローソク オプションを設定する 82

け

結果

- エクスポートする 102
- 複数の負荷テストを表示する 93

結果の分析 8

こ

コマンドライン インターフェイス

- 概要 106
- コマンド 107
- コマンド ファイル 108
- 製品を起動する 106

根本原因の分析 98

し

時系列

- NULL 値 97
- TSD ファイル 8, 23
- カスタマイズする 97
- グラフ マーカー 97
- 正規化する 97
- 線幅 97
- データ オプションを設定する 81
- データ結果ファイルを指定する 91
 - データ出力
 - デフォルトのオプションを設定する 82
- デフォルトの計算間隔 97
- ファイルの種類 101

システムの選択

- ダイアログ ボックス 25

実行 18

自動監視 27

せ

製品

- 概要 5
- 起動する 6
- 機能 5
- 製品を起動する 6
- セキュア シェル 44
- 接続が失われた 101
- 接続を失ったコントローラ 101
- 設定
 - 監視する 68
 - プロキシ サーバー 53

そ

相互負荷テスト レポート 106

属性プール 25

測定値

- Citrix 80
- CORBA 78
- IIOP 75
- SQL データベース 79
- TUXEDO 79
- Web 74
- Web フォーム 77
- Web ページ 78
- インターネット 73
- カスタム カウンタ 80
- カスタム タイマ 80
- グループ タイプ 72
- 全般 73
- 関連させる 98
- データベース 76
- トランザクション 76
- 名前を変更する 71
- 比較 70
- 複製する 71
- ミドルウェア 79
- 要約 72

測定値の限界

- 表示 93

測定値の比較 70

た

- ダイアログ ボックス
- システムの選択 25
- 単独実行 18

ち

- 地域比較レポート 94, 95
- 中間ファイル 101

つ

ツアー

- ユーザー インターフェイス, Performance Explorer 6

追加する

- Silk Central の結果 105
- 属性 25

て

データ ソース

- JMX 23
- PerfMon 17
- UNIX 18
- ウィザード 9
- カスタム 17
- 自動検出 15
- スキャンする 16
- 定義済み 10
- 定義する 9

データ ソース ツリー メニュー 8

データ変換 23

テスト結果
 カスタム
 表示する 92
 検討する 84
 表示する 92
テンプレート
 監視用 68
 割り当てる 85

は

パスワードの変更 72
パフォーマンス測定値
 WebSphere 32
パフォーマンス モニタ 68

ひ

表示テーマ 9

ふ

負荷テスト
 表示する 85
 監視グラフ 95
 監視レポート 96
 グラフにまとめる 94
 結果を表示する 85, 95
 レポートにまとめる 94
複数実行 18
プロキシ サーバー 53

ほ

保存したクエリ 26

ま

マージウィザード
 概要 99
 カスタマイズする 99
 使用する 99
マージされたファイル 101

め

メモリ使用率
 エージェントの健全度 73

ゆ

ユーザー インターフェイス

Performance Explorer 概要 6
ユーザー ファイル 101

り

リアルタイム
 監視する 8
 ファイル 101
リモート監視
 Windows で有効にする 51
リモート実行 18
リモート マシン
 監視の要件 50

れ

レポート
 地域比較 94
 Web ブラウザで表示する 96
 概要 8
 カスタム生成する 93
 監視 8
 監視する 70
 監視レポートを作成する 71
 結果を表示する 92
 結果をまとめる 94
 コマンドラインでエクスポートする 97
 説明を変更する 91
 相互負荷テスト 106
 相互負荷テストを生成する 106
 デフォルト オプション 80
 デフォルトのオプションを設定する 82
 テンプレート 85
 名前を変更する 91
 負荷テストを監視する 96

ろ

ログイン資格情報
 変更する 72

わ

ワークフロー バー
 Performance Explorer 7