Borland®
A MICRO FOCUS COMPANY

Silk Performance Explorer 10.0

Help

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

2013-06-18

Contents

Performance Explorer 10.0

Welcome to Performance Explorer 10.0.

Performance Explorer allows you to view measurements obtained through real-time monitoring and to analyze results of past load tests. An array of graphical tools enables both real-time monitoring of the target server while tests are running and exhaustive analysis of the results after tests have completed. Exploring test results is made easy by a workflow bar with a click-through UI, enhanced by drag functionality.

Getting Started

This section provides the information you need to monitor servers and analyze test results using Silk Performance Explorer.

Performance Explorer

Performance Explorer allows you to view measurements obtained through real-time monitoring and to analyze results of past load tests. An array of graphical tools enables both real-time monitoring of the target server while tests are running and exhaustive analysis of the results after tests have completed. Exploring test results is made easy by a workflow bar with a click-through UI, enhanced by drag functionality.

Performance Explorer Overview

With Performance Explorer real-time monitoring, live charts provide a customizable display of the most relevant performance information from the target server. Monitoring is available for a comprehensive collection of Web servers, application servers, and database servers. You can open multiple charts at the same time allowing you to watch a graphic display of Web server performance and operating system performance simultaneously. A menu tree editor with drag functionality allows you to combine elements from any data source into charts.

After a test, you can chart the performance of the target server from both the client side and the server side. Response-time measurements display the client perspective, while throughput data displays the perspective from the server side. Charts and graphs are fully customizable, and they can contain as many or as few of the measurements taken during the test as you require. You can open multiple charts, using information from one or multiple tests, at once to facilitate contrast and compare operations. Templates for the most typical test scenarios, such as Web, Database, IIOP, are provided. You can populate these default charts easily and quickly with the data you need. Drag functionality enables chart elements to be combined from any data source. You can place information on client response times and server performance in a single chart so that you can directly view the effect of server performance on client behavior.

Using the advanced drag functionality of the menu tree editor, you can combine information elements from any data source in any number of selected charts. You can even add and remove measurements from different sources to produce permanent or temporary charts and reports that suit your needs. You can save modified settings with each project to ensure that Performance Explorer always opens with your preferred view.

Performance Explorer also offers a comprehensive XML/XSL-based overview report that combines user group data from the baseline report file (`baseline.brp`) with time-series, test-result information. You can customize the overview report to your needs by changing the provided texts and by inserting charts of your interest. Saving your changes as a template allows the reuse of your individual settings.

Starting Performance Explorer

Before beginning, make sure you know how your application is set up and deployed. For example, find out which of your machines powers the database and application servers.

Perform one of the following steps:

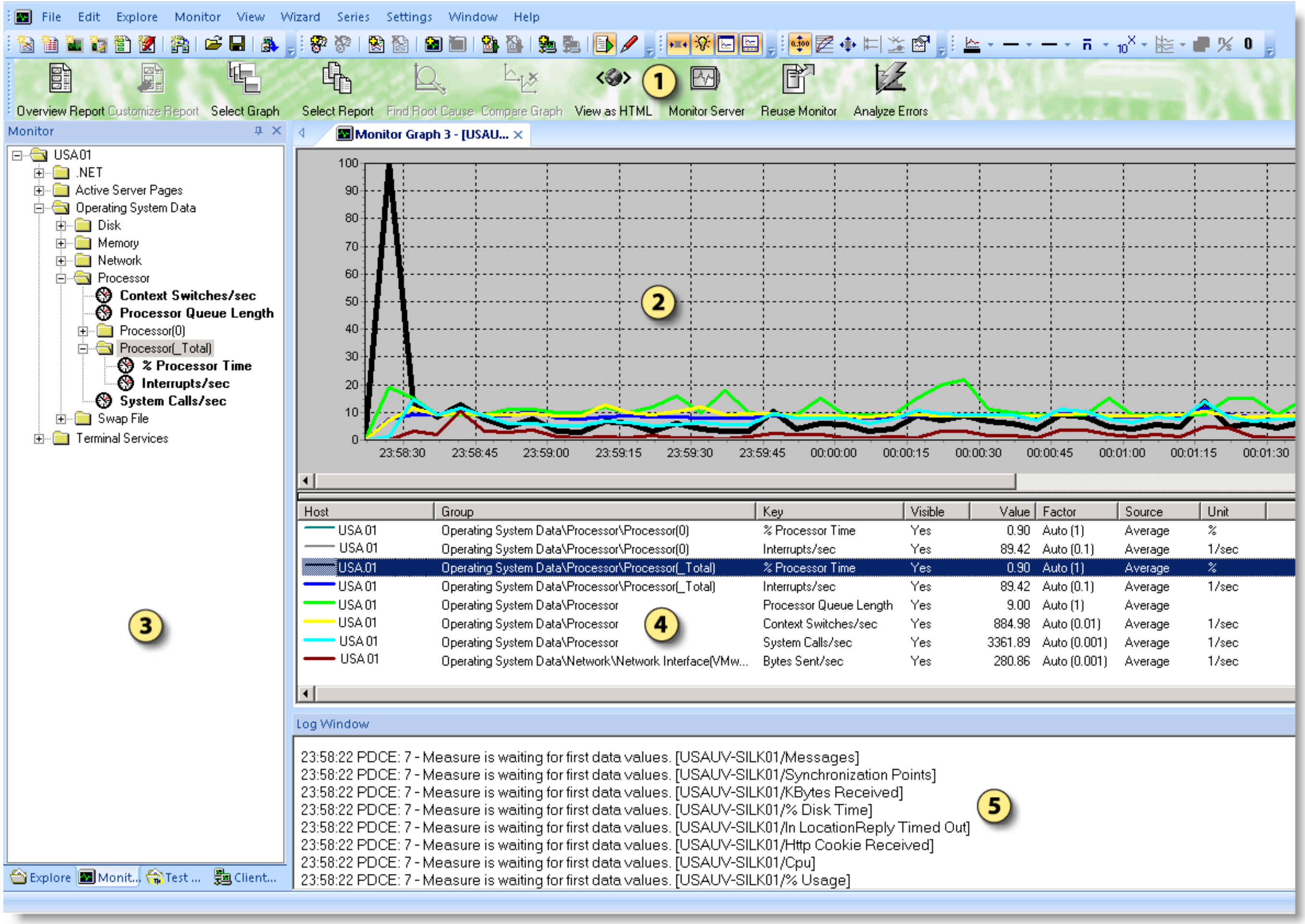
- Choose **Start** > **Programs** > **Silk** > **Silk Performer <version>** > **Analysis Tools** > **Silk Performance Explorer**.
- From Silk Performer, choose **Results** > **Monitor Server** .
- From Silk Performer, choose **Results** > **Explore Time Series** .

Note: You can also launch Performance Explorer from a command-line interface.

When editing a new template or launching Performance Explorer from the Silk Performer results menu, the program opens with several predefined monitor graphs and a predefined data source (your local machine).

Tour of the UI

The main sections of the Performance Explorer interface are highlighted below.



Host	Group	Key	Visible	Value	Factor	Source	Unit
USA 01	Operating System Data\Processor\Processor(0)	% Processor Time	Yes	0.90	Auto (1)	Average	%
USA 01	Operating System Data\Processor\Processor(0)	Interrupts/sec	Yes	89.42	Auto (0.1)	Average	1/sec
USA 01	Operating System Data\Processor\Processor(_Total)	% Processor Time	Yes	0.90	Auto (1)	Average	%
USA 01	Operating System Data\Processor\Processor(_Total)	Interrupts/sec	Yes	89.42	Auto (0.1)	Average	1/sec
USA 01	Operating System Data\Processor	Processor Queue Length	Yes	9.00	Auto (1)	Average	
USA 01	Operating System Data\Processor	Context Switches/sec	Yes	884.98	Auto (0.01)	Average	1/sec
USA 01	Operating System Data\Processor	System Calls/sec	Yes	3361.89	Auto (0.001)	Average	1/sec
USA 01	Operating System Data\Network\Network Interface(VMw...	Bytes Sent/sec	Yes	280.86	Auto (0.001)	Average	1/sec

```
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/Messages]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/Synchronization Points]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/KBytes Received]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/% Disk Time]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/In LocationReply Timed Out]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/Http Cookie Received]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/Cpu]
23:58:22 PDCE: 7 - Measure is waiting for first data values. [USAUV-SILK01/% Usage]
```

1 **Workflow bar**

2 **Graph/Report view**

3 **Tree menu**

4 **Monitor list**

5 **Log window**

Workflow Bar

The workflow bar facilitates the primary actions related to monitoring server performance. The buttons on the workflow bar enable you to perform the following commands:

- **Overview Report**: Generates an overview report, which includes all significant load test results, including general information, user types, charts, and data tables.
- **Customize Report**: Enables you to customize the entries that appear in the overview report template.
- **Select Graph**: Enables you to select a graph template for viewing time-series data (TSD files).
- **Select Report**: Enables you to select a report template for viewing time-series data (TSD files).
- **Find Root Cause**: Enables you to correlate client-side performance issues with server-side measurements.
- **Compare Graph**: Enables you to examine the same measurement type across multiple load tests.
- **View as HTML**: Generates report output in HTML for web-browser viewing.
- **Monitor Server**: Enables you to specify a data source for analysis, or have available data sources detected for you automatically.
- **Reuse Monitor**: Enables you to edit a performance monitor using Silk Performer or to export a monitor for use within Silk Central .
- **Analyze Errors**: Displays all of a load test's errors in a graphical format that allows you to drill down to find detailed information.

Graph/Report View

Server-monitoring data and load-test results are displayed here within graphs and reports.

Performance Explorer displays graphs and reports in a standard format but offers numerous possibilities for customization. You can change colors, line width, and line styles that are used to display measurements. You can also change scaling options and insert or remove markers.

Tree Menu

The tree menu offers multiple views. Using the view tabs, you can browse for test results to analyze, add data sources, download test results from Silk Central, or browse through available performance monitors. Available monitors cover a wide range of server performance metrics and operating system data.

Monitor List

This view lists all active real-time performance monitors and offline time series data (TSD) performance measurements. Key properties and statistics for each monitor and measure are listed in tabular format.

Log Window

This view lists all events that occur during performance monitoring, including errors and query statuses.

Real-Time Monitoring

Performance Explorer supports real-time server-side and client-side monitoring.

Server Monitoring

When used for real-time monitoring, Performance Explorer automatically provides you with one or more live graphical charts that display a default selection of relevant target-server performance statistics.

Performance Explorer incorporates automatic monitoring functionality for a comprehensive range of Web servers, application servers, database servers, and clients that run on commonly used operating systems.

Client-Side Monitoring
Use real-time client-side measures like any other monitoring measure, such as, built-in graphs and monitor reports.

Note: In some instances, virtual users might fail or not start during load-testing and monitoring of client-side real-time measures. The message `SYSTEM: 231 All pipe instances are busy` appears. This error is caused by real-time virus protection interfering with or blocking real-time monitoring. Named pipes are recognized as files and then scanned by virus-protection software.

Results Analysis

Create monitor reports to analyze test data. Performance Explorer can write and save test data to one or more time-series data (TSD) files. At any time using TSD files, you can generate charts that display performance measurements for servers under test from both the client and server perspectives. Response-time measurements reveal the client perspective, while throughput data reveal the server-side perspective.

You can save monitor reports and add them to projects, or archive monitor reports for later use.

You can also customize graphical charts during results exploration. Charts can contain as many or as few test measurements as you require. You can open multiple charts, and add measurements from any recorded test. Use charts and measurements to visually compare select results from different tests or to compare test results to real-time monitoring data.

A comprehensive, HTML-based overview report combines user group data from the baseline report file, `baseline.brp`, with time-series, test result information. You can customize the overview report to meet your needs by editing default text and inserting charts of interest. Saving your changes as a template enables you to reuse custom settings with future reports.

Silk Central

Performance Explorer is fully integrated with Silk Central. You can download the results of Silk Performer tests that are executed using Silk Central to Performance Explorer for analysis.

Defining an Application Look

You can change the overall UI look based on different color schemes.

1. Choose **View** > **Application Look**.
2. Select the application style that you want:
   - `Blue Style`
   - `Black Style`
   - `Silver Style`
   - `Aqua Style`

The UI is then redrawn based on the style you have selected.

Data Sources

This section describes how to define data sources for Performance Explorer.

Define your servers and applications as data sources to collect performance data from them. Define data sources in one of the following ways:

- Select a server or application from a set of predefined data sources.

  Performance Explorer provides predefined data sources for the most popular Web servers, application servers, database servers, and operating systems. You are not restricted to using only these data sources. These data sources are included to speed the selection process for measurements that are to be monitored.

- Use the Data Source Scanner to detect data sources.

  The Data Source Scanner checks your system for a variety of data sources that are commonly found on a range of platforms and returns a list of available data sources.

Both methods require that you open the **Data Source Wizard**, either by using the **Monitor** menu or by right-clicking the **Monitor** pane.

Predefined Data Sources

This page lists the predefined data sources that are available with Performance Explorer and outlines specific installation and configuration requirements for each server type that Performance Explorer can monitor.

System	Installation and Configuration Requirements
.NET	.NET CLR and ASP.NET using PerfMon
Active Server Pages	Performance Explorer can monitor Active Server Pages without custom configuration. Varying counters can be displayed, depending on which Internet Information Server version is installed. • IIS 5.1/6.0 (using PerfMon)

Apache	The monitoring of Apache Web Servers is supported for the following operating systems (using rexec): HP-UX, Linux, Solaris, and OSF1.
	With Apache Web servers, you have the option of using the built-in status report functionality. This option requires that the mod_status module be built in, which is done by default. To obtain more detailed information, select the compile option **ExtendedStatus On**. To enable status reports only for browsers from the `foo.com` domain, add code to your `access.conf` configuration file. The Apache configuration is maintained in a set of files, usually located in `/usr/local/apache/etc` or `/usr/local/apache/conf` (Solaris) or `/etc/httpd/conf` (Linux). By default, the following code is commented out in one of the configuration files.
	`<Location /server-status> SetHandler server-status`
	`order deny, allow deny from all allow from.foo.com`
	`</Location>`
	If it is not, add it to the main Apache configuration file, `httpd.conf`. Ensure that the `Allow` statements are correct. Restart your server. To verify the server's performance module, type the following URL into your browser:
	`http://hostname/server-status?auto`
	Monitoring is enabled if you receive a response like the following:
	Total Accesses: 210
	Total kBytes: 94
	CPULoad: .000278279
	Uptime: 366539
	ReqPerSec: .000572927
	BytesPerSec: .262608
	BytesPerReq: 458.362

System	Installation and Configuration Requirements
	BusyServers: 1 IdleServers: 6 Scoreboard: ______W.. **Note:** The default Apache monitor works only if Telnet is enabled. To monitor an Apache server with Telnet disabled, use the `WGet` command instead of the default Rexec commands, as follows: `wget -q -O - http://hostname/server-status | grep "requests/sec" | cut -c5- | cut -d\ -f1`
Application In-Depth	AppDynamics Diagnostics: Built in dynaTrace Diagnostics: Built in
ASP .NET	Using PerfMon
BEA WebLogic 8.x, 9.x, 10.x, 11.x	Performance Explorer can monitor WebLogic 8.x, 9.x, 10.x, and 11.x servers. For monitoring BEA WebLogic 8.x and 9.0 - 9.1, the JNDI protocol is used to connect by way of the Management EJB (MEJB) to the MBean Server. For versions 9.2 and higher, JSR-160 over IIOP is used. Default connection parameters are preconfigured in Performance Explorer. Support for WebLogic Server 11.x includes the following: • 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) Performance Explorer needs to know the application server installation directory or have access to a copy of it to build the correct classpath. A JMX browser allows the selection of the MBean attributes to monitor. An Easy JMX profile can be used to preselect interesting MBeans. Also using SNMP.
Borland Application Server 6.6	Performance Explorer can monitor Borland Application Server 6.6 by way of JSR-160 standard by using the Corbaloc protocol over IIOP. The application server must be configured to allow access to the MBean Server by way of IIOP. Performance Explorer needs to know the application server installation directory or have access to a copy of it to build the correct classpath. A JMX browser allows the selection of the MBean attributes to monitor. An Easy JMX profile can be used to preselect interesting MBeans. Certain MBeans do not function properly without applying patch 7697 to Borland Application Server.
Custom data	• JMX Data • NT Performance Monitor Data • Rexec Data • SNMP Data
F5 Labs 3DNS	Using SNMP
Helix Server	Using PerMon

System	Installation and Configuration Requirements
IBM Universal Database DB2	Using Snapshot
IBM UniversalDatabase DB2 7.1	Using PerfMon
IBM WebSphere Application Server 6.1	Performance Explorer can monitor WebSphere Server without custom configuration of the application server. JNDI protocol is used to connect by way of the Management EJB (MEJB) to the MBean Server. Default connection parameters are preconfigured in Performance Explorer. Performance Explorer needs to know the application server installation directory or have access to a copy of it to build the correct classpath. Performance Explorer needs to use the same IBM JVM version for monitoring. A JMX browser allows the selection of the MBean attributes to monitor. JMX monitoring of WebSphere requires that you specify an IBM JDK.
IBM WebSphere Application Server 7.0, 8.0, 8.5	Performance Explorer can monitor WebSphere Servers without custom configuration of the application server. This does not apply to WebSphere Server 7.0 with enabled administrative security and inbound CSIv2 transport layer. In such a case SSL is required. For details, see the table below. The JSR-160/Plain IIOP protocol is used to connect by way of the Management EJB (MEJB) to the MBean Server. Default connection parameters are preconfigured in Performance Explorer. Performance Explorer needs to know the application server installation directory or needs to have access to a copy of it to build the correct classpath and to set additional JVM parameters correctly. JMX monitoring of WebSphere requires that you specify an IBM JDK with the same version. If your WebSphere server has disabled administrative security or enabled administrative security and the inbound CSIv2 transport layer is TCP/IP, you can specify a SUN JDK. A JMX browser allows for the selection of MBean attributes for monitoring.
Internet Information Server 5.1, 6.0, 7.0, 7.5	Performance Explorer can monitor IIS Web servers without custom configuration. SNMP is supported only with a custom monitoring configuration. Windows NT SNMP Service must be installed before the IIS Web server is installed.
JBoss 4, 5, 6, 7	Performance Explorer can monitor JBoss 4, 5, 6, and 7 Server without custom configuration of the application server. JNDI protocol is used to connect by way of the Management EJB (MEJB) to the MBean Server. Default connection parameters are preconfigured in Performance Explorer. Performance Explorer needs to know the application server installation directory or have access to a copy of it to build the correct classpath. A JMX browser allows the selection of the MBean attributes to monitor. Two Easy JMX profiles can be used to preselect interesting MBeans. To access your JBoss 5, 6 or 7 server remotely, JBoss must not point to the default location `127.0.0.1`, it must point to your server's external IP address. To point JBoss to the external IP address, proceed as follows: **JBoss 5** In `run.conf.bat` of your JBoss application server, add the following setting: `set "JAVA_OPTS=%JAVA_OPTS% -Djboss.bind.address=<your server's external IP>`

System	Installation and Configuration Requirements
	JBoss 6 Start your JBoss application server from the command line with the command `run.bat -b 0.0.0.0` **JBoss 7** Start your JBoss application server from the command line with the command `Standalone.bat -b 0.0.0.0 -Djboss.bind.address.management=<your server's external IP>`
Micro Focus Enterprise Server	Performance Explorer can monitor Micro Focus Enterprise Server without custom configuration of the application server. **Note:** Monitoring support is available for Windows platforms only. When setting up your Micro Focus Enterprise Server monitoring project, from the **Data Source Wizard**, select **Micro Focus** > **Micro Focus Enterprise Server** > **System (perfmon)**. On the **Connection Parameters** dialog, click **Attributes Configuration** to open the **Attributes Configuration** dialog. - In the **#Perfmon.User - Value** field, enter a Windows user name for the server to be monitored. - In the **#Perfmon.Password - Value** field, enter the corresponding password. - In the **#Perfmon.Domain - Value** field, enter the domain of the server to be monitored. - In the **#Region.Name - Value** field, enter the name of the region or the server name.
Microsoft Proxy Server 2.0	Microsoft Proxy Servers are monitored by the Performance Monitor. No special configuration is required.
Operating System Data	Operating system data can be monitored for the following platforms: - Windows - Windows XP/2003 - Windows 7/Vista/2008 - Linux - System (rexec:vmstat) - System (rstat) - System (ucd snmp) - Solaris - System (rexec:mpstat) - System (rexec:vmstat/iostat/df) - System (rstat) - HP-UX - System (rexec:vmstat/iostat) - System (rstat) - AIX (using rstat) - Unix (using rstat) - OSF1 - System (rexec:vmstat/iostat) - System (rstat)

System	Installation and Configuration Requirements
	If available on the host, information such as processor utilization, disc usage, swap file utilization, and network packets can be monitored. On Microsoft systems, operating system data is monitored by the Performance Monitor. For UNIX systems, a remote execution daemon must be running. Such daemons run on UNIX systems by default. **Note:** When fetching values using `Rexec:vmstat`, there can be cases in which Performance Explorer reflects the values of other measures. This can be caused by VMSTAT on a particular server returning values to an unexpected column. The columns in which Performance Explorer expects to receive values from VMSTAT are defined in `realtime.ini` (`C:\Program Files\Silk\Silk Performer10.0\Include`). You may edit the column values in this file to properly capture the values of desired measures. For example, your server might return values for `System\CPU Usage System` in column `13`, while `Operating System Data\Linux\System (rexec:vmstat)` uses column `14` for `System\CPU Usage System`. To correct this behavior, you must edit `realtime.ini`.
Oracle	Using SNMP and v$sysstat
Oracle 9, 10, 11	Using PerfMon Performance Explorer can monitor Oracle 9, 10, and 11 servers without custom configuration. However, Oracle counters for Windows Performance Monitor are not installed with the default Oracle installation options and need to be made available in the custom installation options. Refer to the Oracle Performance Monitoring section in the Performance Explorer Help for detailed information on monitoring Oracle servers.
Oracle Application Server 10.1.3	Performance Explorer can monitor Oracle Application Server 10.1.3 without custom configuration of the application server. JNDI protocol is used to connect by way of the Management EJB (MEJB) to the MBean Server. Default connection parameters are preconfigured in Performance Explorer. Performance Explorer needs to know the application server installation directory or have access to a copy of it to build the correct classpath. A JMX browser allows the selection of the MBean attributes to monitor. An Easy JMX profile can be used to preselect interesting MBeans.
Oracle Forms Application Server	
PeopleSoft	• Application Server (BEA Tuxedo) • Database System • IBM Universal Database DB2 (using snapshot) • Oracle (using v$sysstat) • Operating System • AIX (using ssh) • HPUX (using ssh) • Solaris (using ssh) • Process Scheduler • Oracle

System	Installation and Configuration Requirements
	- SQL Server 2000 - Web Server - Apache - BEA WebLogic (using SNMP)
SAP and SAP V7	SAPGUI Monitoring, SAPGUI OS-Monitoring
Secure Shell	
Silk Central 2008, 2008 R2, 2009, 2009 R2, 2010, 2010 R2, 2011	Application server and front-end server
Silk Performer 10.0	Agent and controller
SQL Server 7.0, 2000, 2005, 2008, 2008 R2	Using PerfMon The most significant versions of SQL Server can be monitored with the Performance Monitor. No special configuration is required, but make sure that all Performance Monitor counters are enabled on the server that hosts your SQL Server.
SUN JVM 1.5, 1.6, and 1.7	Using EJMX and JMX MBeanServer Performance Explorer can monitor SUN JVM 1.5, 1.6, and 1.7 by way of JSR-160 standard using the RMI protocol. Any application or application server using the platform MBean Server can be monitored in this way. No custom configuration is necessary. A JMX browser allows the selection of the MBean attributes to monitor. An Easy JMX profile can be used to preselect interesting MBeans.
Terminal Services	Using PerfMon
VMware ESX Server preconfigured	Using SNMP
Windows Media Services	Using PerfmMon

Security settings WebSphere Version	Disabled administrative security	Enabled administrative security. Inbound CSIv2 Transport Layer = TCP/IP	Enabled administrative security. Inbound CSIv2 Transport Layer = SSL-supported	Enabled administrative security. Inbound CSIv2 Transport Layer = SSL-required
7.0.0.27	+	+	+	- *
8.0.0.5	+	+	+	+
8.5.0.1	+	+	+	+

*) There is successful connection but not all MBeans are acessible.

Automatic Detection of Data Sources

Data-source detection queries your system for available data sources and suggests what can be monitored by using Performance Explorer. Detection is not intended to find every possible data source. The automatic detection is based on a configuration that closely mimics predefined data sources. It goes through the predefined counters and checks for their availability. For example, data-source detection examines the system for the number of processors, the number of network interfaces, the number of database instances, and how these components can be monitored.

The result is a wide selection of data sources that can be monitored, eliminating the need to manually browse your system for performance data sources.

The purpose of the Data Source Scanner is to provide a high level of access to the performance data sources of the system to be monitored. Details such as the platform on which your system runs (NT, AIX, or Linux) do not matter because the characteristics that you monitor, such as CPU and hard disk utilization, are platform-independent.

Using the Data Source Scanner

Query your system for available data sources, which eliminates the need to manually browse your system for performance data sources.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Have Data Sources detected** option button and then click **Next**. The **Connection parameters** page opens.
3. In the **Hostname** text box, specify the machine to examine.
4. *Optional:* In the **Alias** text box, specify the alias name.

   The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.

   For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.
5. Click **Next**.

   Data-source detection is based on specific configurations. It checks for the following data sources on various platforms:

   - .NET
   - Active Server Pages
   - Apache
   - Coldfusion
   - IBM Universal Database DB2
   - Internet Information Servers
   - iPlanet Directory Servers
   - Microsoft Proxy Servers
   - Microsoft SQL Servers
   - Netscape Directory Servers
   - Netscape Enterprise 3.0
   - Netscape/iPlanet Directory Server
   - Operating System Data, such as hard disk, CPUs, and network interfaces
   - Oracle Databases
   - Rstat which often can be queried on UNIX-based systems and provides large amounts of operating system-related data.
   - SilkPerformer Agent
   - SilkPerformer Controller
   - Silverstream

   While the Data Source Scanner examines your system, it provides feedback regarding how far the detection has progressed. The top progress bar indicates overall progress. The bottom bar shows which protocol is currently being used to examine your system.

   Occasionally, the Data Source Scanner presents a dialog box that requests a user ID and password. These dialog boxes appear when, for example, a remote execution daemon is found on the specified host.

The scanner also checks whether your system resides within a remote NT domain. In such instances, you must provide the domain name of the specified host.

Note: The specified user must possess administration security rights, which are necessary for monitoring NT based systems.

6. When the **Progress** section indicates `Done`, click **Next**. The **Select displayed measures** page opens.
7. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Custom Data Sources

In addition to predefined data sources, Performance Explorer offers the following monitoring interfaces:

- PerfMon – Provides access to all WinNT/Windows 2000 monitoring data.
- REXEC – Provides access to data sources on UNIX-based systems that use the Remote Execution Protocol.
- SNMP – Provides access to monitoring data that is exposed by using the Simple Network Management Protocol.

Monitoring PerfMon Data Sources

PerfMon is a Microsoft-specific monitoring interface that queries performance counters. Establish a connection to a PerfMon data source and create an initial view that contains the measures you want to monitor.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Custom Data** folder and click **NT Performance Monitor Data**.
5. Click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the machine to be monitored.

   The host to be monitored might be located on a different NT Domain.
7. *Optional:* In the **Alias** text box, specify the alias name.

   The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.

   For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.
8. In the **Username** text box, specify a user who has administrative security rights.
9. In the **Password** text box, specify the appropriate password for the username.
10. Click **Next**. The **Browse NT Performance Counters** dialog box opens.
11. *Optional:* Examine your system for available performance counters and add them to the monitoring template.

    To select multiple counters from the list, press `Ctrl` or `Shift`.

    a) Select the object that you want to monitor from the **Performance object** list box.

       By default, **Processor** is selected.

    b) Click the **Select counters from list** option button and then select a counter type.

       By default, **% Processor Time** is selected.

    c) Click the **Select instances from list** option button and then select a instance amount.

       By default, **_Total** is selected.

d) Click **Add** to add selected counters to the measure. The **Counter Usage** page opens.
e) Ensure that the **Is an average measure** check box is checked.
f) Click **Next** and then click **Close**. The **Select displayed measures** page opens.

12. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

REXEC and UNIX Data Sources

Describes how to query Unix-based systems using the Remote Execution protocol.

REXEC and UNIX Data Sources Overview

Data sources on UNIX-based systems can often be queried by using the Remote Execution protocol. Identified data sources appear in the predefined data sources list. For example, you might want to monitor **Context switches/sec** on a Solaris system. By selecting this entry in the predefined list, you do not need to specify remote execution details. When data sources are not found in the list, create a custom data source.

Remote Execution

Remote execution means that you execute commands on the remote host by using the command-line interface. You can also execute shell scripts. To verify shell scripts, use a telnet client, connect to your UNIX system, and run some commands or scripts.

Single Execution

With single execution, the primary domain controller emulator (PDCE) connects to the remote machine and executes the command each time a new data point is requested. With short intervals this setting might place some load on the remote machine. However, the PDCE does not start a new command on the remote machine until the command of the last interval has finished.

Without single execution, the PDCE connects to the remote machine at the beginning of the monitoring session, starts the command, and forwards the output of the command to the parser. The parser is activated at the given monitoring interval and parses available data. The command is executed only once, and the connection to the remote machine remains active during the entire monitoring session. If the connection is lost because, for instance, the command on the remote machine exits, the PDCE attempts to reconnect after a fixed interval.

Single Execution Example

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

The preceding command counts the number of processes running on a UNIX system. Enter this command during the telnet session. The command prints a value (the number of processes running) and then closes. Such behavior is called *single execution*.

You can wrap a `‘while’` loop around such commands.

Multiple Execution Example

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

The preceding example continuously prints the number of processes at five-second intervals. Such behavior is defined as *multiple executions*.

Using REXEC to Query UNIX

Establish a connection to a UNIX-based system that uses the Remote Execution Protocol and add the measures that you want to monitor.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Custom Data** folder and click **Rexec Data**.
5. Click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the machine to be monitored.
7. *Optional:* In the **Alias** text box, specify the alias name.

   The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.

   For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.
8. Specify the port, username, and password that are appropriate for the username.
9. Click **Next**. The **Add Rexec Measures** page opens.
10. Specify the appropriate values to create the measure that you want to monitor.

    Option **Description**

    Measure type For the specified command, enter a type name. All measurements with the same type name are assembled into one group.

    Measure name Enter the measurement name for the specified rexec command. This name is displayed in the Performance Explorer chart.

    Is an average measure Check this check box for the rexec command to be considered an average measurement. This setting affects only the generation of TSD files.

    Explanation Type additional information for the specified rexec command. This information is displayed in Performance Explorer.

    Command Specify the rexec command to execute on the remote machine for data collection, such as `ps -ef | egrep -c “.*”`

    Single execution Check this check box if you want the PDCE to connect to the remote machine and execute the command every time a new data point is requested. With short intervals, this setting might place some load on the remote machine. However, the PDCE does not start a new command on the remote machine if the command of the last interval has not finished.

    By unchecking the **Single execution** check box, multiple execution is enabled.

    Regular expr Every line of data that is returned by the remote machine is filtered by using the given regular expression. If the regular expression does not match a certain line, the line is discarded. Only the remaining lines are used for further steps. By default, `^.*$` are entered.

    For example, a command's response might appear as follows for each interval:

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

    Only the lines with numbers are of interest. The following regular expression allows only lines with numbers and spaces.

    ```
    [1234567890 ].*
    ```

    Any other lines are ignored.

Option	Description
Field index	The line used for data generation is separated into fields. The separation is accomplished by using either white-space characters or custom separator characters. Use this text box to specify the field index that must be converted into a numerical value and used as the next measurement value. This index is a one-based index.
Separators	Use this text box to specify the custom separator characters to use to separate the lines for data generation into fields. By default, the **Whitespaces are separators** check box is checked.
Whitespaces are separators	Check this check box to insert a separator for each empty space.
Lines to skip	Use this text box to specify the number of lines the PDCE ignores after the start of the command on the remote machine. This option might be useful to filter out garbage data that the command returns at the beginning. Only lines that match the provided regular expression use this setting. If more than one valid line is provided for an interval, the last line is used for data calculation.

11. Click **Add**.
12. Repeat the previous steps until you have entered all the commands that you require for measure collection, then click **Close**. The **Select displayed measures** page opens.
13. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Example

Enter the command, in slightly modified form, that was used when defining multiple executions as follows:

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

Note: The command `sleep {%Interval}` is a placeholder for a number provided by Performance Explorer. This number is the collection interval. For example, Performance Explorer might collect new data every five seconds.

A command's response might appear as follows for each interval:

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

The **Field index** value specifies the column that is relevant for your defined measure. If **Field index 1** is specified, the value `123400` is returned from the table. If **Field index 2** is specified, the value `103093` is returned.

Columns in the preceding example are separated by white spaces. However, other characters, such as colons, can be used. Such values are specified in the **Separators** text box. White spaces are the default separators.

Memory Free Custom Example

Consider this command for Linux machines:

`cat/proc/meminfo`

It returns the following table.

	`total`:	`used`:	`free`:	`shared`:	`buffers`:	`cached`:
`Mem:`	`131084288`	`54214656`	`76869632`	`26947584`	`7938048`	`28471296`
`Swap:`	`70148096`	`3387392`	`66760704`			
`MemTotal:`	`128012 kB`					
`MemFree:`	`75068 kB`					
`MemShared:`	`26316 kB`					
`Buffers:`	`7752 kB`					
`Cached:`	`27804 kB`					
`SwapTotal:`	`68504 kB`					
`SwapFree:`	`65196 kB`					

The following table identifies the field values to use to obtain the value for **MemFree** from the preceding table:

Command	`cat /proc/meminfo`
Single execution	Checked
Regular expr	^MemFree:.*$
Field index	2
Separators	Check **Whitespaces are separators**
Lines to skip	0

Querying Through SNMP

Performance Explorer provides a set of predefined data sources that are queried through SNMP. However, you can add an entry for ORADB-MIB in an SNMP datasource.

Establish a connection to a data source to query through SNMP and create an initial view that contains the measures you want to monitor.

SNMP involves an SNMP agent, which is typically provided by the application vendor. For example, Oracle databases expose data sources using their SNMP agents. The client-side counterpart is Performance Explorer, which collects performance data.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Custom Data** folder and click **Snmp Data**.
5. Click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host running the SNMP agent.
7. *Optional:* In the **Alias** text box, specify the alias name.

The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.

For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.

8. Specify the port, community, and version that are appropriate.
9. Click **Next**. The **Add SNMP Measures** page opens.
10. Specify the appropriate values to create the measure that you want to monitor as follows:
    a) Select an MIB from the **Select a MIB, and object, and get the object's value(s)** list box. After you select an MIB, the objects and the object values display in a hierarchical tree below the list box.
    b) Select the object that you want to measure. The object's properties are displayed in the text boxes adjacent to the list box.
    c) Click **Get value(s)** to check the object's availability.

    If the performance counter is queried successfully, the object's value is displayed in the grid below the list box.
11. Click **Add** to add the SNMP measure.
12. Repeat the previous steps until you have entered all the commands that you require for measure collection, then click **Close**. The **Select displayed measures** page opens.
13. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Adding MIB Files to Browse SNMP Agents

To browse SNMP agents for available performance counters, your MIB files must contain textual descriptions of performance counters.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Custom Data** folder and click **Snmp Data**.
5. Click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host running the SNMP agent.
7. *Optional:* In the **Alias** text box, specify the alias name.

   The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.

   For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.
8. Specify the port, community, and version that are appropriate.
9. Click **Next**. The **Add SNMP Measures** page opens.
10. Select **Compile new Mib...** from the **Select a MIB, and object, and get the object's value(s)** list box. The **Compile New MIB** dialog box opens.
11. Type the MIB file name that you want to compile in the **MIB source file** text box.

    Alternative: Click **[...]** to navigate to and select the file.

    Commonly used MIB files are located in the `...\INCLUDE\VENDORMIBS` directory of the Silk Performer installation directory.
12. Click **Compile**. The MIB file is added to the **Compiled MIBs** list box.

CSV-to-TSD Data Conversion

Silk Performer offers a command-line tool (`csv2tsd.exe`) that enables you to convert comma-separated values (CSV) files into time series data (TSD) files.

Converting to the TSD format is useful when you monitor external Windows data that, for security reasons, cannot be monitored by Silk Performer. By converting CSV data to TSD format, external data can be analyzed alongside internal Silk Performer monitoring data within Performance Explorer.

For example, if a performance counter measures the behavior of a remote server and the results of that monitor are saved in CSV format, those results can be imported into Performance Explorer for analysis alongside the results of a Silk Performer load test involving the same server. Using Silk Performer, you then have the opportunity to look for correlations between the two data sets.

JMX Monitoring

The Silk Performer support for Java Management Extensions (JMX) monitoring enables you to monitor MBean attributes that are exposed by Java application servers. MBean attributes that return numeric data are added to Performance Explorer monitors as data sources.

Prerequisites for JMX Monitoring

The prerequisites for JMX monitoring include the following items:

- Java Runtime (JVM) 1.5 or later
- The JVM application that you want to monitor must have an open port for monitoring

Connecting to a JMX Data Source

Establish a connection to a Java application server to monitor MBean attributes.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Perform one of the following steps:
   - To explore all available MBean attributes on an application server, expand the **Application Server** folder and select a JMX application profile from a vendor-specific folder. Profile names end with the extension `.jmx`.
   - To explore a selection of MBeans that contain typical attributes for monitoring, expand the **Application Server** folder and select a JMX application profile from a vendor-specific folder. Profile names end with the extension `.ejmx`.
   - To create a generic JMX monitoring data source, expand the **Custom Data** folder and select **JMX Data**.
5. Click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the JMX server.
7. *Optional:* In the **Alias** text box, specify the alias name.

   The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.

   For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.
8. Specify the port, username, and password that are appropriate for the username.
9. If you opted to create a generic JMX monitoring data source, select a supported application configuration from the **Application** list box.

10. Click **Server Configuration**. The **JMX Connection Configuration** dialog box opens.
11. Specify the application server installation directory where the application server's communication libraries are located.

    The relative classpath entries of the application configuration file, together with the specified application server install directory, form the classpath under which the JMX client runs. The resulting classpath is viewed in the **Resulting classpath** text box.

    Note: It is recommended that you use a UNC path or copy the application library directory of the server to your local machine and specify your local copy as the application server installation directory. To complete this task, click **[...]** next to the **App. server install directory** text box.

    Note: JMX monitoring of WebSphere requires that you specify an IBM JDK. If your WebSphere server has disabled administrative security or enabled administrative security and the inbound CSIv2 transport layer is TCP/IP, you can specify a SUN JDK.
12. In the **Java home directory** text box, specify the installation directory where the client's communication libraries are located.

    Alternative: Click **[...]** to navigate to the appropriate directory.
13. Specify additional libraries in the **Additional classpath** text box.
14. Specify additional virtual machine parameters in the **Additional JVM parameters** text box.

    The parameters on the **Connection** and **Visualization** pages are preconfigured for each server.
15. Click **OK**. The **Connection Parameters** page opens.
16. Click **Next** and then click **Finish**. Performance Explorer connects to the JMX server.
17. Examine the exposed beans in the **JMX Data Source Browser**.
18. Click **Finish**.

Specifying MBeans for Monitoring

Application configuration profiles are stored as XML files at `C:\Program Files\Silk\Silk Performer10.0\include\jmx-config\`. Make any changes to the default configurations in the XML file directory, and create additional profiles in this directory as well. Newly created application configuration profiles are available for use as data sources.

1. Open the **JMX Data Source** browser.

   The **Type Tree** page shows all the available categories of MBeans in a tree structure. The **Type List** tab, shows all available categories in a list format. All available MBeans of the selected category are visible in the **Beans** text box. Each MBean is identified by an object name, which consists of a domain name and one or more pairs of property types and values.
2. Select an MBean of interest and expand the MBean node to view the available attributes.
3. *Optional:* Click **Numeric Attributes** to filter non-numeric attributes.

   Non-numeric attributes are not supported for JMX monitoring.
4. *Optional:* Right-click an attribute and choose **Get Value** to query the value of any attribute.
5. *Optional:* Right-click an attribute that represents an object name and choose **Follow Reference** to navigate to the referenced MBean.

Add the appropriate attributes to the bean-attribute pool.

Adding Attributes to the Bean-Attribute Pool

Before you begin this procedure, perform one of the following steps:

- Specify the appropriate MBeans for monitoring.
- Run an Easy JMX or custom query to identify MBeans of interest.

1. Open the **JMX Data Source Browser**.

2. Choose the attributes to monitor in the **Beans** text box.

   Note: Select multiple beans by pressing `Shift` or `Ctrl`.

   Note: Certain MBeans like those based on JSR-77 might contain complex attributes that consist of several submeasures or statistics. To a certain depth of recursiveness, you can select submeasures to be monitored.
3. Right-click the selected attributes and choose **Add As**. A submenu opens.
4. Choose one of the following options for monitoring the attribute values:
   - **Average** – Useful for count values. By default, this option is selected.
   - **Sum** – Same as **Average**, but shows a different label in the monitor GUI.
   - **Incremental** – Useful for rising values. Instead of the attribute value, the change of the value within a monitor interval is calculated.
5. Click **Close**. The **Select displayed measures** page opens.
6. *Optional:* Select an MBean and click **Add**. Performance Explorer adds all visible attributes of an MBean as a data source.
7. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Easy JMX Queries

Easy JMX queries enable access to the most important attribute values of MBeans that are hosted on JMX MBeanServers. Easy JMX queries are preconfigured, advanced queries that are executed automatically when a connection to an application server's MBeanServer is established. All MBeans that match Easy JMX query criteria are automatically displayed in the **Beans** text box of the JMX Data Source browser. Easy JMX profiles are identifiable in the **System** menu tree on the **System Selection** dialog box by the `EJMX` tags that follow them.

Filtering Beans Using JMX Query Language

Display only the MBeans that match specific filter criteria to narrow the JMX query.

1. Open the **JMX Data Source Browser**. All available MBeans on the selected JMX MBeanServer are visible in the **Beans** text box.
2. Click the **Simple Query** tab.
3. Select a domain from the **Domain** list box.

   Leave this field set to **<All>** to search all domains.
4. *Optional:* Select a property from the **Property field** list box.

   This list displays all the properties that are included in the loaded beans.
5. *Optional:* Select a value for the selected property from the rightmost **Property field** list box.

   The Query text box shows the resulting object name that is used for JMX queries, which are based on the JMX 1.2 (JSR-003) standard.
6. *Optional:* Type an attribute name in the **Attribute filter** text box to filter the list based on a specific attribute. Only MBeans containing the specified attributes are displayed.
7. Click **Run** to execute the query. The **Beans** text box displays only those MBeans that match the filter criteria.

Running Advanced JMX Queries

Create a sophisticated combination of simple queries to monitor data for your environment.

1. Open the JMX Data Source Browser.
2. Click the **Advanced Query** tab.

   Available query types are listed in the upper window.

   For example, you might see **Object Name Query**.
3. Select one of the available operators for the query from the list box of the query.
   For example, you might select the **OR** operator.
4. Select a value for the operator from the subsequent list box.
   For example, you might select **True**.
   [...] appears, enabling you to open the JMX Object Name Query dialog box and browse through a domain list.
5. Click **[...]** and select the appropriate domain name from the domain list. For example, you might select **jboss.cache**.
6. Continue adding elements until your query is complete.
   For example, you might add ***:J2EEServer =local**.
7. Click **OK**.
8. Click the root element of the **Query** menu tree.
9. Type a description of the query in the lower pane of the **Advanced Query** page.
10. Click **Run**.

    All MBeans that meet the query's criteria are displayed.
11. Click **Save**.

    This step allows you to specify a target destination and name for the query. The file extension for JMX Data Queries is `.jdq`.

    Note: Queries that are saved to the `C:\Program Files\Silk\Silk Performer10.0\include\jmx-config` directory are also listed on the **Easy JMX** page and can be run directly from there.

Running Saved Queries

Execute an existing JMX query to access the most important attribute values of MBeans that are hosted on JMX MBeanServers.

1. Open the JMX Data Source Browser.
2. Click the **Advanced Query** tab.
3. Choose **Load** > **Load From File** > **Load**.
4. Browse through the list of pre-installed queries and any custom queries that you might have saved.

   Pre-installed query types are available for a number of server types. These types are the same query types that are available with the Easy JMX profiles.
5. Click **Run**.

   Note: Queries that are saved to `C:\Program Files\Silk\Silk Performer10.0\include\jmx-config` are also listed on the **Easy JMX** page and can be run directly from there.

   Performance Explorer executes the query.

Monitoring Servers

This section explains how to set up Performance Explorer to monitor servers during load tests.

Note: You must define a data source before you can monitor servers and applications.

You can view server performance in a live, graphical display as tests run. Server monitoring generates server-side results that you can archive for future viewing and comparing. Monitoring also reveals, locates, and resolves server bottlenecks, allowing you to examine the performance of operating systems and application servers.

Automatic Monitoring

If **Automatically start monitoring** is enabled in the Silk Performer profile when a load test begins, Silk Performer launches Performance Explorer with the monitoring template you assigned to the profile.

All monitor reports begin writing and saving generated TSD files to the load-test directory. The writing of these monitor files is automatically stopped after load tests are complete.

The names of generated TSD files are formatted as `r@NAMEOFMONITOR@STARTTIME.tsd`, where `NAMEOFMONITOR` is the monitor reports caption. After running the load test, TSD files for each monitor report are defined in your workspace.

Monitoring Silk Central Performance

Use Performance Explorer to monitor performance and reliability metrics of a Silk Central application server and front-end server.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Silk** folder and the **Silk Central** folder with the version that you want to monitor, then select application server or front-end server.
5. Click **Next**.

   The **Connection parameters** page opens.
6. Specify the following settings:
   - In the **Hostname** text box, specify the host name or IP address on which Silk Central is running.
   - *Optional*: In the **Alias** text box, specify an alias for the server which is monitored.
   - If Silk Central does not listen on the default port, specify the JMX port number in the **Port** text box.
   - If JMX authentication is enabled, enter the **Username** and **Password**. Per default JMX authentication is disabled and **Username** and **Password** should be empty.

   Note: Make sure that your local firewall is turned off or that ports 19140 and 19142 are allowed to communicate through the firewall.
7. Click **Next**. The **JMX Data Source Browser** opens.
8. Select the measures that you want to include in the initial monitor view and click **Add**. When you are done adding the measures, click **Close**.

   Tip: Select the **Simple Query** tab and filter the measures by `borland.com` to quickly access the relevant Silk Central measures.

   The **Select displayed measures** page opens.
9. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

IBM DB2 Performance Monitoring

Use Performance Explorer to retrieve performance metrics from IBM DB2 database systems no matter the database platform. For example, the database can run on a SunOs, Linux, or a Windows system.

With Performance Explorer, you can monitor DB2 database systems in a platform-independent way. To accomplish this task, execute a DB2-specific command for retrieving a database's performance and utilization measures and drive a Silk Performer project inside Performance Explorer. This approach also allows Performance Explorer to keep track of the database system's performance in real time.

The Silk Performer project is located in a Essential package named `DB2Monitor.sep`, which is located at `C:\Program Files\Silk\Silk Performer10.0\Monitors`.

Prerequisites for DB2 Monitoring

To monitor DB2, ensure that your environment meets the following requirements:

- IBM DB2 client software (DB2 Connect) must be installed on the machine on which you intend to run Performance Explorer.
- The DB2 command line processor, `db2cmd.exe`, must be installed as part of the client installation. This prerequisite is met by default.
- The appropriate DB2 database user must possess one of the following authorizations:
  - sysadm
  - sysctrl
  - sysmaint
- DB2 snapshot monitoring must be set up correctly.

Testing DB2 Snapshot Monitoring

1. Connect to a DB2 database.

   db2 command:

   ```
   => connect to sample user <dbusername> using <password>
   ```
2. Attach to a DB2 instance.

   db2 command:

   ```
   => attach to <db2instancename> user <dbusername> using <dbusername>
   ```
3. Check if monitor switches are on.

   db2 command:

   ```
   => get monitor switches
   ```

   Monitor Recording Switches

   Switch list for node 0

   ```
   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. Turn on switches.

   db2 commands:

   ```
   => update monitor switches using bufferpool on
   ```

   ```
   => update monitor switches using lock on
   ```

   ```
   => update monitor switches using sort on
   ```
5. Retrieve data from the DB2 UDB system monitor snapshot.

   db2 command:

   ```
   => get snapshot for all databases
   ```

Database Snapshot:

```
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
```

Retrieving IBM DB2 Performance Metrics

Establish a connection to a data source to monitor IBM DB2 DBMS systems and to create an initial view that contains the measures you want to monitor.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Database System** folder and select **IBM Universal Database DB2 (snapshot dll)**.
5. Click **Next**.

   The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which the DB2 database is running.

7. Click **Next**. The **Attributes Configuration** page opens.
8. Define the following monitoring-specific attributes:
   - **Alias** – The database alias is used by the monitoring project in the following DB2-specific command: `"connect to 'alias' user 'user' using 'password'"`.
   - **Instance** – The instance name is used by the monitoring project for executing the following DB2-specific command: `"attach to 'Instance' user 'user' using 'password'"`.
   - **User** – Specify a user with the following DB2 authorizations:
     - sysadm
     - sysctrl
     - sysmaint
   - **Password** – The password for the user.
9. Click **OK**. The **Select displayed measures** page opens.
10. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

IBM WebSphere Application Server Monitoring

With Performance Explorer, you can monitor the following IBM WebSphere application servers:

- IBM WebSphere Application Server 6.1 via JMX
- IBM WebSphere Application Server 7.0 via JMX
- IBM WebSphere Application Server 8.0 via JMX
- IBM WebSphere Application Server 8.5 via JMX

Typical Performance Measures on WebSphere Application Server 6.1

The following table displays typical performance measures and definitions for WebSphere application server.

Performance Measure	Description
BeanModule	Specific data about deployed EJBs, such as how many beans were created.
ConnectionPoolModule	Contains performance counters about JDBC connections, like how many JDBC connections are currently in the pool.
JvmRuntimeModule	Java Virtual Machine specific performance counters.
ServletSessionModule	Contains performance counters about servlet sessions, such as how many sessions are active or the average session live time.
TransactionModule	Contains performance counters providing information about how many transactions are in progress or the average time required to execute one transaction.
WebApplicationModule	Provides information about deployed servlets and JSPs, such as the average time required for the servlet to process requests.

Monitoring IBM WebSphere Application Server

1. Choose **Monitor** > **Add Data Source**. The **Data Source wizard** opens.
2. Click the **Select from predefinded Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Application Server** folder and the **IBM WebSphere Application Server** folder.
5. Click the **IBM WebSphere <version> (JMX MBean Server)** node and then click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** field, specify the host on which WebSphere is running.
7. In the **User** and **Password** fields, specify a user with administrator permissions for the above specified host.
8. *Optional:* In the **Alias** field, specify an alias for the monitored server.
9. Click **Server Configuration...** The **JMX Connection Configuration** window opens.
10. Specify the **Java Home directory** and the **Application Server install directory**. Use a UNC path (for example, `\\<server name>\c$\IBM\WebSphere\AppServer\java`).

    Note: If you cannot specify a UNC path, copy the application server installation directory to your local machine and specify a local path.
11. Specify additional JVM parameters based on WebSphere security settings.

JVM vendor	IBM	SUN
Security settings		
Disabled administrative security	Without additional JVM parameters	Without additional JVM parameters
Enabled administrative security Inbound CSIv2 Transport Layer = TCP/IP	Additional JVM parameter: `-Dcom.ibm.CORBA.ConfigURL` (already predefined)	Additional JVM parameter: `-Dcom.ibm.CORBA.ConfigURL` (predefined)
Enabled administrative security Inbound CSIv2 Transport Layer = SSL-supported	Additional JVM parameters: `-Dcom.ibm.CORBA.ConfigURL` (predefined) `-Dcom.ibm.SSL.ConfigURL`	Does not work
Enabled administrative security Inbound CSIv2Transport Layer = SSL-required	Additional JVM parameters: `-Dcom.ibm.CORBA.ConfigURL` (already predefined) `-Dcom.ibm.SSL.ConfigURL`	Does not work

Note: If your WebSphere server has enabled administrative security and the inbound CSIv2 transport layer is SSL-supported (or SSL-required), copy the application server installation directory to your local machine. Specify **Additional JVM parameters** according to the table above. The **Additional JVM parameters** field holds the predefined parameter `-Dcom.ibm.CORBA.ConfigURL`.

The value for the `-Dcom.ibm.CORBA.ConfigURL` parameter must be the URL path to the file `sas.client.props`. The default `sas.client.props` file is located in the folder `%INSTALL_DIR%/Include/jmx-config/WebSphere%MAJOR_VERSION%`. For example: For WebSphere 7, the URL path is `C:\Program%20Files%20(x86)\Silk\Silk%20Performer%2010.0\Include\jmx-config\WebSphere7`. If your server has specific settings, you can use the `sas.client.props` file from the properties folder of the WebSphere profile. For Example: `file:/C:\Program%20Files%20(x86)\IBM\WebSphere\AppServer1\profiles\AppSrv01\properties\sas.client.props`

Pay attention to the following two variables and their values in your `sas.client.props` file: `com.ibm.CSI.performTransportAssocSSLTLSRequired=false` and `com.ibm.CSI.performTransportAssocSSLTLSSupported=true`

The value for the `-Dcom.ibm.SSL.ConfigURL` parameter must be the URL path to the file `ssl.client.props`, which is found in the properties folder of the WebSphere profile. For example: `file:/C:\Program%20Files%20 (x86)\IBM\WebSphere\AppServer1\profiles \AppSrv01\properties\ssl.client.props`.

Pay attention to the variable `user.root` in the file `ssl.client.props`. It must contain the valid path to the profile folder with the etc directory that contains the key- and trust-files. Blanks in the URL path to the files `sas.client.props` and `ssl.client.props` must be replaced with `%20`.

12. Click **OK**.
13. Click **Next**. The **JMX Data Source Browser** opens.
14. Choose MBeans for those measures that you want to include in the initial monitor view. Select the type of measures (**Average**, **Sum**, or **Incremental**) and click **Add**.
15. Click **Close**. The **Select displayed measures** page opens.
16. Check the checkboxes for those measures that you want to include in the initial monitor view.
17. Click **Finish**.

Internet Information Server (IIS) Performance Monitoring

Monitoring Internet Information Services (IIS)

1. Choose **Monitor** > **Add Data Source**. The **Data Source wizard** opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the `Web Server` folder and the `IIS <version>` folder.
5. Click the `System Statistics (PERFMON)` node and then click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which IIS is running.
7. In the **User** and **Password** text boxes, specify a user with administrator permissions on the above-specified host.
8. *Optional:* In the **Alias** text box, specify an alias for the monitored server.
9. Click **Next**. The **Select displayed measures** page opens.
10. Check the check boxes for those measures that you want to include in the initial monitor view.
11. Click **Finish**.

Available IIS Performance Measures

Performance Measures on IIS 6.0

Measures can be divided into several groups according to perfmon objects.

Web Service Measures

- **Web Service(_Total)\Bytes Received/sec**

  Rate of total bytes transferred by service (received)
- **Web Service(_Total)\Bytes Sent/sec**

  Rate of total bytes transferred by service (sent)

- **Web Service(_Total)\Bytes Total/sec**

  Rate of total bytes transferred by service (sum of bytes sent and received)
- **Web Service(_Total)\Current Connections**

  Current number of connections to the service
- **Web Service(_Total)\Get Requests/sec**

  Total number of HTTP GET requests received by WWW service
- **Web Service(_Total)\Post Requests/sec**

  Number of HTTP requests using POST method

Web Service Cache Measures

- **Web Service Cache(_Total)\Current Files Cached**

  Current number of files whose content is in the user-mode cache
- **Web Service Cache(_Total)\Current Metadata Cached**

  Current number of metadata information blocks currently in the user-mode cache.
- **Web Service Cache(_Total)\Current URIs Cached**

  URI information blocks currently in the user-mode cache
- **Web Service Cache(_Total)\File Cache Hits %**

  The ratio of user-mode file cache hits to total cache requests (since service startup). Note: This value might be low if the Kernel URI cache hits percentage is high.
- **Web Service Cache(_Total)\Metadata Cache Hits**

  The ratio of user-mode metadata cache hits to total cache requests (since service startup)
- **Web Service Cache(_Total)\URI Cache Hits %**

  The ratio of user-mode URI Cache Hits to total cache requests (since service startup)
- **Web Service Cache(_Total)\Kernel:URI Cache Hits %**

  Applies to static unauthenticated content and dynamic content that is marked as cacheable

Performance Measures on IIS 7.0, 7.5

All IIS 6.0 measures are available for IIS 7.0/7.5, but in IIS7 there are additional performance counters that give you insight into worker processes level at runtime.

Web Service Measures - Same as IIS 6.0.

Web Service Cache Measures - Same as IIS 6.0.

W3SVC_W3WP Cache Measures

- **W3SVC_W3WP(_Total)\Requests / Sec**

  HTTP requests/sec being processed by the worker process
- **W3SVC_W3WP(_Total)\Active Requests**

  Current number of requests being processed by the worker process
- **W3SVC_W3WP(_Total)\Active Threads Count**

  Number of threads actively processing requests in the worker process
- **W3SVC_W3WP(_Total)\Current File Cache Memory Usage**

  Current number of bytes used by user-mode file cache
- **W3SVC_W3WP(_Total)\Current Files Cached**

  Current number of files whose contents are present in user-mode cache
- **W3SVC_W3WP(_Total)\Current URIs Cached**

  URI information blocks currently in the user-mode cache

- **W3SVC_W3WP(_Total)\Current Metadata Cached**

  Number of metadata information blocks currently present in user-mode cache
- **W3SVC_W3WP(_Total)\Metadata Cache Hits**

  Total number of successful lookups in the user-mode metadata cache (since service startup)
- **W3SVC_W3WP(_Total)\Metadata Cache Misses**

  Total number of unsuccessful lookups in the user-mode metadata cache (since service startup)
- **W3SVC_W3WP(_Total)\Metadata Cache Flushes**

  Total number of user-mode metadata cache flushes (since service startup)
- **W3SVC_W3WP(_Total)\File Cache Hits / sec**

  Rate of successful lookups in file cache during last sample interval
- **W3SVC_W3WP(_Total)\File Cache Misses / sec**

  Rate of unsuccessful lookups in file cache during last sample interval
- **W3SVC_W3WP(_Total)\Metadata Cache Hits / sec**

  Rate of successful lookups in metadata cache during last sample interval
- **W3SVC_W3WP(_Total)\Metadata Cache Misses / sec**

  Rate of unsuccessful lookups in metadata cache during last sample interval
- **W3SVC_W3WP(_Total)\Uri Cache Hits / sec**

  Rate of successful lookups in URI cache during last sample interval
- **W3SVC_W3WP(_Total)\Uri Cache Misses / sec**

  Rate of unsuccessful lookups in URI cache during last sample interval

Troubleshooting IIS Monitoring

Issues With x64/x86 Counters

For support, please refer to *http://support.microsoft.com/kb/891238*.

Issues With W3SVC_W3WP

Note that this group of counters monitors worker processes at runtime. So when no workers are running, this group of counters returns NULL values.

Microsoft SQL Server Performance Monitoring

Monitoring Microsoft SQL Server

1. Choose **Monitor** > **Add Data Source**. The **Data Source wizard** opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the `Database System` folder and the `SQL Server <version>` folder.
5. Click the `System Statistics (PERFMON)` node and then click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which SQL Server is running.
7. In the **User** and **Password** text boxes, specify a user with administrator permissions on the above-specified host.
8. *Optional:* In the **Alias** text box, specify an alias for the monitored server.
9. Click **Next**. The **Select displayed measures** page opens.

10. Check the check boxes for those measures that you want to include in the initial monitor view.

11. Click **Finish**.

Available Microsoft SQL Server Performance Measures

Performance Measures on Microsoft SQL Server 2005

The following metrics can be monitored on Microsoft SQL Server 2005 systems:

- **Process(sqlservr)\% Processor Time**

  % Processor Time is the percentage of elapsed time that all of the threads of this process used the processor to execute instructions.
- **SQLServer:Access Methods\Full Scans/sec**

  Number of unrestricted full scans. These can either be base table or full index scans.
- **SQLServer:Access Methods\Index Searches/sec**

  Number of index searches. Index searches are used to start range scans, single index record fetches, and to reposition within an index.
- **SQLServer:Access Methods\Table Lock Escalations/sec**

  The number of times locks on a table were escalated.
- **SQLServer:Buffer Manager\Buffer cache hit ratio**

  Percentage of pages found in the buffer cache without having to read from disk.
- **SQLServer:Buffer Manager\Checkpoint pages/sec**

  Number of pages flushed by checkpoint or other operations that require all dirty pages to be flushed.
- **SQLServer:Buffer Manager\Lazy writes/sec**

  Number of buffers written by buffer manager's lazy writer.
- **SQLServer:Buffer Manager\Page lookups/sec**

  Number of requests to find a page in the buffer pool.
- **SQLServer:Buffer Manager\Page reads/sec**

  Number of physical database page reads issued.
- **SQLServer:Buffer Manager\Page writes/sec**

  Number of physical database page writes issued.
- **SQLServer:Buffer Manager\Readahead pages/sec**

  Number of pages read in anticipation of use.
- **SQLServer:Cursor Manager by Type\Active cursors**

  Number of active cursors.
- **SQLServer:Cursor Manager by Type\Cursor memory usage**

  Amount of memory consumed by cursors in kilobytes (KB).
- **SQL Server:Databases\ Active Transactions**

  Number of active update transactions for the database.
- **SQL Server:Databases\ Shrink Data Movement Bytes/sec**

  The rate data is being moved by Autoshrink, DBCC SHRINKDATABASE or SHRINKFILE.
- **SQL Server:Databases\ Transactions/sec**

  Number of transactions started for the database.
- **SQLServer:General Statistics\User Connections**

  Number of users connected to the system.
- **SQLServer:Locks(_Total)\Average Wait Time (ms)**

The average amount of wait time (milliseconds) for each lock request that resulted in a wait.

- **SQLServer:Locks(_Total)\Lock Waits/sec**

  Number of lock requests that could not be satisfied immediately and required the caller to wait before being granted the lock.
- **SQLServer:Locks(_Total)\Number of Deadlocks/sec**

  Number of lock requests that resulted in a deadlock.
- **SQLServer:Memory Manager\Target Server Memory (KB)**

  Total amount of dynamic memory the server is willing to consume.
- **SQLServer:Memory Manager\Total Server Memory (KB)**

  Total amount of dynamic memory the server is currently consuming.
- **SQLServer:SQL Statistics\Batch Requests/sec**

  Number of SQL batch requests received by server.
- **SQLServer:SQL Statistics\SQL Compilations/sec**

  The number of SQL compilations per second.
- **SQLServer:SQL Statistics\SQL Re-Compilations/sec**

  The number of SQL re-compiles per second.
- **SQLServer:Transactions\Free Space in tempdb (KB)**

  The amount of space (in kilobytes) available in tempdb. There must be enough free space to hold both the snapshot isolation level version store and all new temporary objects created in this instance of the database engine.
- **SQLServer:Transactions\Transactions**

  The number of currently active transactions of all types.

Performance Measures on Microsoft SQL Server 2008, 2008 R2

All Microsoft SQL Server 2005 counters are still available in Microsoft SQL Server 2008 and 2008 R2 systems. Additional metrics are also available:

- **SQL Server:Databases\Tracked transactions/sec**

  Number of committed transactions recorded in the commit table for the database.
- **SQL Server: Databases\Write Transactions/sec**

  Number of transactions which wrote to the database in the last second.
- **SQL Server: General Statistics\Connection reset/sec**

  Total number of connection resets per second.
- **SQL Server: General Statistics\Tempdb rowset id**

  Number of duplicate tempdb rowset id generated
- **SQL Server: SQL Statistics\Misguided plan executions/sec**

  Number of plan executions per second in which a plan guide could not be honored during plan generation
- **SQL Server: SQL Statistics\Guided plan executions/sec**

  Number of plan executions per second in which the query plan has been generated by using a plan guide.

Oracle Forms Performance Monitoring

Use Performance Explorer to retrieve performance metrics from Oracle Forms Dynamic Monitoring System (DMS). Performance Explorer can monitor the DMS performance Web page that is provided through the Oracle Forms HTTP server when the DMS module is installed.

Web page requests and the parsing of performance metrics are achieved by driving a Silk Performer project through Performance Explorer. This approach allows you to keep track of application server performance in real time.

The Silk Performer project is located in a SilkEssential package named `OraFormsDMS.sep`, which is located at `C:\Program Files\Silk\Silk Performer10.0\Monitors`.

The Oracle Forms application architecture includes three tiers:

- Oracle database
- Oracle application server
- Client applet

Performance Explorer should be used to gather server-side measures from the database and application servers.

To monitor the database server a set of predefined Oracle database measures is available.

Prerequisites for Oracle Forms Monitoring

To monitor Oracle Forms, ensure that your environment meets the following requirements:

- Oracle Forms Dynamic Monitoring System must be installed on the Oracle Forms application server.
- The performance page must be accessible from the monitoring machine.

Monitoring Oracle Forms

Note: In addition to default measures for the application server, Performance Explorer offers a BDL monitor that allows querying of performance data from the Oracle Forms Dynamic Monitoring System (DMS). Please review Oracle Technology Network instructions (*http://otn.oracle.com*) for setting up DMS on your Oracle Application Server.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.

   Optional: To monitor the application server you can use measures that are provided by the platform upon which the application server runs. Choose from predefined sets or use `PERFMON`, `SNMP` or `REXEC` and manually define the measures that are of interest to you.
4. Expand the **Application Server** folder and the **Oracle** folder.
5. Select **Oracle Forms AS** and click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which the Oracle application server is running. You may optionally add an **Alias** name for the application server.
7. Click **Next**. The **Attributes Configuration** page opens.
8. Specify the following settings:

   - URL of the **DMS-Site**. The standard format is `http://server:port/dms0`
   - **Proxy** (if you need to connect using an HTTP proxy)
   - `Proxy-Port` of the HTTP proxy
   - **Username** (if you need to authenticate against the Web server to access the DMS site)
   - **Password** for accessing the DMS site

   This monitor provides interesting measures that are provided by the DMS including:

   - Common Apache measures
   - Module information for HTTP, OC4J, and PLSQL
   - Database connection statistics
9. Click **OK**. The **Select displayed measures** page opens.

10. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Oracle Forms Client-Side Measures

Client-side measures are automatically generated by load test agents during Oracle Forms load tests. They are aggregated into a single result file by the load test controller.

Agents create measures for the following:

- Time taken by an action on a control (for example, button press or control edit)
- Number of bytes received for an action
- Number of bytes sent for an action
- Number of messages received for an action
- Number of messages sent for an action
- Number of round-trips for an action

Oracle Performance Monitoring

Use Performance Explorer to retrieve performance metrics from Oracle database systems no matter the database platform. For example, the database can run on a SunOs, Linux, or a Windows system.

Monitoring Oracle Database via Perfmon

Oracle Database Monitoring Prerequisites

Installing Oracle Client and Oracle Counters for Windows

Before you monitor Oracle Database, ensure that the following requirements have been met:

- SQL*NET client software must be installed on the machine on which you run Performance Explorer.
- SQL*Plus must be available on the machine on which you run Performance Explorer.
- Oracle Counters for Windows Performance Monitor must be available.
- The Oracle client must have DB administrator permissions and be able to recognize the targeted database server.

Note: Oracle Counters for Windows Performance Monitor is not installed with default installation options (for example, Instant Client and Administrator installations). It is only available via custom installation options.

Adding Oracle Counters to Windows Performance Monitor

1. Open file `<Oracle Home>\network\admin\tnsnames.ora` and add the connection string to the database you want to monitor.
2. Start the command line and test the connection with the command `sqlplus <user>/<password>@<database>`
3. Configure which database is to be monitored using the `operfcfg` command: `<Oracle Home>\bin\operfcfg [-U username] [-P password] [-D TNS_Alias_for_database]`

   If successful, you will receive the message `operfcfg: New registry values have been successfully set.`
4. Start **Windows Performance Monitor Start** > **Run** > **Perfmon**.
5. Oracle Counters for Windows Performance Monitor can now be added via the **Add Counters** dialog box.

Monitoring Oracle Performance (Perfmon)

1. Choose **Monitor** > **Add Data Source**. The **Data Source wizard** opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the `Database System` folder and the `Oracle 9/10/11 (perfmon)` folder.
5. Click the `System Statistics ( PERFMON )` node and then click **Next**. The **Connection Parameters** page opens.
6. In the **Hostname** text box, specify the host on which the Oracle Client and Counters for Windows Performance Monitor are running.
7. In the **User** and **Password** text box, specify the user with administrator permissions on the above specified host.
8. *Optional:* In the **Alias** text box, specify an alias for the host where the Oracle database is running.
9. Click **Next**.
10. The **Select displayed measures** page opens.
11. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

Performance Counters

Oracle Database Buffer Cache

The counter is `phyrds/gets %`. The percentage of phyrds/gets is calculated as a miss ratio. The lower the `Miss` counter, the better. To improve performance, increase the number of buffers in the buffer cache, if memory is available on the computer. This value is not time-derived.

Oracle Database Redo Log Buffer

The counter is `redo log space requests`. The value of this counter must be near zero. If this value increments consistently, then processes have had to wait for space in the redo log buffer. In this case, it may be necessary to increase the size of the redo log buffer.

Oracle Database Data Dictionary Cache

The counter is `getmisses/gets %`. The value of this counter must be less than 10-15% for frequently accessed data dictionary caches. If the ratio continues to increase over this threshold while your application is running, then increase the amount of memory available to the data dictionary cache.

To increase the memory available to the cache, increase the value of initialization parameter `SHARED_POOL_SIZE`. This value is not time-derived.

Oracle Database Library Cache

The counter is `reloads/pins %`. This is the percentage of SQL statements, PL/SQL blocks, and object definitions that required reparsing. Total Reloads must be near zero. If the ratio of Reloads to Pins is greater than 1%, then reduce the library cache misses. This value is not time-derived.

Oracle Database DBWR stats1

The two counters available, buffers scanned/sec and LRU scans/sec, are helpful in tuning Buffer Cache. Buffers scanned/sec is the number of buffers DBWR scanned in each second. The buffers scanned are on the LRU (Least Recently Used) list. LRU scans/sec is the number of times DBWR scanned the (Least Recently Used) buffer list in each second.

Oracle Database DBWR stats2

The two counters available, timeouts/sec and checkpoints/sec, are helpful in determining how much work DBWR has been requested to perform. Timeouts/sec is the number of times DBWR timed-out in each second. DBWR is on a three second timeout interval. If DBWR has not been posted within a three second interval, then it times out.

Checkpoints/sec is the number of checkpoint messages processed by the database writer each second. Whenever a checkpoint occurs, DBWR must be messaged (posted) to "`write dirty buffers to disk.`"

Oracle Database Dynamic Space Management

The counter is `recursive calls/sec`. Dynamic extension causes Oracle Database to execute SQL statements in addition to those SQL statements issued by user processes. These SQL statements are called recursive calls.

Oracle Database Free List

The counter is `free list waits/requests %`. Contention for free lists is reflected by contention for free data blocks in buffer cache. You can determine if contention for free lists is reducing performance by querying V$WAITSTAT.

If the number of free list waits for free blocks is greater than 1% of the total number of requests, then consider adding more free lists to reduce contention.

Oracle Database Sorts

The available counters are sorts in memory/sec and sorts on disk/sec. The default sort area size is adequate to hold all data for most sorts. However, if your application often performs large sorts on data that do not fit into the sort area, then you may increase sort area size.

Known Issues

Silk Performer can monitor both 32bit and 64bit Oracle Database performance, but only the 32bit Oracle client version (Oracle Counters for Windows Performance Monitor) is supported.

Note: The 32bit Oracle client works fine on a 64bit OS and it is not an issue to monitor a 64bit database from a client machine running a 64bit OS.

Monitoring Oracle Database via V$SYSSTAT

You can monitor DB2 database systems in a platform-independent way by fetching performance information from an Oracle database's V$SYSSTAT table.

Prerequisites for Oracle Monitoring

Before you monitor Oracle, ensure that the following requirements have been met:

- Oracle Net (part of Oracle Net Services) client software is installed on the machine on which you run Performance Explorer.
- The Oracle client must be a 32-bit version, must have DB administrator permissions and be able to recognize the targeted database server.
- The relevant project attributes have been changed accordingly for your Oracle installation (choose **Menu Project** > **Project Attributes**).
- SQL*Plus is available on the machine on which you want to run Performance Explorer.
- Log on using the specified user name, password, and server information and perform the query `SELECT * FROM V$SYSSTAT`.

  If the logon attempt is successful, the environment is most likely set up correctly.

If the logon attempt fails, you can judge from the error messages generated by SQL*Plus what caused the problem.

Monitoring Oracle Performance

Drive a Silk Performer project from inside Performance Explorer to keep track of the database system's performance in real time. The Silk Performer project is located a Essential package named `OracleMonitoring.sep`, which is located at `C:\Program Files\Silk\Silk Performer10.0\Monitors`.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Database System** folder and the **Oracle (v$sysstat)** folder.
5. Click the **System Statistics (v$sysstat)** node and then click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which the Oracle database is running.
7. *Optional*: In the **Alias** text box, specify an alias for the host where the Oracle database is running.
8. Click **Next**. The **Attributes Configuration** page opens.
9. If necessary, change the attribute values.

   Typically, the Oracle TNS name, UserId, and Password are included in these settings.
10. Click **OK**. The **Select displayed measures** page opens.
11. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

SAP Performance Monitoring

Use Performance Explorer to monitor performance and reliability metrics of a SAP system. Performance Explorer is shipped with two monitors specially designed for monitoring SAP installations.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Expand the **Application Server** folder and the **SAP** folder.
4. Depending on the type of SAPGUI monitor that you want to use, select the appropriate monitor type.

   The following types of SAPGUI monitors are available:

   - SAPGUI Monitoring (ST02): Monitors buffer related metrics. SAP transaction `ST02` is executed when running this monitor.
   - SAPGUI Monitoring (ST03N): Monitors application specific metrics. SAP transaction `´ST03n` is executed when running this monitor.
   - SAPGUI Monitoring (ST04): Monitors database related metrics. SAP transaction `ST04` is executed when running this monitor.
   - SAPGUI Monitoring (ST07): Monitors user distribution on the SAP system. SAP transaction `ST07` is executed when running this monitor.
   - SAPGUI OS-Monitoring (ST06): Monitors operating system specific metrics. SAP transaction `ST06` is executed when running this monitor.
5. Click **Next**.

   The **Connection parameters** page opens.

6. In the **Hostname** text box, specify the host of the machine to be monitored

   This value is for display purposes only. It is not used in the monitor itself.
7. Click **Next**. The **Attributes Configuration** page opens.
8. Define the following monitoring-specific attributes:

   - **ConnectionString** – Complete connection string to the SAP server. If you are not sure, record the logon sequence with Silk Performer. The connection string is the first parameter of `SapGuiOpenConnection`.
   - **Username** – SAP username with access rights to the monitoring transaction.
   - **Password** – Password for the SAP user.
   - **ClientNum** – Client number for the logon procedure, such as `850`.
   - **Language** – Language to use for logon, such as `EN`.
   - **Entity** – The data entity that should be monitored, such as `Dialog`, `RFC`, or `Background`.

     Note: The **Entity** attribute does not have to be provided for the SAP OS-Monitor.
   - **TimeFrame** – SAP provides the average values of the past interval that you define.

     Note: The **Timeframe** attribute does not have to be provided for the SAP OS-Monitor.
   - **Server -** *(ST03N only)*The server that should be monitored. Its the name of the tree node that is to be selected in ST03N. Total will return measures from the overall SAP System
9. Click **OK**. The **Select displayed measures** page opens.
10. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Secure Shell Performance Monitoring

Use Performance Explorer to retrieve performance metrics by using a secure shell. For example, retrieve CPU-utilization or memory-utilization statistics from a SunOs system. Performance Explorer is shipped with a highly configurable performance-monitoring project that allows for the scripting of any monitors by using the secure shell. This process is done by driving a Silk Performer project inside Performance Explorer. This capability also allows you to keep track of system performance in real time.

The Silk Performer project is located in a SilkEssential package named `Ssh.sep`, which is located at `C:\Program Files\Silk\Silk Performer10.0\Monitors`.

Monitoring Performance by Using a Secure Shell

The system that you want to monitor by using a secure shell must be running a secure shell daemon. Make sure the monitored user has permission to log on multiple times.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Miscellaneous** folder and click **Secure Shell**.
5. Click **Next**.

   The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host that you want to monitor.
7. Click **Next**. The **Attributes Configuration** page opens.
8. Define the following monitoring specific attributes:

- **Username** – The user to be used when logging on to a remote system by using a secure shell.
- **Password** – The user's password.
- **Command(x)** – The preconfigured package allows for retrieving up to five performance measures by using secure shell as the following table shows:

Attribute	Type	Value
Command1	string	`ps -ef | egrep -c \”.*\”`
Command1.Active	boolean	`true`
Command1.Column	number	`1`
Command2	string	`mpstat | head -n 2 | tail -1`
Command2.Active	boolean	`true`
Command2.Column	number	`13`

9. Click **OK**. The **Select displayed measures** page opens.

10. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

Example Command2

This example works for a SunOs system. The following command is sent to the server.

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

This command returns a one-line response in the following format.

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

The attribute `Command2.Column` specifies which column to pass on to Performance Explorer for keeping track of the performance counter. In this case, it is column 13, which is the percentage of user time on a SunOs system.

`Command2.Active` specifies only whether this measure must be collected.

Example Command1

The following command is sent to the server.

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

This command counts the number of processes running on a SunOs system and could return a one-line response in the following format.

```
87
```

Therefore, one line is returned with exactly one column.

The attribute `Command1.Column` specifies the column to forward to Performance Explorer. In this case, it is column `1`.

`Command1.Active` specifies only whether this measure must be collected.

To learn how to modify monitoring projects, see the Silk Performer Advanced Concepts Help.

WebLogic (SNMP) Monitoring

This section introduces how to monitor BEA WebLogic using Performance Explorer and explains important WebLogic performance measures.

By default, the WebLogic SNMP agent is not running. You can enable SNMP support by using the WebLogic Administration Console. Open the console by typing the URL `http://weblogichost:port/console` into your browser. In the menu tree on the left, select the SNMP node. This selection opens the **SNMP configuration** page. Within this page, verify that the SNMP agent is enabled. You must restart WebLogic to enable SNMP support.

Monitoring BEA WebLogic (SNMP)

Establish a connection to a data source to query BEA WebLogic through SNMP and to create an initial view that contains the measures you want to monitor.

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next**. The **System selection** page opens.
4. Expand the **Application Server** folder and the **BEA WebLogic (SNMP)** folder.
5. Select the **SNMP** node and then click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which WebLogic is running.
7. Click **Next**. The **Add SNMP Measures** page opens. **BEA-WEBLOGIC-MIB** is selected in the list box of available MIBs.

   The following entries for monitoring the performance of your WebLogic Server are available:

   - bea.wls.jdbcConnectionPoolRuntimeTable
   - bea.wls.servletRuntimeTable
   - bea.wls.ejbEntityHomeRuntimeTable
   - bea.wls.ejbStatelessHomeRuntimeTable
   - bea.wls.executeQueueRuntimeTable
8. Select **jdbcConnectionPoolRuntimeEntry** and click **Get value(s)** to retrieve all available JDBC connection pools.
   For example, when monitoring the Pet Store sample application, a connection pool named `ServerRuntime:petstoreServer` is returned.

   Interesting performance counters exposed by connection pools include the following counters:

   - **JdbcConnectionPoolRuntimeActiveConnectionsCurrentCount** – Returns the number of connections that are currently being used.
   - **JdbcConnectionPoolRuntimeWaitingForConnectionCurrentCount** – Returns the number of waiters for connections. This number is greater than zero when all connections are in use and additional requests for connections are being sent to the connection pool.
   - **JdbcConnectionPoolRuntimeWaitSecondsHighCount** – Returns the maximum number of seconds a client waits to check out a connection from the connection pool.
9. To monitor the number of active connections, search for the **jdbcConnectionPoolRuntimeActiveConnectionsCurrentCount** column in the table that was previously returned and select the cell.

   Note: Connection pools provide a large number of performance counters. You might need to resize some columns to view the columns in which you are interested.
10. Click **Add**.
11. Repeat the previous steps until you have entered all commands you require for measure collection, then click **Close**. The **Select displayed measures** page opens.

12. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.

A connection to the specified host is established, and an initial view that contains the measures you selected is displayed.

WebLogic (JMX) Monitoring

This section explains prerequisites for monitoring WebLogic (JMX) Server, including configuring IIOP and JMX Management Server.

Enabling JMX Management Server

Enable the JMX Management Server in the Oracle WebLogic console at **Configuration** > **General** > **Advanced Settings**. Set both `Compatibility Mbean Server Enabled` and `Management EJB Enabled`. This enables both the legacy and the new JMX interface.

You must restart your system for these changes to go into effect.

Enabling and Configuring IIOP

You must enable and configure the IIOP protocol in Oracle WebLogic 11g R1 before you can monitor WebLogic (JMX).

1. In the Change Center of the Oracle WebLogic administration console, click **Lock & Edit**.
2. In the left pane of the console, expand `Environment` and select **Servers**.
3. Select the **Protocols** tab, then select `IIOP`.
4. Check the **Enable IIOP** check box to enable the IIOP protocol.
5. To modify the default configuration, click **Advanced**.
6. To specify a default IIOP user name and password:
   a) In the **Default IIOP User** field, enter a user name.
   b) In **Default IIOP Password** field, enter a password.
7. To activate these changes, in the Change Center of the administration console, click **Activate Changes**.

You must restart your system for these changes to take effect.

Monitoring WebLogic (JMX)

1. Choose **Monitor** > **Add Data Source** . The **Data Source** wizard opens.
2. Click the **Select from predefined Data Sources** option button.
3. Click **Next.** The **System selection** page opens.
4. Expand `Application Server` and select the WebLogic server version to be monitored.

   Select either `EJXM` or `JMX`.
5. Click **Next**. The **Connection parameters** page opens.
6. In the **Hostname** text box, specify the host on which WebLogic is running.
7. Specify the WebLogic **Port** (Default is `7001`).
8. Do not specify a user or password.
9. Click **Server Configuration**.
10. Specify the **Java Home directory** and the **Application Server install directory**.

    Use a UNC path, including the folder `lib`. Example: `\\<server name>\Oracle\Middleware\wlserver_10.3\server\lib`.

Note: If you can not specify a UNC path, copy `wljmxclient.jar` from the folder `<WL_HOME>\server\lib` to your local machine and add it to the classpath.

11. Click **OK**.
12. Click **Next**. The **JMX Data Source Browser** opens.
13. Select measures that you want to include in the initial monitor view.
14. Click **Finish**.

Monitoring Remote Machines

This section describes the requirements and setup of remote machines and provides troubleshooting tips.

Requirements for Monitoring Remote Machines

Windows 7 and Windows Vista target machines

The Remote Registry service must be running on the machine that you want to monitor. This service does not run by default on Windows Vista and Windows 7 machines.

Windows 2000 target machine

- User is in Administrators group
- User is in Users group and possesses the `Access this computer from the network` privilege

Windows XP target machine

- User is in Administrators group
- User is in Users group, and the path `Software\Microsoft` is added to the new network access setting: `Remotely accessible registry paths` under `Local Policies/Security Options`.

Windows 2003 target machine

- User is in Administrators group
- User is in the Network Configuration Operators and Performance Monitor Users groups
- User is in Users group, and the path `Software\Microsoft` is added to one of the following `Network access` settings:
  - Remotely accessible registry paths setting
  - Remotely accessible registry paths and subpaths under `Local Policies/Security` options setting

This approach works well for tests with machines in different trusted domains and with users from those domains.

For more information

Find additional information regarding monitoring from a URL at the following location:

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

Enabling Remote Monitoring for Windows NT

You must grant at least READ access to the following files:

- `"%windir%\system32\PERFCxxx.DAT`

- `"%windir%\system32\PERFHxxx.DAT`

Where `xxx` is the basic language ID for the system. For example, use `009` for English.

⚠ **Caution:** Using Registry Editor incorrectly can cause serious, system-wide problems that might require you to reinstall Windows. Use this tool at your own risk.

1. Using REGEDT32, grant the user account at least READ access to `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib` and all subkeys of that key.
2. Using REGEDT32, grant the user account at least READ access to `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg`.

Enabling Remote Monitoring for Windows XP

An error message might occur when trying to monitor a computer that is running the Windows XP operating system. If so, enable monitoring remotely.

1. Open Windows Explorer on the target Windows XP computer.
2. Choose **Tools** > **Folder Options** .
3. Click the **View** tab.
4. Uncheck the **Use Simple File Sharing** check box.
5. Click **OK**.
6. Verify privileges on the target Windows XP computer as follows:
   - If the system root is on an NTFS partition, you must possess at least READ access to the following files:
     - `%SystemRoot%\System32\Perfc009.dat`
     - `%SystemRoot%\System32\Perfh009.dat`
   - You must possess at least READ access to the following registry keys on the remote computer:
     - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg`
     - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib`
7. Reboot the Windows XP computer.

A user who possesses administrator rights to the remote computer can now view the remote counters.

Running Diskperf on a Windows Computer

The Windows parameter sets cannot collect information about the monitored computer unless the Disk Performance Statistics Driver (Diskperf) has been run on the monitored computer.

By default, Diskperf installs the Disk Performance Statistics Driver above the fault tolerant driver, `Ftdisk.sys`, in the I/O Manager disk-driver stack. Use the Devices Control Panel to determine if Ftdisk is started in your configuration.

1. Open a command prompt window.
2. At the command prompt, type one of the following commands and then press `Enter`:

Option	Description
`diskperf -y`	Enables the counters on a standard disk configuration.
`diskperf`	Indicates whether disk counters are enabled.

Option	Description
`diskperf -yv`	Enables disk counters on mirror and stripe sets and other noncontiguous partition sets and installs the performance statistics driver below the fault tolerant driver in the I/O stack.
`diskperf -n`	Disables disk counters.

Note: To run Diskperf on a computer, you must be a member of the Administrator’s local group on that computer. To monitor the physical disks in disk configurations that include Ftdisk, run the `diskperf -yv` command.

Running Diskperf on a Remote Computer

The Windows parameter sets cannot collect information about the monitored computer unless the Disk Performance Statistics Driver (Diskperf) has been run on the monitored computer. If you are using a remote computer, perform these steps to ensure that you can collect information.

1. Open a command prompt window.
2. Enter the command name, followed by the computer name.
   For example, enter `diskperf\\`*`thecomputername`*.
3. Restart the remote computer.

Grant READ Access to Perflib Subkeys

After the Performance Monitor (PerfMon) account connects to the registry of the monitored computer, the account on the monitored computer needs READ access to the subkeys that PerfMon uses. The computer uses the access control list (ACL) for the `Perflib\LanguageID` subkey to determine who has access to PerfMon data in the Registry, where *LanguageID* is the numeric code for the spoken language installed with Windows NT. The `Perflib\LanguageID` subkey is located in the following Registry path:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib
\LanguageID
```

Granting READ Access to Perflib Subkeys

By default, the Everyone group possesses READ access to the `Perflib\LanguageID` subkey. You might need to grant account permission to read the `Perflib\LanguageID` subkey if the default permissions on the monitored computer have changed.

1. On the computer that you want to monitor, start Regedt32.
2. Click the `LanguageID` subkey.
3. Choose **Security** > **Permissions** . The ACL for the subkey is displayed in the **Registry Key Permissions** dialog box.
4. Click **Add** to add the account to the ACL and grant READ permission.
5. Choose **Security** > **Permissions** to check the ACL for each subkey in the path to the LanguageID subkey.

Verify that the account possesses READ access to all subkeys in any ACLs that permit it.

Troubleshooting Remote Machines

The following error messages can occur when monitoring Windows-based machines:

- `Computer Name Not Found`
- `Diskperf Not Installed`

- No READ Access to Perflib Subkeys
- Unable to Connect to Windows XP Machine
- Error=AGENT_ERROR_TCP_SEND_REQUEST_FAILED

Computer Name Not Found

Possible Cause	Resolution
You might receive this error message when trying to monitor a remote computer from a non-administrator account.	Enable remote monitoring on a Windows NT computer.

Diskperf Not Installed

Possible Cause	Resolution
The Windows parameter sets cannot collect information about the monitored computer unless the Disk Performance Statistics Driver (Diskperf) has been run on the monitored computer.	Run Diskperf on the monitored Windows computer.

No READ Access to Perflib Subkeys

Possible Cause	Resolution
By default, the Everyone group possesses READ access to the `Perflib\LanguageID` subkey. You might need to grant account permission to read the `Perflib\LanguageID` subkey if the default permissions on the monitored computer have changed.	Grant an account READ permission to the `Perflib\LanguageID` subkey.

Unable to Connect to Windows XP Machine

Possible Cause	Resolution
This error can occur when monitoring computers that are running the Windows XP operating system.	Enable remote monitoring.

Error=AGENT_ERROR_TCP_SEND_REQUEST_FAILED

Possible Cause	Resolution
This problem can occur if your organization uses proxy auto-configuration files to direct requests to a proxy server, and you specify the automatic configuration script address rather than the registered name of the proxy server. If your organization uses a non-passive proxy, in most cases you must also specify the port number.	Verify the following proxy settings: • Correct entry of the proxy server name, user name, and password • Ability to ping the proxy server from the RSM computer • Ability of the RSM computer to connect to the portal computer

`System 5 - Access denied`

Possible Cause	Resolution
When monitoring a Windows 2008 machine using CUSTOM / PERFMON option, logging in with credentials of a user added to the Administrators group generates a `System 5 - access denied` error.	The user with which you run Performance Explorer needs to be added to the Administrators group on the remote computer that you want to monitor. If you specify different user credentials in Performance Explorer's **Data Source Wizard**, that user also needs to be added to the Administrators group on the remote computer. To monitor remote machines successfully, you need to launch Performance Explorer (`PerfExp.exe`) under the same user account that is being used to connect to the remote machine(s). **Tip:** To launch `PerfExp.exe` under a specific user account, press `shift + right-click` and select **Run as different user**.

Generating BDL Monitoring Projects

Silk Performer projects, or more precisely BDL scripts, can be used to collect performance data for monitored systems. This section outlines the steps that are required to assemble BDL monitoring projects for use with Performance Explorer and thereby enable real-time display of collected data.

Silk Performer BDL Monitoring Projects

The Performance Data Collection Engine (PDCE) processes performance data collected by BDL scripts. Performance data collected with such scripts can be displayed in Performance Explorer in real-time. Entities collected by the PDCE are called measures, which may be specified in BDL using the following functions:

- MeasureInc
- MeasureIncFloat
- MeasureStart & MeasureStop
- MeasureSetTimer
- MeasureSet

These functions are called within transactions. One-to-many measures are allowed within each transaction. Furthermore, monitoring projects cab use exactly one script defining exactly one usergroup. This usergroup may define one `init` transaction, one `end` transaction and one `main` transaction.

Wrapper functions, defined in `bdlMonitor.bdh`, are used to deliver additional functionality that is necessary for BDL realtime monitoring. See the Supplemental tutorials for information on using these functions:

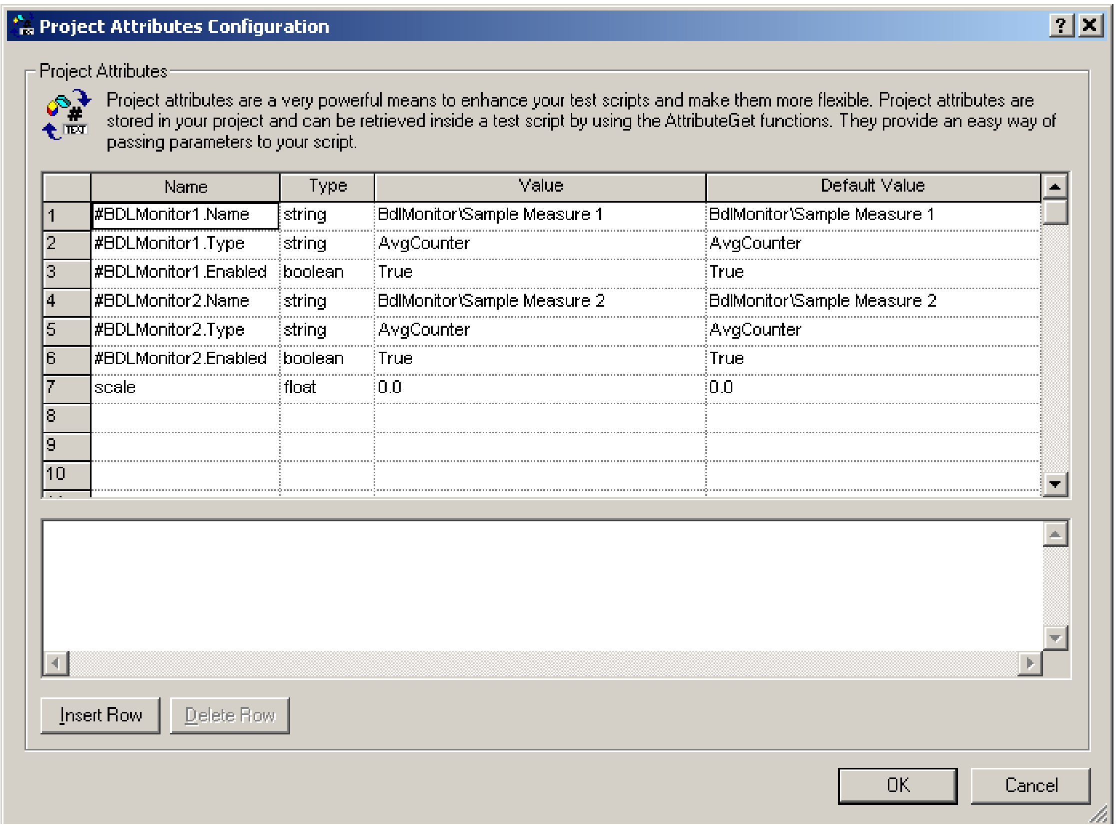
- MonitorInc
- MonitorIncFloat
- MonitorStart and MonitorStop
- MonitorSet
- MonitorSetFloat
- MonitorSetTimer

Creating a BDL Monitoring Project

1. Start Silk Performer and create a BDL monitoring project.
   a) Click the **Outline Project** button on the Workflow bar. The **Workflow - Outline Project** dialog box opens.
   b) Enter a name for the project in the **Name** text box and an optional description in the **Description** text box.
   c) From the **Application Type** list, select **Monitoring** > **Bdl Monitor for Performance Explorer**.
   d) Click **Next**.
2. On the **Model Script** dialog box, select the **Open existing script** option button.
3. Select the script `BdlMonitorSample.bdf` and click **OK**.

   This brings up a pre-configured monitoring script that can be used as a template.
4. Select **Project** > **Project Attributes** to define the project attributes.

   Each measure exported to Performance Explorer requires a fixed set of project attributes.



Project Attributes Configuration

Project Attributes

Project attributes are a very powerful means to enhance your test scripts and make them more flexible. Project attributes are stored in your project and can be retrieved inside a test script by using the AttributeGet functions. They provide an easy way of passing parameters to your script.

	Name	Type	Value	Default Value
1	#BDLMonitor1.Name	string	BdlMonitor\Sample Measure 1	BdlMonitor\Sample Measure 1
2	#BDLMonitor1.Type	string	AvgCounter	AvgCounter
3	#BDLMonitor1.Enabled	boolean	True	True
4	#BDLMonitor2.Name	string	BdlMonitor\Sample Measure 2	BdlMonitor\Sample Measure 2
5	#BDLMonitor2.Type	string	AvgCounter	AvgCounter
6	#BDLMonitor2.Enabled	boolean	True	True
7	scale	float	0.0	0.0
8				
9				
10				

To be viewed with Performance Explorer, each measure requires at least these three project attributes:

- **Name:** Name to be shown in Performance Explorer
- **Type:** An average value or a cumulative value.
- **Enabled:** Reserved. Always set to true.

5. Review the sample monitor script

   As specified in the project attributes, the project exports two measures that can be viewed in Performance Explorer in real-time.

Look for `MonitorInc` and `MonitorIncFloat` in the `TMon` transaction. This is where the last snapshot is handed over to Performance Explorer.

```
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. Edit the `.conf` file.

   The sample `.conf` file is available in the **Project** tree under the **Data Files** node. Double-click the `.conf` file to open it and change the value of the `Type` entry. This indicates where in the Performance Explorer hierarchy the project is to be located. For example, use `<Type>Monitoring\Sample Project</Type>` to locate the monitoring project in Performance Explorer under **Monitor** > **Add DataSource** > **Predefined data sources** at the location **Monitoring\Sample Project**.

7. Export the project to a single ZIP archive.

   Choose **File** > **Export Project**. Check the **Zip to single archive file** check box.

   For the **Export location**, set `C:\Program Files\Silk\Silk Performer\Monitors\BdlMonitorSample.sep`.

   Note: Save the file with an `.sep` extension.

8. Start a Realtime BDL Monitoring Project

   a) Start Performance Explorer and choose **Monitor** > **Add Data Source**. Select the **Select from predefined Data Sources** option button and click **Next**.

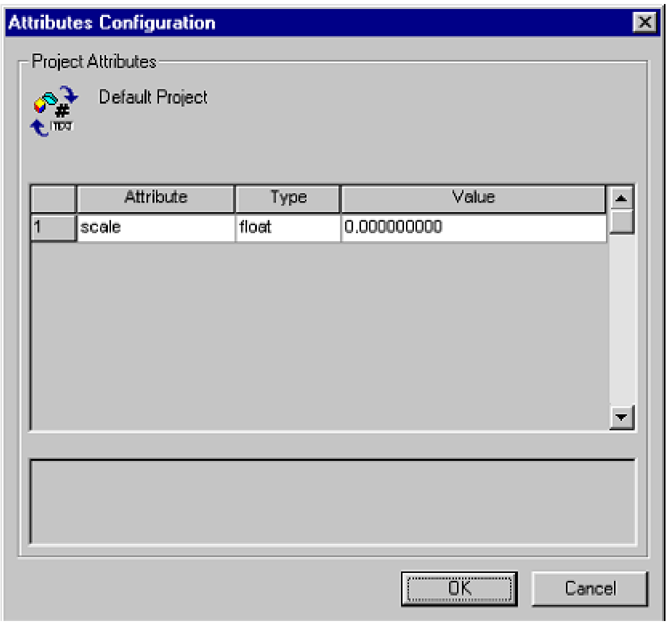
   b) Select the newly created monitoring project (for example, **Monitoring** > **Sample Project**) and click **Next**.

c) Enter the name of the host that is to be monitored (select `localhost` or the host recommended by Performance Explorer and click **Next**).

   This brings up a dialog on which project attributes can be modified (only one attribute can be modified in this instance).

d) Enter a value greater than `0`. This value will be used in the monitoring project. See the sample script for details.

e) Click **OK**. This brings up a choice of measures specified in the BDL monitoring project.



	Attribute	Type	Value
1	scale	float	0.000000000

f) Select both measures and click **Finish**. This launches the monitoring project. The values can now be seen in real-time using a monitoring graph.

Tutorial

This tutorial consists of two use cases: (1) a simple use case to get you started and (2) an advanced use case. The first use case involves a monitor project that consists of one user group with a single transaction and a single measure.

The second use case illustrates how to have several user groups, transactions, and measures in a single project.

Basic Use Case

In this use case you will create a project that keeps track of the number of processes running on a SunOs. SunOs can usually be accessed through remote execution. Compare with the Win2000 command line tool rexec. To count the number of processes running on a SunOs, execute `'ps -ef | egrep -c ".*"'` within an X-Terminal, Telnet session, rexec, or other preferred tool. For example, at a DOS prompt type:

```
c:\>rexec yourSunHost -l username "ps -ef | egrep -c \".*\""
```

This returns the number of processes running on your SunHost. The goal here is to continuously track and display this value in a Performance Explorer real-time chart.

Creating a New Project

Create a new project of application type `Monitoring/Bdl Monitor for Performance Explorer` as explained earlier in this section. Enter a simple project description that reflects the overall purpose of the monitoring project to be displayed in Performance Explorer - for example, "A powerful project used to collect the number of running processes on SunOs systems."

Planning Project Attributes

Begin by entering a name for the measure in the project attributes, for example `CountNrOfProcesses`. This name will also be used in the Performance Explorer hierarchy. A hierarchy is introduced with `'\'`.

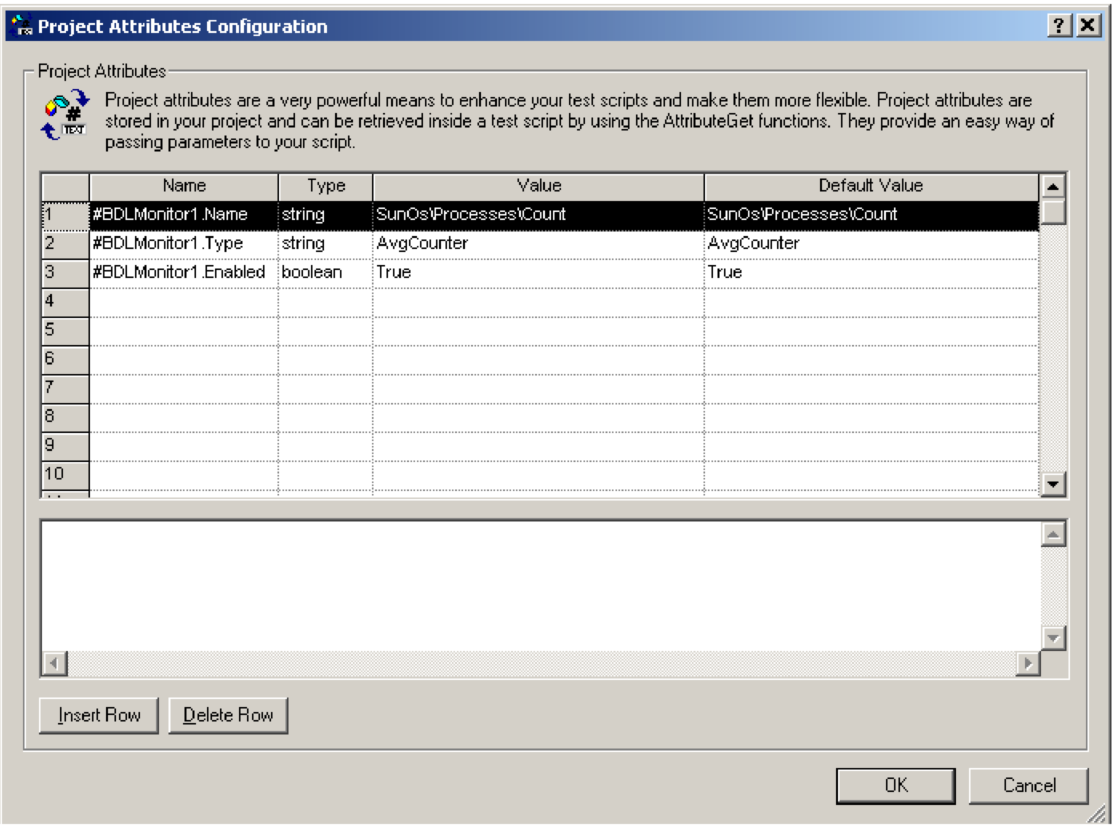
For example, to create this hierarchy:

```
OS
  Processes
     CountNrOfProcesses
     Another Counter
```

Specify the name of the measure as `SunOs\Processes\CountNrOfProcesses`.

This will be an `AvgCounter` (for average) counter and it should be `collected` by `default.`

These values must be transferred into the project's attributes. Open the attribute editor by choosing **Project** > **Project Attributes**.



Project attributes are a very powerful means to enhance your test scripts and make them more flexible. Project attributes are stored in your project and can be retrieved inside a test script by using the AttributeGet functions. They provide an easy way of passing parameters to your script.

	Name	Type	Value	Default Value
1	#BDLMonitor1.Name	string	SunOs\Processes\Count	SunOs\Processes\Count
2	#BDLMonitor1.Type	string	AvgCounter	AvgCounter
3	#BDLMonitor1.Enabled	boolean	True	True
4				
5				
6				
7				
8				
9				
10				

The values in the project attributes identify a single measure. Several attributes (lines in the dialog) may be required to create a measure however. For example, all entries beginning with #BDLMonitor1 in the above example. This is how settings are defined for a measure.

The measure is now defined with a name, a type, and the setting Enabled. A second measure will begin with #BDLMonitor2 (covered in the Advanced Use Case).

Creating a BDL Monitoring Script

To remotely execute certain command line tools in BDL, for example "ps" on a SunOs, three functions are required:

- Connect to a remote machine's "exec" server.
- Send the command, to be executed on the remote machine.

- Close the connection

```
// 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;

  WebTcpipSendBin(hCon, sSend, nSize);
  WebTcpipRecvExact(hCon, NULL, 1);
  WebTcpipRecv(hCon, sBuf, sizeof(sBuf), nRec);
  Request := number(sBuf);
end Request;

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

A function wrapper is needed around the Silk Performer `MeasureInc` functions. This function can be used in all monitoring projects. A function named `MonitorInc` is created to access project attributes. This function accesses the attributes you specified earlier.

The `MonitorInc` function can also be imported from an existing bdh, `bdlMonitor.bdh`.

```
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;
```

Now the transaction that will take the snapshot using all the functions that have been defined can be coded. This transaction also accesses the project file attributes. The goal is to later have these attributes set in Performance Explorer. For now however, to ensure that the script works, four attributes need to be added to the project attributes.

Note: Attribute names are case sensitive.

- `host:` Assign it a sample value, for example `sunserver`
- `command:` Assign it a sample value, for example `ps -ef | egrep -c ".*"`
- `user:` Assign a sample value, for example `root`
- `password:` Assign a sample value for testing purposes

Open the project attributes editor by choosing **Project** > **Project Attributes** and add these additional attributes. All are of type string except for the attribute password which is type password. Assign values to the attributes for testing purposes. Choose a description for each attribute that conveys the purpose of the attribute.

```
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;
```

The project now consists of the newly created script. Save the project and verify that it works by initiating a TryScript run. Have `nVal` printed or written to a log file to verify that the script works. If the script works, save and close the project.

Packaging the BDL Monitoring Script

To export the project as a single ZIP file, go to **File** > **Export Project**. Check the **zip to single archive** check box and export the file to `C:\Program Files\Silk\Silk Performer 10.0\Monitors\CountProcess.sep`. (Note the `.sep` extension).

Before the `.sep` file can be used with Performance Explorer the `.conf` file for the `.sep` must be modified. The `.conf` file looks like this:

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

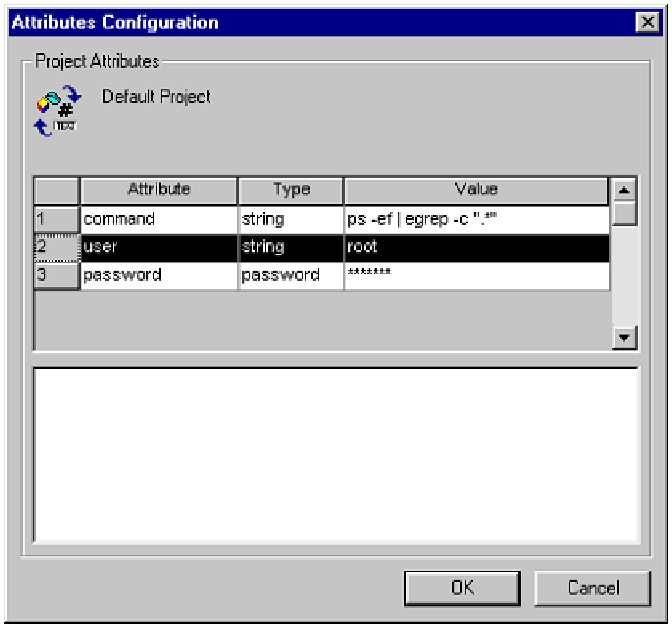
For the type setting, specify a hierarchy separated by "\". This hierarchy will be reflected in Performance Explorer.

Using the Monitoring Project inside Performance Explorer

1. Launch Performance Explorer and go to **Monitor** > **Add Data Source**. Select the **Select from predefined Data Sources** option button and click **Next**.
2. Select the newly created monitoring project and click **Next**.
3. Specify the host that is to be monitored, preferably a machine running SunOs. Click **Next** to bring up a dialog for setting project attributes. This dialog displays the attributes that you set (except for the host which was specified previously).

   Therefore, the dialog displays the following attributes:

   - user
   - password
   - command



Attributes Configuration

Project Attributes

Default Project

	Attribute	Type	Value
1	command	string	ps -ef | egrep -c ".*"
2	user	string	root
3	password	password	*******

4. Select the measures (in this case only a single measure) and click **OK**. The measures are then immediately handed over to the data collection mechanism.

Advanced Use Case

This use case includes a project with two user groups, the user group from the previous use case and a new user group that measures HTTP requests. The two URLs and host are configurable through project attributes.

Creating a Silk Performer Project

Enter a simple project description that is to be displayed in Performance Explorer. The description should describe the overall purpose of the monitoring project. For example, a `powerful project used for collecting the number of running processes on SunOs systems and measuring two configurable HTTP requests.`

Planning Project Attributes

For this tutorial you will plan three measures. Pick measure names that indicate the hierarchy and purpose of the measures. In this example, a measure counts the number of processes on a SunOs system and includes two measures that measure HTTP requests (Performed using `WebUrl`):

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

The intended hierarchy will then look like the following:

```
SunOs
  Processes
    CountNrOfProcesses
  Http Server
    URI1
    URI2
```

A BDL script with two user groups and two transactions is written. See the table below to see how measures are assigned to transactions and how transactions relate to user groups.

User Group	Transaction	Measure
VMonitor	TSnap	CountNrOfProcesses
VWebMon	TWebSnap	URI1
VWebMon	TWebSnap	URI2

With projects that include numerous transactions and/or user groups, measures must be specified with the corresponding user group and assigned measure. This differs from the basic use case outlined previously, which involved only one transaction and one user group.

Open the project attributes editor at **Projects** > **Project Attribute** and enter the following data:

Name	Type	Value
#BDLMonitor1.Name	string	SunOs\Processes \CountNrOfProcesses
#BDLMonitor1.Type	string	AvgCounter
#BDLMonitor1.Enabled	boolean	True
#BDLMonitor1.Script	string	remote.bdf
#BDLMonitor1.Usergroup	string	VMonitor
#BDLMonitor1.Transaction	string	TSnap
#BDLMonitor2.Name	string	SunOs\Http Server\URI1

Name	Type	Value
#BDLMonitor2.Type	string	AvgCounter
#BDLMonitor2.Enabled	boolean	true
#BDLMonitor2.Script	string	remote.bdf
#BDLMonitor2.Usergroup	string	VWebMon
#BDLMonitor2.Transaction	string	TWebSnap
#BDLMonitor3.Name	string	SunOs\Http Server\URI2
#BDLMonitor3.Type	string	AvgCounter
#BDLMonitor3.Enabled	boolean	True
#BDLMonitor3.Script	string	remote.bdf
#BDLMonitor3.Usergroup	string	VWebMon
#BDLMonitor3.Transaction	string	TWebSnap
host	sunserver	sunserver
command	string	ps -ef | egrep -c ".*"
user	string	root
password	password	***********
URI1	string	**/**
URI2	string	/manual/ibm/index.html

Creating a BDL Monitoring Script

To remotely execute certain command line tools in BDL, for example "ps" on a SunOs, three functions are required:

- Connect to a remote machine's "exec" server.
- Send the command, to be executed on the remote machine.
- Close the connection

```
// 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;

  WebTcpipSendBin(hCon, sSend, nSize);
  WebTcpipRecvExact(hCon, NULL, 1);
  WebTcpipRecv(hCon, sBuf, sizeof(sBuf), nRec);
  Request := number(sBuf);
end Request;

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

A function wrapper is needed around the Silk Performer `MeasureInc` functions. This function can be used in all monitoring projects. A function named `MonitorInc` is created to access project attributes. This function accesses the attributes you specified earlier.

The `MonitorInc` function can also be imported from an existing bdh, `bdlMonitor.bdh`.

```
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;
```

Now the transaction that will take the snapshot using all the functions that have been defined can be coded. This transaction also accesses the project file attributes. The goal is to later have these attributes set in Performance Explorer. For now however, to ensure that the script works, four attributes need to be added to the project attributes.

Note: Attribute names are case sensitive.

- `host:` Assign it a sample value, for example `sunserver`
- `command:` Assign it a sample value, for example `ps -ef | egrep -c ".*"`
- `user:` Assign a sample value, for example `root`
- `password:` Assign a sample value for testing purposes

Open the project attributes editor by choosing **Project** > **Project Attributes** and add these additional attributes. All are of type string except for the attribute password which is type password. Assign values to the attributes for testing purposes. Choose a description for each attribute that conveys the purpose of the attribute.

```
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;
```

The project now consists of the newly created script. Save the project and verify that it works by initiating a TryScript run. Have `nVal` printed or written to a log file to verify that the script works. If the script works, save and close the project.

Packaging the Project

Export the project to a single ZIP file. Go to **File** > **Export Project**. Check **Zip to single archive** and export the file to `C:\Program Files\Silk\Silk Performer\Monitors\Advanced.sep` (note the SEP file extension).

Before the SEP file can be used with Performance Explorer, the CONF file for the SEP must be modified. The CONF file appears as:

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

For the type setting, specify a hierarchy that is to be reflected in Performance Explorer, separated by backward slashes ("\").

Using the Monitoring Project in Performance Explorer

1. Start Performance Explorer and go to **Monitor** > **Add Data Source**. Choose **Select from predefined Data Sources**.

2. Select the newly created monitoring project and click **Next.**
3. Enter the name of the server that is to be monitored. For example, enter `sunserver` and click **Next**. A dialog box appears. Define the following attributes:
    - **command:** As with the basic test case you set up earlier in this tutorial, a command that counts the number of processes.
    - **user:** A valid user account for accessing the remote machine.
    - **password:** The password.
    - **URI1:** The URI where response time is to be measured.
    - **URI2**: A second URI where response time is to be measured.
4. Select the measures (in this case there is only one) and click **OK**. The measures are immediately handed over to the data collection mechanism. The choice corresponds to the hierarchy defined in the project attributes.

Best Practices

This section outlines suggestions for working with real-time monitoring projects as relates to the writing of monitoring scripts.

Loops

Performance Explorer executes transactions that run snapshots at regular intervals. For example, a snapshot transaction might be executed every 10 seconds. In such a case, Performance Explorer assumes that the entire transaction is executed within this one interval. Otherwise Performance Explorer returns a warning stating that it can not collect the interval's snapshot.

Do not use endless loops such as the following:

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

ThinkTimes and Wait

Be careful with wait statements. Think-times are ignored. A monitoring transaction must be executed within a given interval. Wait statements block execution and transactions may not return on time.

Example

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

Initialization

Do not place function calls in `TInit` transactions that may stop runtime. For example, `OraLogon` may generate a `ProcessExit` on a failed logon. Performance Explorer will not catch this message and will assume that the project is still running. Of course, Performance Explorer will report that it cannot gather data, but it will not be able to report why this is happening.

Other Best Practices

Name Length

Attribute value lengths must not exceed 79 characters.

Performing a Try Script run for the Project

Execute a Try Script run to verify that the project is working before you create an SEP file.

Special Project Attributes

The monitoring-project attribute `host` can be accessed within BDL scripts using the following function call:

```
AttributeGetString("host", sHost, sizeof(sHost));
```

Instead of `host`, an attribute called `#MonitorHost` may also be queried. Those two attributes are treated equally.

In Performance Explorer the value for this attribute is set using the following dialog:

Collecting Data

This section describes how to assign a monitoring template to a project and describes the types of data available to include in charts and reports.

Setting Monitoring Options

Silk Performer enables you to define monitoring templates for your projects and to configure projects to automatically launch Performance Explorer when load tests begin.

1. In Silk Performer, expand the **Profiles** node in the menu tree.
2. Right-click the profile that you want to configure and choose **Edit Profile**.

   Tip: Alternatively, you can choose **Settings** > **Active Profile** from the menu bar.

   The **Profile - [<profile name>]** dialog box opens, and the **Replay** category is displayed in the shortcut list on the left.
3. In the shortcut list, click the **Results** icon.
4. Click the **Monitoring** tab.
5. In the **Monitoring options** area, check the **Automatically start monitoring** check box to automatically launch Silk Performer's monitoring facility when the test starts.
6. To automatically use the monitoring template that best suits the project, click the **Use default monitoring template** option button.
   For example, if you are creating a Web project, the template specifies the measurements that are useful for Web load tests.
7. To use a custom monitor template, click the **Use custom monitoring template** option button and perform one of the following steps:

   - Type the name of the custom template file (`.pew`) that you want to use to monitor your server. Silk Performer creates a copy of the standard monitor template.
   - Click the folder icon in the name field to select an existing monitoring template.
8. *Optional:* Click **Edit Custom Monitor Template** to add or remove any monitoring performance data.

   When you click this button, Silk Performance Explorer opens. Perform the following steps:

a) Add or remove any monitoring performance data.
b) Save the Silk Performance Explorer workspace to apply your changes to the template.

9. In the **Performance Monitor integration** area, check the **Compute online performance data** check box to compute data for additional performance measurements to be displayed in the Windows Performance Monitor.

   You can use this data to view concurrent users, transaction throughput, sent and received data, and executed SQL statements.
10. Click **OK** to save your settings.

When you perform a test, Silk Performer displays the server performance data that is relevant to the type of server under test.

Client-Side Measures

With Performance Explorer, you can collect data for client-side monitoring in addition to server data.

Configuring Performance Explorer

To use real-time client-side measures, such as in built-in graphs and monitor reports, add a connection entry for the client-side data source.

1. Choose **Monitor** > **Add Client-Side Data Source**. The **Data Source Wizard** opens.
2. In the **Hostname** text box, specify the controller machine name.
3. In the **Alias** text box, specify the alias for the controller machine that provides the client-side measures.
4. Click **Finish**. A connection entry for the data source appears in the menu tree. The client-side measures that are available on this data source, such as page timers, are listed in folders in the menu tree.

   Note: Check the **Enable real-time measures** check box in the **Workload Configuration** dialog, which instructs the runtime system to supply host data measures. A green data-source connection icon on Performance Explorer's **Client-Side Measures** page indicates a successful connection to the data source. A red data-source connection icon indicates that the data source is not connected.

Removing Client-Side Measures

You can remove a client-side measure node from the Performance Explorer menu tree. However, Silk Performer replaces the measure node if it receives updated information for the measure.

1. Click the **Client-Side Measures** pane.
2. Right-click the measurement node that you want to delete and choose **Remove Client-Side Data Source**. A confirmation dialog box prompts you to remove the node.
3. Click **Yes**.

Monitor Types

Describes the purpose of monitors and lists the types available.

Monitor Types Overview

Monitors collect performance counters on systems and poll their measurements at specified intervals. Define a performance counter as a data source to include it in a monitor.

Monitor types include the following objects:

- Graphs – Display measured counters in a graph.

- Reports – Write measured counters to a TSD file.

Polling Intervals

If your agents and servers are heavily used, polling for performance measures will cost CPU resources. Choosing a short interval on a highly used system might influence test results.

Note: For large load tests, choose intervals greater than 60 seconds.

Server-side vs. Client-side Measurement Comparison

To compare server-side measurements with client-side measurements after running a load test, monitor reports must be defined.

Monitor reports write a time-series data (TSD) file that you can include in an overview report or use as the basis for a graph. Monitor graphs are useful for providing a graphical overview of the state of a system.

Note: Data shown in monitor graphs cannot be written to a TSD file and, therefore, cannot subsequently be included in overview reports.

Creating a Monitor Graph

Monitor graphs are useful for providing a graphical overview of the state of a system.

1. Choose **File** > **New** > **Monitor Graph** .
2. To add more than one graph, right-click a measurement in the **Monitor** menu tree and choose **Add To View**.

   Alternative: Drag and drop measures from the **Monitor** menu tree to the monitor graph.
3. Right-click the host name in the monitor graph and choose **Properties**. The **Properties** dialog box opens.
4. Click the **Monitor** tab.

   Monitor graphs include intervals that define the refresh rate for retrieving new values.
5. In the **Interval** boxes, specify how often intervals are saved.
6. In the **Page size** box, specify how many intervals show on each page.
7. In the **History size** box, specify how many intervals to store in the graph history.
8. Click **OK**.

Creating a Monitor Report

A monitor report can be defined to capture measures during load tests. Monitor reports write a time-series data (TSD) file that you can include in an overview report or use as the basis for a graph.

1. Choose **File** > **New** > **Monitor Report** .

   Note: For SAPGUI data sources, it is recommended that you change the monitor interval to 20 seconds or higher. This can be done via the Monitor Report's **Options** dialog on the **TSD-Writer** tab.
2. To add more than one measure to the report, right-click a measurement in the **Monitor** menu tree and choose **Add To View**.

   Alternative: Drag measures from the **Monitor** menu tree to the report.
3. Choose **Monitor** > **Write Monitor Data**.

   Alternative: Click **Write Monitor** on the toolbar.

   The **Save As** dialog box opens.
4. In the **File name** box, type a file name.
5. In the **Save as type** box, ensure that **Monitor Time Series File (r@*.tsd)** is specified.

6. Click **Save**. If you are running a load test and the workspace you have assigned to your profiles contains a monitor report, all monitor reports begin writing automatically when the test starts and stop when the load test finishes.

All monitor reports are stored in the result directory of the load test.

Host Data

Performance Explorer provides wizards for renaming, duplicating, and updating host data.

Duplicating/Renaming Measures

Duplicate or rename measures to monitor similar measures on different hosts.

The Duplicate feature copies measures from one host to another. Renaming means that the host name of the measures is changed.

1. Right-click a hostname in the **Monitor** pane and then choose **Rename host** or **Duplicate host**. The **Duplicate Host** wizard opens to the **Connection parameters** page.
2. In the **Hostname** text box, type a name for the data source to be duplicated or renamed.
3. *Optional:* In the **Alias** text box, specify the alias name.

   The alias should be a more descriptive synonym for the monitored server. It is also useful to group measures on a particular machine.

   For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.

   Note: Measures are grouped into measure groups when they use the same data source and connection parameters. For example, if you have PerfMon measures with different credentials, you have different measure groups.
4. Click **Next**. Depending on the specified data source, fields with various connection parameters appear. These fields are automatically populated with the original values.

   If there are configurable attributes for the specified data source, a **Change Attributes** button appears.
5. *Optional:* Check **Ignore measures** in the group check box if the measures are to be ignored.
6. Click **Finish**. The measures are checked to confirm validity. If invalid measures are discovered, you have the option of changing the connection parameters or ignoring the measures.

   Note: It is not possible to duplicate or rename invalid measures.

Changing User Names and Passwords for Existing Monitors and Workspaces

The security policies of many companies require that user passwords expire at regular intervals. Such policies can create inconveniences in automatic access of tools like Performance Explorer. For such environments, Performance Explorer enables you to change user names and passwords for existing monitors and workspaces.

1. Right-click the hostname for which you want to edit credentials in the **Monitor** pane and then choose **Edit credentials for host**. The **Edit credentials for host** wizard opens.
2. Verify that the hostname and alias match the host for which you want to modify credentials and then click **Next**.The **Change Attributes** button is displayed on the **Connection parameter** page.
3. Click **Change Attributes**.The **Attributes Configuration** dialog box opens.
4. To change the user name, in the **User** row, change the **Value** cell.
5. To change the password, in the **Password** row, change the **Value** cell.
6. Click **OK**.

7. Click **Finish**.

Measurements

A *measure*, also called a *measure type*, is a generic term for a measurement of performance taken during a load test. Measures can be either timers or counters. *Timers* collect data related to response times, and counters collect data on throughput and load-test events.

Measurements can be divided into *groups*. Each measurement group contains a number of scripted *measurement types* that are related to a specific use or a certain type of application.

Performance Explorer provides the following measurement types:

- Summary measurements
- Transactions measurements
- Web form measurements
- Web page measurements
- CORBA-specific measurements
- SQL Database-specific measurements
- Middleware-specific measurements
- Citrix-specific measurements
- Custom timer measurements
- Custom counter measurements

Summary Measurements

The Summary measurement group contains summarized measurements on a global level. It contains measurement types that aggregate individual measurements from other measurement groups as well as measurement types that represent information on a global level that are not included in other measurement groups. The summary measurement group contains only counters; it does not contain timers. It represents a fixed set of global measurements that are collected only during the measurement period, thus excluding the warm-up and shutdown period of the load test. The Summary group contains measurement types that are available as real-time data and time-series data (TSD).

General Measurements
Displays the measurement types that are shown at the beginning of the overview report.

Measurement	Description
Active users	The number of active virtual users. A virtual user is regarded as active if it has been started and is currently in one of the following states: executing, warm up, wait database, loading documents, wait resource, rendezvous, thinktime, and shutdown.
Transactions	The number of Silk Performer transactions that have completed, regardless of whether or not they were successful. The overview report shows the following information: • A graph indicating how many transactions ended in a given second. • Average number of ended transactions per second. • Total number of ended transactions.
Errors	The number of API errors, including Internet, database, and middleware APIs. An API error is counted if the severity is `error` or higher (`transaction exit` or `process exit`). An error is ignored if the severity is `informational` or `warning`.

Agent Health Control Measurements
Describes the measurements that are used for analyzing an agent's health.

Measurement	Description
CPU utilization	The percentage of time that a processor is busy executing tasks. If the processor utilization is equal to or near 100%, the computer has run out of available processing capacity.
Memory	The percentage amount of memory allocated (committed) by the driver machine in relation to the total amount of physical RAM. High memory utilization indicates that the driver machine is low on available memory, and this situation can cause disk swapping, which degrades the overall performance of the computer. **Note:** Task Manager may show different memory statistics depending on your operating system.
Responsiveness	For most Silk Performer project types the responsiveness measure is determined directly by the internal measure *System Health*. For Silk Performer projects of type SAP GUI or browser-driven load testing (BDLT), the responsiveness measure also takes into account the health of the application under test (SAP GUI or `perfBrowserHost.exe`). This health is indicated by the internal *App Health* value. The overall *Responsiveness* measure corresponds to the worse of these two health values.

Internet-Related Measurements

Describes the measurements that are related to the internet connection.

Measurement	Description
Concurrent connections	The number of currently open TCP/IP connections.
Requests sent	The number of HTTP requests sent by browser-level functions, the number of data packets sent by TCP/IP-level functions, or the number of commands sent by POP3, SMTP, FTP, or LDAP function calls. Silk Performer also includes LDAP traffic in this measurement; it reports the number of LDAP requests sent. IIOP uses other counters.
Requests failed	The number of the requests listed above that are not sent or executed successfully. IIOP uses other counters.
Request data sent	The amount of data (in KB) sent to the server, including header and body content information as well as all TCP/IP-related traffic (HTTP, native TCP/IP, IIOP, POP3, SMTP, FTP, and LDAP), and secure traffic over SSL/TLS. It does not include data overhead caused by encryption using SSL/TLS or other API-related traffic like TUXEDO ATMI, ODBC, and Oracle OCI.
Responses received	The number of HTTP responses received by browser-level functions, the number of data packets received by TCP/IP-level functions, and the number of responses received by POP3, SMTP, FTP, and LDAP functions. IIOP uses other counters.
Responses failed	The number of the expected responses that are not received successfully. IIOP uses other counters.
Response data received	The amount of data (in KB) received from the server, including header and body content information as well as TCP/IP-related traffic (HTTP, native TCP/IP, IIOP, POP3, SMTP, FTP, and LDAP), and secure traffic over SSL/TLS. It does not include API-related traffic like TUXEDO ATMI, ODBC, and Oracle OCI.
Connects successful	The number of successful TCP/IP connects to the remote hosts established by browser-level, TCP/IP-level, IIOP, SMTP, POP3, FTP, or LDAP functions.

Measurement	Description
Connects failed	The number of TCP/IP connection failures caused by invalid host names, invalid IP addresses, invalid ports, unavailable remote hosts, network-related problems, or authentication failures. This counter is used by browser-level, TCP/IP-level, IIOP, SMTP, POP3, FTP, and LDAP functions. By default, browser-level functions perform up to three connect attempts (up to two retries). Only if all attempts fail, a failure is recorded.
Connects retries	The number of attempts to connect to the server if the initial connection fails.
Page data	The amount of data (in KB) for all requests and responses related to the request and retrieval of an entire Web page, including all embedded objects.
Embedded objects data	The amount of data (in KB) for all requests and responses related to the request and retrieval of objects embedded in a Web page.

Web-Specific Measurements

Describes the measurements that are specific for the Web.

Measurement	Description
HTTP request retries	The number of times a client is forced to retry a request. This may happen when the server closes a persistent connection (available with Keep-Alive or HTTP/1.1) because of a timeout occurring while the client tries to send a request. This counter is used only by browser-level functions.
HTTP redirections	The number of HTTP redirections performed due to a Web server response with status code 3xx. This measurement is incremented only by Web functions.
HTTP re-authentications	The number of re-authentications necessary because of a status code 401 or 407 returned by the Web server. This counter is used only by Web functions that perform basic authentication.
HTTP hits	The number of HTTP requests that arrive at the Web server. This measurement is incremented only by Web functions.
Hits failed	The number of failed HTTP requests. This measurement is incremented only by Web functions.
HTTP 1xx responses	The number of HTTP responses containing status codes indicating that their content has the status `informational`. This measurement is incremented by Web functions.
HTTP 2xx responses	The number of HTTP responses containing status codes indicating successful requests. This measurement is incremented by Web functions.
HTTP 3xx responses	The number of HTTP responses containing status codes indicating redirected requests. This measurement is incremented by Web functions.
HTTP 4xx responses	The number of HTTP responses containing status codes indicating client errors. This measurement is incremented by Web functions.
HTTP 5xx responses	The number of HTTP responses containing status codes indicating server errors. This measurement is incremented by Web functions.
HTTP cache conditional reloads	The number of requests sent to the server to download an item if its content has changed since it was last placed in the cache.
HTTP cache hits	The number of times requests that are fulfilled from the cache.

Measurement	Description
HTTP cookies sent	The number of cookies sent by browser-level functions. If the header line of an HTTP cookie request contains multiple cookies, each cookie in the header is counted.
HTTP cookies received	The number of cookies received by browser-level functions. If the HTTP response includes multiple cookie header lines, each cookie of these header lines is counted.
HTTP pages	The number of HTML pages requested by the page-level API or by the browser-level API when Java script simulation is enabled.

IIOP-Related Measurement Types

The following table describes the scripted measurements which relate to IIOP operations.

Measurement	Description
IIOP messages	The number of IIOP messages sent by Silk Performer.
IIOP requests	The number of IIOP request messages sent by Silk Performer.
IIOP loc requests	The number of IIOP locate request messages sent by Silk Performer.
IIOP fragments sent	The number of fragmented IIOP messages sent by Silk Performer.
IIOP replies OK	The number of IIOP reply messages with status `OK` received by Silk Performer.
IIOP replies exception	The number of IIOP reply messages with the status `exception raised` received by Silk Performer. This count includes system exceptions as well as user exceptions.
IIOP replies obj moved	The number of IIOP reply messages with the status `object has moved`. This status message causes Silk Performer to reissue the request to the new location.
IIOP request timed out	The number of times the time limit for receiving data (specified in the load-testing script) was exceeded while waiting for a pending reply. Whenever the time limit is exceeded, no reply is received and Silk Performer sends an IIOP `Cancel Request` message to inform the server about the aborted operation.
IIOP loc replies obj here	The number of IIOP `Locate Reply` messages with status `object here` received by Silk Performer.
IIOP loc replies obj unknown	The number of IIOP `Locate Reply` messages with status `object unknown` received by Silk Performer.
IIOP loc replies obj moved	The number of IIOP `Locate Reply` messages with status `object moved` received by Silk Performer.
IIOP message errors	The number of IIOP error messages received by Silk Performer.
IIOP fragments received	The number of fragmented IIOP messages received by Silk Performer.
IIOP loc request timed out	The number of times that the time limit for receiving data (specified in the load-testing script) was exceeded while waiting for a pending locate reply. Whenever the time limit is exceeded, no locate reply is received and Silk Performer sends an IIOP `Cancel Request` message to inform the server about the aborted operation.

Database-Specific Measurements

The following table describes the measurement types that are specific to the database.

Measurement	Description
SQL commands	The number of SQL commands sent to the DBMS for execution.
Open database logins	Counts the currently open database sessions. This counter is relevant for the OCI, ODBC, and ODBC high-level interfaces (This measurement is not available in report files).
Open database cursors	Counts the currently opened database cursors (handles). This counter is relevant for the OCI, ODBC, and ODBC high-level interfaces. For ODBC, only OdbcClose calls with the SQL_DROP option decrease this measure value. (This measurement is not available in report files.)

Transaction Measurements

The Transaction measurement group contains response-time data on the transaction level. For each transaction defined in your load-test script, a measurement of this group type is automatically created, using the name of the transaction as the key for the measurements. Only measurements that contain a counted value are displayed. This fact means that `Trans.failed` is not displayed if none of the transactions fail.

Note: The transactions `TInit` and `TEnd` do not count towards the load-test measurements because beginning and ending transactions should not contain measurements that relate to the application under test.

The following table describes the corresponding measurement types.

Measurement	Description
Trans. ok	The response time for successful transactions, in seconds. A transaction time is reported in this measure type if no API function call performed within the transaction returns an error message.
Trans.(busy) ok	The busy time for successful transactions, in seconds. A transaction busy time is the transaction response time without any think time. A transaction time is reported in this measure type if no API function call performed within the transaction returns an error message.
Trans. failed	The response time for failed transactions, in seconds. A transaction time is reported in this measure type if at least one error of severity `error` or higher occurs within the transaction.
Trans.(busy) failed	The busy time for failed transactions, in seconds. A transaction busy time is the transaction response time without any think time. A transaction time is reported in this measure type if at least one error of severity `error` or higher occurs within the transaction.
Trans. canceled	The response time for canceled transactions, in seconds. A transaction time is reported in this measure type if a return statement in the script is issued with a return value of `1`.

Web Form Measurements

The Web Form measurement group contains the measure types related to forms declared in the `dclform` section of a load-test script. The Web Form measurement group provides response time and throughput rates for form submissions with the POST, GET, and HEAD command sent by scripting function calls. For each form declared in the dclform section, a measurement group is created automatically with the name of the form as the key. The following table describes the corresponding measurement types:

Measurement	Description
Server busy time	Measurement, in seconds, that starts after the user has sent the complete request and ends when the user receives the first part of the response.
Round trip time	The time, in seconds, that it takes to send a request and receive the appropriate response. It also includes the connect and shutdown time when the connection is not persistent (the connection can persist if the Keep-Alive option is enabled or HTTP/1.1 is used).
Hits	The number of successful form submissions.
Request data sent	The amount of data (in KB) sent to the Web server during submission of a form. This measurement also includes HTTP header and body content information. The count specifies the number of Web server requests sent.
Response data received	The amount of data (in KB) received from the Web server in response to the submission of a form. This measurement also includes HTTP response header and body content information. The count specifies the number of Web server responses received.

Web Page Measurements

The Web Page measurement group contains the measurement types related to Web pages. For each Web page that is downloaded during a load test, a measurement group is created automatically with the name of the Web page as the key. The following table describes the corresponding measurement types:

Measurement	Description
Page time	The time, in seconds, that it takes to retrieve an entire Web page from the server, that is, for retrieving all HTML documents and all embedded documents, such as images, videos, and audio files.
Document download time	The time, in seconds, that it takes to retrieve all HTML documents of a Web page.
Server busy time	The time, in seconds, that the server takes to process a Web page request. This measurement starts after a virtual user has sent the last byte of the request and ends when the user retrieves the first byte of the response.
Page data	The amount of data (in KB) for all requests and responses related to the request and retrieval of an entire Web page including all embedded objects.
Embedded objects data	The amount of data (in KB) for all requests and responses related to the request and retrieval of objects embedded on a Web page.

CORBA-Specific Measurements

The IIOP measurement group contains the measurement types related to IIOP operations that are called within a load-testing script. It provides response time and throughput ratings for the IIOP operation calls issued by the API function `IiopRequest`. With the `IiopObjectSetMeasures` function you can specify a prefix to the name of the measurement groups associated with the operations of a certain CORBA object. If you decide not to do so, the name of the operation is used as the name of the corresponding measurement group. The following table describes the corresponding measurement types:

Measurement	Description
Round trip time	The time measured, in seconds, from when the processing of the `IiopRequest` function call begins to when the processing of the reply of the CORBA object ends.
Server busy time	Measurement, in seconds, that starts after a user has sent the last byte of the CORBA operation call and ends when the user retrieves the first byte of the response. **Note:** Depending on the content of the application and the request, server busy time might begin before the last byte is sent and continue after the user has received the first byte. Therefore, server busy time is, in many cases, only an approximation.

For non-blocking IIOP requests, these measurements are performed in the same way. This approach can lead to a situation where the sum of all your measured times exceeds the total time of your load test.

If a CORBA object decides to redirect a request, the `IiopRequest` function has to reissue this request to a different location. The result is two request-reply message pairs, where both are measured separately. As in the case of the measurement times for the `IiopRequest` function, the sum of both (separately measured) request-reply pairs is supplied.

SQL Database-Specific Measurements

The following table describes the corresponding measurement types:

Measurement	Description
Parse	The time, in seconds, that it takes to prepare (Oracle terminology: parse) a SQL command. Preparing a SQL command usually includes a server round-trip during which the SQL command is sent to the server. The SQL command is then parsed and transformed into an internal format that can be reused on subsequent calls of that SQL command. **Note:** If you use the `OraParse` function with the `ORA_DEFERRED` option, the parsing of the SQL command is deferred to the execute step, and the server round-trip time is produced for the parse step. In this case, this measurement type is meaningless.
Exec	The time, in seconds, in which a SQL command is executed. Executing a SQL command includes at least one server round-trip for sending the command and returning the result to enable the SQL command to be executed. Because of the considerable amount of data processing required on the server for a SQL command, executing SQL commands usually generates a major part of the load on the database server. A separate measurement type in the SQL measurement group is used for fetching data. Fetching data can either be included as part of the execute step or in a separate step. In the latter case, results are not included in the SQL measurement group.
ExecDirect	The time, in seconds, in which a SQL command is prepared and executed.

Middleware-Specific Measurements

The TUXEDO measurement group contains the measurement types related to TUXEDO service calls performed in a load-testing script. It provides response time and throughput ratings for the TUXEDO service calls issued by the functions `Tux_tpcall`, `Tux_tpacall`, `Tux_tpsend`, `Tux_tprecv`, `Tux_tpgetrply`, `Tux_tpenqueue`, and `Tux_tpdequeue`. For each TUXEDO service used in these TUXEDO functions, a measurement group is created automatically with the name of the service as the key. The following table describes the corresponding measurement types:

Measurement	Description
Response time	The response time, in seconds, for a TUXEDO service call. For synchronous service calls, the response time includes the time the server needs to process the call. For asynchronous service calls, Silk Performer reports the time that elapses between sending the request and receiving the reply, plus the time spent waiting for the request. Asynchronous service calls do not measure the time a server needs to execute the service. For send and request calls, Silk Performer measures the time for sending data across an open connection as well as the time for receiving data from an open connection. For queuing communication, both the time required for sending the service call to the queue and the time for retrieving the results of a service call are measured.
Request data sent	The amount of data (in KB) sent to the TUXEDO server for submission of a given service. Silk Performer measures the application data buffer size used by the service calls rather than the transferred bytes on the network.
Response data received	The amount of data (in KB) received from the TUXEDO server by a given service. Silk Performer measures the application data buffer size used by the service calls rather than the transferred bytes on the network.

Citrix-Specific Measurements

The Citrix Synchronization Points measurement group states how many synchronization functions already have been executed. The main purpose of this measurement is to visualize the progress of a transaction.

Measurement	Description
Citrix Synchronization points	The number of executed Citrix synchronization functions (CitrixWaitForXXX).

Custom Timer Measurements

The Custom Timer measurement group contains the timers that are defined in the load-testing script with the `MeasureStart` and the `MeasureStop` functions, as the following table shows.

Measurement	Description
Response time	The time, in seconds, measured between `MeasureStart` and `MeasureStop` function calls.

Note: You can pause and resume timers with the `MeasurePause` and the `MeasureResume` function.

Custom Counter Measurements

The Custom Counter measurement group contains the counters that are defined in the load-testing script with the `MeasureInc` function, as the following table shows.

Measurement	Description
Custom counter	Counts values that are set with the `MeasureInc` function.

Default Options

You can change the default options for graphs and reports.

Setting Default Chart Options

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Check the **Highlight Selected Series** check box to display the currently selected series with an emphasized, thicker line.

   Note: While this option might be useful as long as you are working with charts, uncheck the check box before you export or print charts.
3. Check the **Overlay series** check box to assume that all time series that are displayed in a chart start at the same time.

   Use this option when you are viewing a chart that contains time series from tests with different starting times.
4. From the **Background colors** lists, select the colors that you want to use to enhance your charts.

   To use a custom color, click **[...]** to the right of the list box.

   Performance Explorer displays a vertical color transition from the first to the second color as the background image.
5. From the **Text and grid color** list box, select the color that you want to use for the axis information and the grid in your charts.

   To use a custom color, click **[...]** to the right of the list box.
6. Use the **Scaling Options** area to determine whether you want to display real time-series data values or to scale values between 0 and 100.

   - Click the **Scale displayed values between 0 ... 100** option button to display scaled values rather than real time-series data values.
   - Click the **Display real values** option button to display real time-series data values rather than scaled ones.
7. Click **OK**.

Setting Default Time-Series Data Options

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Click the **Series** tab.
3. If you want to display null values by default, check the **Display NULL values** check box in the **Default settings** area.

   Use the **Show Null Values** icon on the toolbar to specify whether to display null values separately for each series.
4. From the **Show summary and counter values per** list box, select the time unit for which you want time series data to be displayed.

   - Select **Interval** if you want to display data for the computation interval contained in the TSD file.
   - Select **Second** to scale displayed data to one-second intervals.

   To change the default computation interval, use the **Time Series** tab of the **Profile Settings – Results** dialog box.
5. From the **Default line width** list box, select a line width to use whenever you add a new time series to a chart.
6. To display a chart marker for each measure value of each series, indicating the exact measure value, check the **Show chart markers** check box.
7. In the **Decimal digits** text box, type the number of digits to display in the chart markers.

   Use chart markers to display exact measurement values for each series within a chart.

8. Click **OK**.

Setting Default Options for Exporting Results Graphs

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Click the **HTML View** tab.
3. In the **Chart picture resolution** area, select an option to determine the resolution of the image that is generated when you export a chart as an HTML document.

   The size of an image file is directly proportional to its resolution. As a result, images with higher resolutions take longer to transfer over the Web than images with lower resolutions.
4. Use the **Picture type** area to select the picture format of your chart in an HTML document.
   - Click the **JPEG** option button to include the chart in an HTML document as a JPEG file.
   - Click the **BMP** option button (only with Internet Explorer) to include the chart in an HTML document as a BMP file.
5. Click **OK**.

Setting Default Candle Graph Options

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Click the **Candles** tab.
3. From the **Border color** list box, select the color you want to use for the candle borders in your current graph.

   To use a custom color, click **[...]** to the right of the list box.
4. In the **Candle width** spin box, enter the width for the candles in your current graph.
5. From the **Marker color** list box, select the color you want to use for the markers.

   To use a custom color, click **[...]** to the right of the list box.
6. In the **Marker height** spin box, specify the height for the markers.
7. Click **OK**.

Setting Default Monitoring Options

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Click the **Monitor** tab.
3. In the **Monitor options** area, in the **Interval** boxes set the length of the intervals at which the graph is updated.

   Note: When modifying the monitor interval to a value lower than the default minimum interval during a run, the interval will be reset to the minimum monitor interval upon reopening the saved workspace.
4. In the **Page size** spin box, set the number of measure points currently displayed per graph on the horizontal axis.
5. In the **History size** spin box, set the number of measure points stored for the entire graph on the horizontal axis.
6. From the **Background colors** lists, select the colors that you want to use to enhance your charts.

   To use a custom color, click **[...]** to the right of the list box.

   Performance Explorer displays a vertical color transition from the first to the second color as the background image.

7. From the **Text and grid color** list box, select the color that you want to use for the axis information and the grid in your charts.

   To use a custom color, click **[...]** to the right of the list box.
8. Click **OK**.

Setting Default Time-Series Data Writer Options

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Click the **TSD-Writer** tab.
3. In the **Interval** boxes, select a time-interval value and a time unit for the time-series data that is written.
4. Click **OK**.

Setting Default Reporting Options

1. From the Performance Explorer menu bar, choose **Settings** > **Options** .

   The **Performance Explorer Options** dialog box opens to the **Chart** page.
2. Click the **Reporting** tab.
3. To automatically display a summary overview report when you start exploring the results of a test, check the **Generate overview reports automatically** check box.

   An overview report contains the most important results of a test, such as the number of virtual users that were active, the number of transactions that were executed, and the number of errors that occurred.
4. To display null values in charts, check the **Show NULL values in charts** check box.

   If you uncheck this check box, Performance Explorer replaces each null value with a direct line from the preceding to the following value.
5. To use a previously stored template for the creation of overview reports, check the **Use template when creating a new overview report** check box, click **[...]**, and select the template. The selected template is used to create a new overview report for all projects.
6. Click **OK**.

Customizing Graph Display

Performance Explorer displays graphs and reports in a standard format but offers numerous possibilities for customization. You can change colors, line width, and line styles that are used to display measurements. You can also change scaling options and insert or remove markers.

Hide measurements by right-clicking the graph and unchecking the **Visible** check box.

Note: Hidden measures do not appear in the overview report after export.

Make such changes in the default display settings. All views that are opened after you make these changes feature the new default settings. You can also customize a selected view and instantly view the results.

1. Right-click the graph that you want to customize and choose **Properties**.

   Alternative: Click **View Properties** on the toolbar.

   The **Properties** dialog box opens to the **General** tab.
2. In the **Caption** text box, type or change the name of the current graph.
3. In the **Description** text box, type a description for the current graph.
4. Click the **Chart** tab to edit or modify display options for the current graph.

5. Check the **Highlight Selected Series** check box to display the currently selected series with an emphasized, thicker line.

   Note: While this option might be useful as long as you are working with charts, uncheck the check box before you export or print charts.

6. Check the **Overlay series** check box to assume that all time series that are displayed in a chart start at the same time.

   Use this option when you are viewing a chart that contains time series from tests with different starting times.

7. From the **Background colors** lists, select the colors that you want to use to enhance your charts.

   To use a custom color, click **[...]** to the right of the list box.

   Performance Explorer displays a vertical color transition from the first to the second color as the background image.

8. From the **Text and grid color** list box, select the color that you want to use for the axis information and the grid in your charts.

   To use a custom color, click **[...]** to the right of the list box.

9. Use the **Scaling Options** area to determine whether you want to display real time-series data values or to scale values between 0 and 100.

   - Click the **Scale displayed values between 0 ... 100** option button to display scaled values rather than real time-series data values.
   - Click the **Display real values** option button to display real time-series data values rather than scaled ones.

10. Click **OK** to implement your changes in the graph.

11. Change the appearance of any time series element in your graph by right-clicking it. This opens the **Series Color**, **Line Width**, and **Line Style** list boxes.

12. To expand an area in your graph, drag the left edge of the area you want to expand to the right edge of the area. The area is highlighted as you drag and expands when you release the mouse button.

13. To undo an area expansion, drag in the opposite direction on the graph or click **Zoom out** on the toolbar.

14. Choose the way in which measurement information for the time period in question is displayed from the **Source Type** list box on the toolbar. Performance Explorer calculates and displays the following information:

    - The sum of the values.
    - The average of the values.
    - A count.
    - The minimum or maximum value.
    - The standard deviation from the average.

15. Use the **Series scaling** list box on the toolbar to alter the multiplication factor by which selected elements are scaled.

16. Use the **Series Type** list box on the toolbar to display graph elements as the following objects:

    - A simple line (default).
    - A candle graph, in which each measurement is displayed with a vertical marker containing a central button. The central button represents the average value measured during the time period. The top and bottom of the marker represent the highest and lowest deviation from that average.
    - A candle with sticks graph, in which two further values are displayed as lines above and below the marker. They indicate the maximum and minimum values measured during the period.
    - An area series type, in which the area below the graph line is entirely filled with color.

17. Click **Show Series Markers** on the toolbar to show or hide actual measurement values in the graph.

18. Click **Show Null Values** on the toolbar to show or hide values where the measurement is zero.

19. Click **Toggle Highlight** on the toolbar to highlight selected graph elements with a bold line.
20. Click **Toggle Legends** on the toolbar to display or hide the lower part of the graph window in which the measure type legends appear.
21. Click **Toggle Frames** on the toolbar to show or hide the title bar.
22. Click **Toggle Scaling** on the toolbar to choose between showing real values and showing values scaled between 0 and 100 in the vertical axis.

    Note: Initially a scale that provides a clear display is chosen.
23. Click **Logarithmic Y-axis** on the toolbar to choose between normal and logarithmic scaling of the vertical axis of your graph.
24. Click **Toggle Overlay** on the toolbar to match the starting point of all displayed measures to zero on the horizontal axis.

    This is useful when making comparisons between measurements that start at different times.

Results Analysis

Performance Explorer supports a range of functionality for displaying, reporting, and analyzing load-test results. You can display measurements of your choice in a variety of graphical and tabular formats. Such user-defined test results can contain as many elements as you require. You can even compare results of different load tests within a single graph. Additional functionality supports the reproduction and exploration of past server monitoring.

Assigning Overview Report Templates to Projects

With Silk Performer you can define an overview report template for each load-testing project used with Performance Explorer.

Note: Performance Explorer's command-line interface also offers the `/OVT:<template>` command for assigning overview report templates for use with Performance Explorer.

1. In Performance Explorer, expand the **Profiles** node in the menu tree.
2. Right-click the profile that you want to configure and choose **Edit Profile**.

   Alternative: Choose **Settings** > **Active Profile** .

   The **Profile - [<profile name>]** dialog box opens.
3. In the shortcut list on the left, click the **Results** icon.
4. Click the **Time Series** tab.
5. To define an overview report template for generating overview reports with Performance Explorer for this testing project, click **[...]** on the **Overview report template** portion of the **Time Series** page and perform the following steps:
   a) From the **Select Template File** dialog box, select the overview report template OVT file that you want to have applied to your project.
   b) Click **Open**.

   The file that you specify here is used as an overview report template for the generation of all overview reports for this testing project. The template file name is also preselected on the **Overview Report Template workflow** dialog box when new overview reports are created manually.
6. Click **OK** to save your settings.

Viewing Results

Analysis of load-test results typically consists of four high-level steps:

1. Consulting the **Overview Report** for an initial impression of test results.
2. Examining general measurement types, such as the number of active virtual users, the number of executed transactions, and the number of errors that occurred.
3. Reviewing measurements that are of particular importance for the type of application under test. Performance Explorer provides predefined templates that contain the most relevant measurement types for almost any kind of application.
4. Examining measurements of your choice in more detail. You can choose any number of measurements and combine them in graphs and reports.

Viewing Test Results

Performance Explorer includes a wizard that assists you in selecting load-test results for analysis.

1. Click the **Explore** tab.
2. Perform one of the following steps:

    - Choose **Explore** > **Add Loadtest**.
    - Right-click in the **Results** menu tree and choose **Add Loadtest**.

    The **Project Selection** dialog box opens and lists all the projects available in `<my documents>\Silk Performer10.0\Projects`.
3. Select the project for which you want to view load-test results and click **Next**. The **Load Test Selection** dialog box opens and lists all load tests of the selected project.
4. Select the load test that you want to analyze and click **Finish**. The overview report of the selected load test opens.

The menu tree on Performance Explorer's **Explore** page displays load-test results based on a results structure rather than on files.

The menu tree shows results in the following structure:

- Project_1
    - Loadtest_1
        - Client Measures
            - Health Control
            - Summary General
            - ...
            - Transaction
            - Agents
                - Agent_1
                    - Health Control
                    - Summary General
                    - ...
                    - Transaction
                - ...
                - Agent_n
            - Files
                - File_1 (for example: remerged file 1)
                    - Health Control
                    - Summary General
                    - ...

          - Transaction
        - ...
        - File_n (for example: remerged file n)
    - Server Measures
      - Server_1
      - ...
      - Server_n
  - ...
  - Loadtest_n
- ...
- Project_n

Overview Reports

Overview reports display important load-test result information.

Overview Report Structure

The overview report includes the most important test results in both tabular and graphical form. It consists of the following sections:

- General information
  - General project settings
  - Project attributes
- Summary tables
- Ranking
- User types
- Custom charts
- Custom tables
- Detailed charts

Section	Description
General information	General information includes administrative information in tabular form and important load-test results in graphical form. Administrative information includes the following information: • Project name • Description of the project • Test number • Description of the load test • Date of the load test • Number of agent computers • Number of virtual users running Graphical charts display the following information: • Number of active virtual users • Response-time measurements for transactions • Number of errors that occur over time Transaction response times are provided for the following transactions:

Section	Description
	• Successfully executed transactions • Failed transactions • Cancelled transactions Additional charts display summary measurements related to the type of load-testing project. For example, in the case of Web application testing, response time measurements for Web pages are displayed in a graph.
Summary tables	This section contains a summary of aggregate measurements, which it displays in tabular form for all users. The first table provides general information, such as the number of transactions that were executed and the number of errors that occurred. All following tables provide summary information that is relevant to the type of application under test.
Ranking reports	This section shows a tabular ranking of the 10 worst measures per measure group. For example, in the case of Web-application testing, the slowest, largest, most server resource-consuming, and most network resource-consuming Web pages are displayed. In the case of browser-driven load testing, the slowest action times are displayed. In the case of database application testing, the slowest and most resource-consuming SQL commands are displayed.
User types	For each user group, detailed measurements are provided in tabular form. The following measurements are included: • Transaction response times • Individual timers • Counters • Response time and throughput measurements related to the type of application under test (Web, CORBA, COM, or TUXEDO). In addition, errors and warnings are listed for each user group. Optionally, percentile graphs can be displayed for measures that have been enabled with percentile calculation. Percentile graphs enable you to analyze the percentages of actions that execute within certain time limits. With the help of percentile graphs, you can determine the percentage of transactions, page-load times, and custom timers that meet the performance criteria defined for your system. Additionally, by considering the shape of a graph, you can analyze the distribution of execution times. To view a percentile chart for a particular measure, use the `MeasureCalculatePercentiles()` function.
Custom charts	This section contains graphs that have been added manually. You can add and remove charts from this section at any time. You can save your changes as a template to be used for future summary reports.

Section	Description
Custom tables	This section contains tables that have been added manually. You can add and remove tables from this section at any time. You can save your changes as a template to be used for future summary reports.
Detailed charts	This section provides enlarged versions of charts included in the report. Click a reduced version of a chart to jump to the enlarged version. Click the chart again to return to the reduced version.

Viewing Overview Reports

Performance Explorer automatically provides you with an overview report after load tests are complete.

By default, overview reports are generated automatically. If you have disabled automatic generation of reports, click **Overview Report** on the workflow bar.

Customizing Overview Reports

Performance Explorer enables to you add customized charts and reports to overview reports. You can remove charts and reports from an overview report at any time, and you can save your changes as a template to apply to overview reports that you generate in the future.

Adding a Default Chart to the Overview Report

Before you perform this task, generate and view the overview report.

1. In the Performance Explorer results window, click the overview report to which you want to add the chart.
2. Click **Customize Report** on the workflow bar. The **Custom Overview-Report Wizard** dialog box opens.
3. Check the check boxes for all entries that you want to display in the overview report and click **Next**.

   By default, all available entries are displayed.
4. Check the **Add new view** check box and click **Next**.
5. Select the type of graph you want to insert into the overview report.
6. Click **Finish**. Performance Explorer opens a new view with the type of chart you have selected. In addition, Performance Explorer inserts the chart into the Custom Charts section of the overview report.

   Note: The corresponding graph in the overview report is automatically updated when you add new measurements to or remove existing measurements from the view that contains the graph.

Adding a Custom Chart to the Overview Report

Before you begin, generate the overview report and the graph that you want to add to it.

1. In the Performance Explorer results window, click the overview report to which you want to add the chart.
2. Click **Customize Report** on the workflow bar. The **Custom Overview-Report Wizard** dialog box opens.
3. Check the check boxes for all entries that you want to display in the overview report and click **Next**.

   By default, all available entries are displayed.
4. Check the check box for the chart that you have just created.
5. Check the **Add new view** check box and click **Next**.
6. Select the template type that you want to use for the chart and click **Finish**. Performance Explorer inserts the chart you have selected into the Custom Charts section of the overview report.

Note: The corresponding graph in the overview report is automatically updated when you add new measurements to or remove existing measurements from the view that contains the graph.

Adding a Default Report to an Overview Report

1. In the Performance Explorer results window, click the overview report to which you want to add the report.
2. Click **Customize Report** on the workflow bar. The **Custom Overview-Report Wizard** dialog box opens.
3. Check the check boxes for all entries that you want to display in the overview report and click **Next**.

   By default, all available entries are displayed.
4. Check the **Add new view** check box and click **Next**.
5. Click the report template that you want to use and then click **Finish**. Performance Explorer opens a new view with the type of report you have selected. In addition, Performance Explorer inserts this report into the Custom Tables section of the overview report.

   Note: The corresponding report in the overview report is automatically updated when you add new measurements to the view that contains the report, or when you remove measurements from the view.

Adding a Custom Report to the Overview Report

Before you begin, generate the overview report and the report that you want to add to the overview report.

1. In the Performance Explorer results window, click the overview report to which you want to add the report.
2. Click **Customize Report** on the workflow bar. The **Custom Overview-Report Wizard** dialog box opens.
3. Check the check boxes for all entries that you want to display in the overview report and click **Next**.

   By default, all available entries are displayed.
4. Check the check box for the report that you created.
5. Check the **Add new view** check box and click **Next**.
6. Click the report template that you want to use and then click **Finish**. Performance Explorer opens a new view with the type of report you have selected. In addition, Performance Explorer inserts this report into the Custom Tables section of the overview report.

   Note: The corresponding report in the overview report is automatically updated when you add new measurements to the view that contains the report, or when you remove measurements from the view.

Removing a Chart from the Overview Report

1. In the Performance Explorer results window, click the overview report from which you want to remove the chart.
2. Click **Customize Report** on the workflow bar. The **Custom Overview-Report Wizard** dialog box opens.
3. Click **Next**.
4. Uncheck the check box for the chart that you want to remove from the overview report and then click **Finish**. Performance Explorer removes the selected report from the Custom Charts section of the overview report.

Removing a Report from the Overview Report

1. In the Performance Explorer results window, click the overview report from which you want to remove the report.

2. Click **Customize Report** on the workflow bar. The **Custom Overview-Report Wizard** dialog box opens.
3. Click **Next**.
4. Uncheck the check box for the report that you want to remove from the overview report and then click **Finish**. Performance Explorer removes the selected report from the Custom Tables section of the overview report.

Changing Text Areas

Within Performance Explorer overview reports, many text areas are predefined. You can change them according to your needs.

1. In an overview report, click **Click here to edit text** to change the corresponding text.
2. Save the report as a template for future reports.

Saving A Template

Save a template to use it for future overview reports.

1. Customize an overview report according to your needs.
2. Choose **File** > **Save As Template**.
3. In the **File name** text box, type a name for the new template.

   The template is stored at `<my documents>\Silk Performer10.0\Projects` for use by all projects.
4. In the **Save as type** text box, ensure that **Overview Template (.ovt)** is selected and then click **Save**.

Setting a Default Template

Specify the template to use to create all new overview reports for all projects.

1. Choose **Settings** > **Options** to specify that this template be used when creating future overview reports. The **Options** dialog box opens.
2. Click the **Reporting** tab.
3. Check the **Use template when creating a new overview report** check box.
4. In the **Template** text box, type the name and location of the template to use.

   Alternative: Click **[...]** to select the template from the **Select Template File** dialog box.
5. Click **OK**.

The selected template is used to create all new overview reports for all projects.

Note: This setting is overridden when a default overview template is specified by using the Performance Explorer `/OVT:<template>` command-line option or when a template is specified in the corresponding Silk Performer project.

Changing the Name or Description of a Chart or Report

Change the name or description of a chart or report to better describe the shown information.

1. Right-click the graph or report and choose **Properties**. The **Properties** dialog box opens.
2. In the **Caption** text box, type a caption.
3. In the **Description** text box, type a description.
4. Click **OK**.

Specifying the Time-Series Data (TSD) Results File

Specify the time-series data (TSD) file to use before attempting to complete the following tasks:

- View specific results
- Generate a custom report or graph
- Combine results from multiple files

1. Choose **Explore** > **Add Results** .

   Alternative: Right-click the project name in the menu tree and choose **Add Results**.

   The **Select Result File** dialog box opens.
2. Locate the TSD file for which you want to view a graph and click **Open**. Performance Explorer displays a menu tree with all the measurements that the selected file contains.
3. If Silk Performer prompts you, specify whether you want to load all relevant files of the load test.

Specific Results

After you have examined the general measurements that were recorded during a load test, you can look at more particular measurements, which are typical for the type of server or application under test. For example, when you load test a Web application, you examine the number of successful connections to the server and the number of HTTP hits and concurrent connections. For a database application, the number of database logon attempts, active database cursors, and executed SQL statements are of particular interest.

Viewing Specific Results in a Graph

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the graph.

1. Click **Select Graph** on the workflow bar. The **Template** dialog box opens.
2. In the **Templates** area, click the template that corresponds with the measures that you want to view.

   To view the default graph, click **General**.
3. Click **Finish**. A graph displaying the measure you selected opens.

Viewing Specific Results in a Report

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the report.

1. Click **Select Report** on the workflow bar. The **Template** dialog box opens.
2. In the **Templates** area, click the template that corresponds with the measures that you want to view.

   To view the default report, select **General**.
3. Click **Finish**. A report displaying information about the measure you selected opens.

Custom Results

To view specific details of the performance of your application, create custom graphs and reports in which you indicate the measurements that are displayed in a graph or report.

Generating a Custom Graph

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the graph.

1. Choose **File** > **New** > **Graph** . A new graph opens.
2. In the menu tree, expand the folder containing the measure types you want to add to the graph.
3. Drag a measure you want into the graph.
4. Repeat this process as necessary to add more elements to the graph.
5. *Optional:* If you have defined measure bounds in the **Automatic Threshold Generation** dialog box or directly in the script, you can display these measure bounds in the graph. Right-click a measure in the monitor list and click **Display Measure Bounds**.

Generating a Custom Report

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the report.

1. Choose **File** > **New** > **Report** . Performance Explorer displays a menu tree with all the measurements that the selected file contains.
2. In the menu tree, expand the folder containing the measure types you want to add to the report.
3. Drag the measure you want to the report.
4. Repeat the procedure as necessary to add more elements to the report.

Displaying Measure Bounds

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the graph.

Before you start a load test, you must define a lower and upper bound within the **Automatic Threshold Generation** dialog box or directly within the script. The values of the bounds are displayed in the Overview Report.

To make the measure bounds visible within the graph:

1. Choose **File** > **New** > **Graph** . A new graph opens.
2. In the menu tree, expand the folder containing the measure types you want to add to the graph.
3. Drag a measure into the graph.

   Note: See below regarding the measures for which graphical bounds can be displayed.

4. Right-click the measure in the monitor list and select **Display Measure Bounds**. The graph shows a red and a yellow colored area. The red colored area is the area above the upper bound. The yellow colored area is the area between the upper bound and the lower bound.

   Note: The colored areas are only displayed if the measure exceeds the configured thresholds. Performance Explorer only shows graphical measure bounds for one measure at a time. When you display measure bounds, the scale of the graph may adjust automatically. The column **Bounds** in the monitor list shows the measures for which graphical bounds can be displayed.

Currently, Performance Explorer shows graphical bounds for the **Average**, **Maximum**, and **Minimum** values of the following measures:

- Page and Action Timer - Action Time
- Page and Action Timer - Page Time
- Timer - Response Time[s]
- Transaction - Trans. (busy) Ok

Viewing Results from Multiple Load Tests

Performance Explorer can combine the results of multiple load tests into a single graph or report, allowing you to compare two or more load tests. For example, you might want to view results from an earlier test that took place before you fine-tuned the performance of your server with results from a more recent test.

Combining Results from Multiple Load Tests in a Graph

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the graph.

1. Choose **File** > **New** > **Graph** . A new graph opens.
2. Choose **Explore** > **Add Results** .

   Alternative: Right-click the project name in the menu tree and choose **Add Results**.

   The **Select Result File** dialog box opens.
3. Select the recorded load tests from which you want to include data in the custom graph and then click **Open**.

   Press `Shift` and `Ctrl` to select multiple files from the list.

   The selected TSD files are added to the available resources in the menu tree.
4. In the menu tree, expand the folders that contain the load test and measurement types you want to include in your graph.
5. Drag the required measurements to the graph.

Combining Results from Multiple Load Tests in a Report

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the report.

1. Choose **File** > **New** > **Report** . A new report opens.
2. Choose **Explore** > **Add Results** .

   Alternative: Right-click the project name in the menu tree and choose **Add Results**.

   The **Select Result File** dialog box opens.
3. Select the recorded load tests from which you want to include data in the custom report and then click **Open**.

   Press `Shift` and `Ctrl` to select multiple files from the list.

   The selected TSD files are added to the available resources in the menu tree.
4. In the menu tree, expand the folders that contain the load test and measurement types you want to include in the report.
5. Drag the required measurements to the report.

Region Comparison Reports

Region comparison reports compare cloud-based load test measures across geographic regions, with results from different regions compared side by side, one region per column.

Comparing Total Results and Region-Specific Results

A number of options are available for configuring region comparison reports. Total test results, which contain measure results from all regions, can be compared side by side with region-specific measure results. Alternatively, region comparison reports can be configured to include data only from select combinations of individual region results.

Additional regional results can be added to an existing report (by dragging and dropping individual region nodes from the **Explore** tab into the report) and total results can be added to an existing report (by dragging and dropping the **Client Measures** node into the report).

Baseline Pane

The **Baseline Pane** (at the bottom of each region comparison report) includes a **Region** column, which contains either the name of a specific region or `Total` if the row contains all test results. The **Agents** column displays the number of agents that were used for the test in each specific region.

Any region can be defined as the baseline to which other region results are measured.

Generating Region Comparison Reports

Before completing this task you must execute a cloud-based load test in Silk Performer that utilizes agents from multiple geographic regions.

1. On the Silk Performer **Results** tab, expand a load test results time series data file (for example, <test_results>.ltz).
2. Double-click the **Time Series** node to launch Performance Explorer.

   The time series data is then loaded into Performance Explorer and available on the **Explore** tab.

   Merged measures/time series data are available for each cloud region included in the load test (for example, `EU-Ireland` and `US-West`) in the same way that measures/time series data are provided for project agents. A `Project Agents` node is available that includes results from any project agents that were included in the test (those agents that are listed beneath the `Agents` node in the Silk Performer **Project** tree), enabling you to evaluate project agent results against specific region-based results.
3. Expand the test results node and the **Client Measures** node.
4. Right-click the **Regions** node and select **Region Comparison Report**.

   Alternative: Click the **New Region Comparison Report** button on the toolbar.

   A region comparison report is generated, comparing the measures of all regions side by side, one column per region, in a format similar to a cross load-test report. A maximum of six regions can be included in a single region comparison report.
5. *Optional:* You can alternatively generate a report for only a single region by right-clicking a region node and choosing **Region Comparison Report**. You can subsequently drag and drop additional region nodes from the **Explore** tab into the report to dynamically merge region results and compare them side by side.

   Only regions from the same load test can be integrated into a report.
6. *Optional:* To define one region's results as the baseline for a report and have the other regions' results compared to the baseline region, right-click the region entry (in the baseline pane at the bottom of the report) and choose **Set as Baseline**.

   In report rankings the baseline region is always listed as the first region. A **b** icon or **baseline** tag identifies the baseline region. All measures that are evaluated against the baseline are identified with heat fields. Pass your cursor over any heat field to view the details of the heat-field setting.

Viewing Load-Test Results and Server-Monitoring Data

Performance Explorer can combine server-monitoring data and load-test results into a single graph or report. To accomplish this task, write the monitoring data to a TSD file while the server is load tested. After the load test is complete, save the load-test results and import the TSD file containing the monitoring data into a graph or report containing the load-test reports.

Viewing Load-Test Results and Server-Monitoring Data in a Single Graph

1. Choose **File** > **New** > **Graph** . A new graph opens.
2. Choose **Explore** > **Add Results** .

   Alternative: Right-click the project name in the menu tree and choose **Add Results**.

The **Select Result File** dialog box opens.

3. Select the recorded load tests from which you want to include data in the custom graph and then click **Open**.

   Press `Shift` and `Ctrl` to select multiple files from the list.

   The selected TSD files are added to the available resources in the menu tree.
4. In the menu tree, expand the folders that contain the load test and measurement types you want to include in your graph.
5. Drag the required measurements to the graph.

Viewing Load-Test Results and Server-Monitoring Data in a Report

1. Choose **File** > **New** > **Report** . A new report opens.
2. Choose **Explore** > **Add Results** .

   Alternative: Right-click the project name in the menu tree and choose **Add Results**.

   The **Select Result File** dialog box opens.
3. Select the recorded load tests from which you want to include data in the custom report and then click **Open**.

   Press `Shift` and `Ctrl` to select multiple files from the list.

   The selected TSD files are added to the available resources in the menu tree.
4. In the menu tree, expand the folders that contain the load test and measurement types you want to include in the report.
5. Drag the required measurements to the graph.
6. Drag the required measurements to the report.

Viewing Results in a Web Browser

Performance Explorer can convert test results, graphs, and reports into HTML documents. This feature enables you to include graphs and reports in any kind of documents, publish them online, or print them in useful formats.

Viewing a Results Graph in a Web Browser

Create or open the graph that you want to view in a Web browser.

1. Click **View as HTML** on the workflow bar. The **View as HTML** dialog box opens.
2. In the **Caption** text box, type a caption.
3. In the **Description** text box, type a description.
4. Click **OK**. The Web browser opens and displays the graph, which is ready for publication on the Web.

Viewing a Results Report in a Web Browser

Create or open the report that you want to view in a Web browser.

1. Click **View as HTML** on the workflow bar. The **View as HTML** dialog box opens.
2. In the **Caption** text box, type a caption.
3. In the **Description** text box, type a description.
4. Click **OK**. The Web browser opens and displays the report, which is ready for publication on the Web.

Exporting HTML Reports

Visually communicate project status and error conditions to others by using Performance Explorer graphs. Since others might not have access to Performance Explorer, Performance Explorer allows you to export all HTML reports as Web archives (MHT files), which you can easily send in an email.

1. With the chart you want to archive loaded into Performance Explorer, perform one of the following steps:
   - Choose **File** > **Export View**.
   - Click the **Export View** icon on the toolbar.

   The **Export As** dialog box opens.
2. Browse to the directory where you want to save the archive.
3. In the **File Name** text box, type a name for the archive.
4. Select **Web Archive (.mht)** from the **Save as type** list box.
5. Click **Save** to generate the Web archive report.

Exporting Reports Through the Command-Line Interface

HTML reports can also be exported as Web archives (MHT files) through the command-line interface. To export reports, use the command-line parameter `/EXPORTOVR:<target file>`.

This parameter saves the active OVR as HTML or as an MHT file, depending on the file extension of the target file specification.

Note: The last loaded TSD file, or the last TSD file specified through the command line (`/TSD:<tsd file>`) is used for exporting.

The command-line parameter `/ACTION:OVERVIEWREPORT[:<template>]` allows you to specify a template file for the report.

The following example illustrates exporting a Web archive file through the command line:

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

Customizing Time-Series Settings

You can customize TSD and real-time series to suit your needs.

1. Choose **Settings** > **Options**. The Silk Performance Explorer **Options** dialog box opens.
2. Click the **Series** tab.
3. *Optional:* Uncheck the **Display Null values** check box to hide null values by default.

   Tip: Click **Show Null Values** on the toolbar to specify whether to display null values separately for each series.
4. Select the time unit in which the time-series data is displayed from the **Show summary and counter values per** list box, as follows:
   - Select `interval` to display data for the default computational interval (every five seconds).
   - Select `second` to display data for one-second intervals.

   Tip: To change the default computational interval, use the **Interval** text box on the **Monitor** tab. Click **Normalize Series** on the toolbar to display one-second-interval normalized data separately for each series.

   Note: Normalizing time series is allowed only for data of counter or summary source types.
5. From the **Default line width** list box, select the line width to use whenever you add a new time series to a chart.
6. Check the **Show chart markers** check box in the **Markers** area to display a chart marker for each measure value of each series, indicating the exact measure value.
7. In the **Decimal digits** text box, specify the number of digits to display in the chart markers.
8. Click **OK**.

Root Cause Analysis

Before you perform this task, specify the time-series data (TSD) file that is to be used as the data source for the graph.

Performance Explorer's automatic result-correlation feature facilitates root-cause analysis of network and server bottlenecks by correlating client-side issues with corresponding server-side measurements. Automatic result correlation identifies the server-side measurements that are most closely associated with specific client-side errors, thereby enabling you to better identify server-side problems and expedite debugging efforts. Result correlation also works in the reverse: Server-side issues can be correlated with client-side measures.

Automatic result correlation statistically correlates key measures with dependent measures. For example, if a significant increase in server response time is detected by a client-side measure at 18:20 (6:20 p.m.), automatic result correlation can identify the server-side measures that contributed to this drop in client-side performance.

1. Click within the measurement graph that you want to analyze and drag your cursor to the right to select the time frame to analyze.

   Note: To slide the time line forward or backward in time, right-click the time line and drag your mouse right or left. The time line can also be moved vertically along its Y axis.

   Note: To select a shorter period of time for analysis, drag the time line to the right. To select a longer period of time for analysis, drag the time line to the left.
2. Click **Find Root Cause** on the workflow bar.

   Alternative: Right-click in the graph and choose **Root Cause Analysis**.

   The **Find Root Cause - Correlation Settings** dialog box opens. The base measure appears in the **Base Measure** box.
3. From the **Correlate with** list box, select the type of measurement you want to correlate with the base measurement.
4. Adjust the date- and time-selection settings by selecting a new start date, start time, end date, or end time from the appropriate list box.

   Date and time selection settings are defined automatically based on the time frame selected in the time line.
5. In the **Results** area, specify how to filter measure results based on how well they match.

   Choose one of the following options:

   - Click the **Best [x] correlation numbers** option button and specify the number of measures to be returned in the text box.
   - Click the **Minimum correlation of [x]%** option button and specify a minimum correlation relevance by typing a value in the text box.
6. Click **Next**.

   Note: If you are correlating against client and server measures, client measures, or server measures, advance to step 10.

   The **Correlation Measures Properties** page opens.
7. When correlating against a custom measure, you must define which measure groups are to be correlated. Select the groups you want to correlate against by checking the **Measures** check boxes.
8. *Optional:* Add other measures by clicking **Add File** and browsing to and selecting a time-series data file.

   You can also remove measures by selecting them and clicking **Remove File**.
9. Click **Next** to run the correlation. The **Correlation Results** page opens.

By default, returned measures are listed in order of degree of correlation, and all correlations are selected.

10. Click the **Correlation Group** and **Name** column headers to sort returned measures.
11. Uncheck the check boxes for the measures that you do not want to include in the correlation graph.
12. Click **Finish** to generate the correlation graph.

Merging Files

The merge wizards are available from the **Wizard** menu. The **Merge** and **Custom Merge** wizards are similar in that they merge multiple time-series data (TSD) files into a single TSD file.

The **Remerge** wizard creates a new TSD file that features a different interval than the original file.

Merging TSD Files from a Project into a Single Overview Report

The **Merge** wizard merges several time-series data (TSD) files from a load test into a single TSD file that contains all measurements included in the source files.

1. Choose **Wizard** > **Merge Wizard** . The **Project Selection** dialog box opens.
2. Select the project that contains the load test for which you want to merge time series data and click **Next**. The **Load Test Selection** dialog box opens.
3. Select the load test that you want to merge into the single destination file and click **Next**. The **File Selection** dialog box opens.
4. Select the time series data files that you want to merge and click **Next**.The **Advanced Merging Properties** dialog box opens.
5. In the **Output interval** spin box, specify how many seconds pass between each interval for merging time-series data.

   For longer running load tests, an increased output interval is useful because it results in a smaller target file.
6. Choose one of the following merge types:

   - Click the **Relative** option button to specify that all series start at the same time.
   - Click the **Absolute** option button to consider the individual start time of each series.

     If you are merging files from different load tests, the start times of those files are different, so use the **Relative** merge type.

     If you choose **Absolute**, you can specify a time frame. Only data within this time frame is merged into the target file.
7. Adjust the date- and time-selection settings by selecting a new start date, start time, end date, or end time from the appropriate list box.

   Date and time selection settings are defined automatically based on the time frame selected in the time line.
8. Click **Next** or **Finish** to start the merging process.

After merging, an overview report is generated and displayed.

Merging TSD Files from a Directory into a Single Overview Report

The **Custom Merge** wizard merges several time-series data (TSD) files from a load test into a single TSD file that contains all measurements included in the source files.

1. Choose **Wizard** > **Custom Merge Wizard** . The **Directory Selection** dialog box opens.
2. In the **Source directory** text box, specify the directory where the time series data files to merge are located.*Alternative:* Click **[...]** to open the **Browse for Folder** dialog box and select the location.

3. In the **Target file** text box, type the name of the file in which to include the merged TSD file.
4. Click **Next**.The **File Selection** dialog box opens.
5. Choose a load-test directory and a target file name for the merged TSD file.

   The file name has the format `m@@<TARGETFILENAME>.tsd`.
6. Select the TSD files from the load-test directory that you want to merge into your single destination file and click **Next**. The **Advanced Merging Properties** dialog box opens.
7. In the **Output interval** spin box, specify how many seconds pass between each interval for merging time-series data.

   For longer running load tests, an increased output interval is useful because it results in a smaller target file.
8. Choose one of the following merge types:

   - Click the **Relative** option button to specify that all series start at the same time.
   - Click the **Absolute** option button to consider the individual start time of each series.

     If you are merging files from different load tests, the start times of those files are different, so use the **Relative** merge type.

     If you choose **Absolute**, you can specify a time frame. Only data within this time frame is merged into the target file.
9. Adjust the date- and time-selection settings by selecting a new start date, start time, end date, or end time from the appropriate list box.

   Date and time selection settings are defined automatically based on the time frame selected in the time line.
10. Click **Next** or **Finish** to start the merging process.

After merging, an overview report is generated and displayed.

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.

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. In the **Output interval** spin box, specify how many seconds pass between each interval for merging time-series data.

   For longer running load tests, an increased output interval is useful because it results in a smaller target file.
4. Choose one of the following merge types:

   - Click the **Relative** option button to specify that all series start at the same time.
   - Click the **Absolute** option button to consider the individual start time of each series.

     If you are merging files from different load tests, the start times of those files are different, so use the **Relative** merge type.

     If you choose **Absolute**, you can specify a time frame. Only data within this time frame is merged into the target file.
5. Adjust the date- and time-selection settings by selecting a new start date, start time, end date, or end time from the appropriate list box.

   Date and time selection settings are defined automatically based on the time frame selected in the time line.

6. Click **Next** or **Finish** to start the merging process.

After merging, an overview report is generated and displayed.

Types of Time-Series Files

Time Series File Type	Description
Merged Files	Merged files begin with `m@...` and contain the time-series data from all agents. They are generated by the load-test controller.
Intermediate Files	Intermediate files begin with `I@...` and are generated by the load-test agents containing all time-series data from one agent including data from all virtual users running on that agent.
Agent Files	Agent files begin with `a@...` and contain the time-series data from one agent. During a load test, these agent files are generated on the agent and are then merged together with all user files into an intermediate file.
User Files	User files begin with `u@...` and contain the time-series data from one virtual user. During a load test, these user files are generated on the agents and are then merged together with all other user files into an intermediate file.
Real-Time Files	Real time files begin with `r@...` and contain the time-series data generated by server-side monitoring.
Agent Files per User Type	Agent files per user type begin with `t@...` and contain the time-series data of a certain user type from one agent.
Merged Files per User Type	Merged files per user type begin with `k@...` and contain the time-series data for a certain user type from all agents. They are generated by the load-test controller.
Region Files	Region files begin with `n@...` and contain the time-series data for a certain region.

Note:

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

Merging Files When the Controller Loses Contact with Agents

If the controller-agent connection fails, agent files reside on the agent in the local results directory.

Depending on when the connection is lost, you might find either user-only or merged agent files.

If you want these results included in the intermediate file of your load-test controller, you must manually merge the files together.

1. Copy all files to the directory where the intermediate file is stored.
2. Choose **Wizard** > **Custom Merge Wizard** . The **Custom Merge Wizard** opens.
3. Select the directory where all the files are stored and select all the files to merge.
4. Click **Finish**.

A new, merged file containing all the data is generated.

CSV File Export

Export results in CSV format to analyze them offline in spreadsheet programs.

Exporting Results in CSV Format

Using Performance Explorer's **Export** wizard, time-series data (TSD) files can easily be exported into comma-separated values (CSV) files for offline analysis in spreadsheet programs such as MS Excel.

Note: In Silk Performer, the functionality offered by the **Export** wizard can be automated through the TSD File Export command-line tool.

1. Choose **Wizard** > **Export Wizard** . The **File Selection** page opens.
2. Click **[...]** to the right of the **Source file** text box to select the TSD file that you want to export.

   By default, the **Export Wizard** uses the name of the TSD file for the name of the CSV file.
3. Click **[...]** to the right of the **Export file** text box to specify a name and export location for the CSV file that is to be created.

   By default, the **Export** wizard uses the location of the TSD file for the location of the CSV file.
4. Click **Next**.The **Time Series Selection** page opens, enabling you to specify the measures for the export.
5. Select the measures that you want to export to the output file.

   At least one measure must be selected. Use **Select All** and **Clear All** to select all measures or to clear all measure selections.
6. Click **Next**. The **Options Selection** page opens.
7. In the **Export type** area, perform one of the following steps:
   - Click the **Standard** option button and specify the data source elements of the result file that must be included in the CSV file.
   - Click the **Dump** option button to include all raw data in the CSV file.
8. In the **Data source** area, check the appropriate check boxes for the element types that you want to export.
9. In the **Export options** area, check the **Include header information** check box to include header information in the CSV file.
10. Define how decimal points are to be signified by selecting an option from the **Decimal point** list box.
11. Click **Finish** to generate the CSV file based on your specifications.
12. To view the output file, click **Yes** on the **Export completed** dialog box. Otherwise, click **No** to close the **Export** wizard without opening the output file.
    For example, to launch MS Excel, click **Yes**.

Using the Command-Line Tool to Automate CSV File Export

The Time-Series Data (TSD) File Export Tool is a command-line tool that enables you to automate the export of TSD files to CSV formatted files. The command-line tool offers the same functionality offered by the **Export** wizard without the ease of a GUI.

1. Choose **Start** > **Programs** > **Silk** > **Silk Performer <version>** > **Analysis Tools** > **TSD File Export Tool**.
2. At the command prompt, enter specifications for the file export based on the following syntax.

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

Allowed Commands	Description
`tsdFiles`	Enter the name of the exported TSD file.
`[csvFile]`	Enter the name of the created CSV file.
`[-Delimiter char]`	Define the delimiter character.
`[-DecimalPoint char]`	Define the character that signifies decimal points in the output file.
`[-dump]`	Ignores `-DecimalPoint` parameter.

3. Press `Return` or `Enter` to generate the output file.

Example

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

Comparing Graphs

After you have identified load-test measurements of interest, you might want to examine the same measurement types from a different load test. Performance Explorer includes a feature that facilitates a file comparison. When you select a time-series data (TSD) file that you want to compare with a selected graph, Performance Explorer creates a new graph by using only the measurement types that are displayed in the current graph.

1. Open the graph that contains the measurement you want to compare and click **Compare Graph** on the workflow bar. The **Compare TSD-File** dialog box opens.
2. Locate the TSD file from the recorded load test that you want to compare with the active results and then click **Open**. A new graph opens, populated with as many identical elements as are available.
3. *Optional:* Resize and relocate the graphs to facilitate comparison.

Error Analysis

Display all errors in a graphical format and drill down to find detailed information.

Analyzing Errors

Performance Explorer includes a utility that assists in the analysis of errors that occur during load tests. While the overview report displays some information on errors, the Analyze Errors feature displays all errors in a graphical format that lets you drill down to find detailed information.

1. After a Silk Performer load test finishes, perform one of the following steps:
    - Click **Analyze Errors** on the workflow bar.
    - Choose **Wizard** > **Analyze Errors**.
    - Click the **Explore** tab, right-click the results pane and choose **Analyze Errors**.

   The **Error Analysis Source** dialog box opens.
2. In the **File name** text box, type the TSD file name for the load-test results you want to examine.

   Alternative: Click **[...]** to browse for the TSD file.
3. Ensure that **Merged Time Series File (m@*.tsd)** is specified in the **Files of type** text box.
4. Click **Open**.
5. Click **Finish**. An error graph opens in which you can use the standard zooming features of Performance Explorer to drill down to the specific information that you need.

Error Analysis Graph

The Error Analysis graph contains the following information:

View	Description
Error Groups	The **Error Groups** pane displays group information for all the errors that occurred during the analyzed load test. Use the zoom and shift features to view information for specific time frames.
Error Details	The **Error Details** pane list all occurrences of errors in the selected time span. Double-click an error or right-click an error and select **Show TrueLog** to invoke TrueLog Explorer with the respective TrueLog file that contains visualized information for the error occurrence to help you find the root cause of the selected error.
Legend	The **Legend** pane lists the displayed curves of the chart, which by default is only the error summary of the error analysis. It also displays more detailed information about occurrences, minimum and maximum values, and so on. Hold your cursor over the graph to see the number of error occurrences for a specific time frame. The **Value** column updates as you move your cursor over the graph.

Silk Central Integration with Performance Explorer

Performance Explorer integration with Silk Central enables you to use Performance Explorer for in-depth analysis of Silk Performer test runs that are invoked via Silk Central. Start Performance Explorer directly from Silk Central, and result packages are transferred easily between Silk Central and Performance Explorer. Upload the results of the most recent Silk Performer load-test runs, excluding recent Try Script runs, to Silk Central and associate the results with Silk Central test definitions.

Performance Explorer offers an **Add Silk Central results** icon on the toolbar to assist you in fetching relevant Silk Central test results, and a **Silk Central** tab beneath the results pane to help you integrate and organize test results.

Adding Results from Silk Central

1. In Performance Explorer, click the **Silk Central** tab, then click the **Click here to Add Silk Central results** link.

   Alternatively: Click **Add Silk Central results** on the toolbar.

   The **Open Results from** Silk Central dialog box opens.
2. In the **Hostname** text box, specify the Silk Central host to which you want to connect.
3. In the **Port** text box, specify the port to use to connect to Silk Central.
4. In the **Username** text box, specify a user who has access to Silk Central.
5. In the **Password** text box, specify the appropriate password for the username.
6. Click **Next**.

   The settings are supplied automatically if you have already configured access to Silk Central from Silk Performer.
7. From the Silk Central project list, select the project from which you want to open Silk Performer results.
8. Click **Next**.
9. From the test definition menu tree, select the test definition from which you want to open Silk Performer results.

10. Click **Next**. The **Select execution** page opens.
11. Select the execution of the test definition for which you want to view results.

    Using the **Test executions from last [ ] day(s)** box, you can specify the number of days into the past for which you want to include definitions in the **Executions** list.
12. Click **Finish** to load the selected execution results into Performance Explorer. Performance Explorer's **Silk Central** page displays the path to the results file in the Silk Central GUI and an overview report is automatically generated.

Downloads Available on Demand

The **Silk Central** page offers file downloads on demand. Measure-result files remain in the Silk Central database until you explicitly request that they be opened in Performance Explorer for viewing.

1. Browse through the nodes of the file menu tree on the Performance Explorer **Silk Central** tab.
2. Double-click a measure to open it for viewing within Performance Explorer.

Cross Load-Test Reports

Cross load-test reports enable you to compare the results of up to four test executions side-by-side. Cross load-test reports are structured like standard baseline reports in that they offer summary reports and statistics regarding transactions, page and action timers, Web forms, rankings, and errors.

Generating a Cross Load-Test Report

Generate a cross load-test report to compare the results of up to four test executions side-by-side.

For rankings, the baseline test is always listed as the first test.

1. Within Performance Explorer, click the **New Cross Loadtest Report** button on the toolbar. The **Cross Loadtest Report** window opens.
2. Open a maximum of four tests.

   Each new report is assigned a title with a unique number from one to four.

   Each new load-test execution you open appears on the **Silk Central** page alongside the other executions you have imported.
3. From the **Explore** tab, drag the executions that you want to include in the report into the **Cross Loadtest Report** window.

   Alternative: Right-click a test definition in the file menu tree on the Performance Explorer **Silk Central** tab and choose **Cross Loadtest Report**. All executions of the selected test definition will then be automatically added to a cross load-test report.

A **b** icon or **baseline** tag identifies the baseline execution in each report. All measures that are evaluated against the baseline are identified with heat fields.

Note: Pass your cursor over any heat field to view the details of the heat-field setting.

Note: To define a different execution run as the baseline to which the other tests should be compared, right-click the execution and choose **Set as Baseline**.

Command-Line Parameters

Performance Explorer offers a number of parameters that you can execute from a command line-interface.

Starting Performance Explorer from the Command Line

Start Performance Explorer from a command-line interface by using the following syntax.

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

For example, you might type the following command.

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

Or, you might type the following command.

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

Available Commands

The following table lists the commands that you can use in a command-line interface.

Command	Description
/TSD:*<file>*	Adds the file to the workspace.
/OVT:*<template>*	Set *<template>* as preselection of the overview report template when manually generating an overview report.
/MONITORDIR:*<dir>*	Sets the output directory for monitor files.
/MONCREATE:*<dir>*	Starts in Monitoring mode (BDL Script Generation).
/ACTION:STDGRAPH	Uses the last given /TSD:<file>.
/ACTION:SELGRAPH	Uses last given /TSD:<file>.
/ACTION:MONSERVER	Launches the server-monitoring wizard.
/ACTION:STARTALLMONITORS	Starts all monitor graphs.
/ACTION:STOPALLMONITORS	Stops all monitor graphs.
/ACTION:OVERVIEWREPORT [:*<template>*]	Generates an overview report for the last given /TSD:<file>. Can optionally apply a template file.
/EXPORTOVR:*<file>*	Exports a recently generated OVR view to <file>. Use `file.htm` for HTML or `file.mht` for WebArchive export.
/NOGUI	Specifies whether Performance Explorer is executed visibly or invisibly (GUI-less mode). `/NOGUI` functions appropriately only when `/NOEXIT` is used in a DOS-based command line.
/SINGLEINSTANCE[:NOSTARTUP]	Searches for an existing instance. When used with `NOSTARTUP`, no new instances are created.
/PEW:*<WorkspaceFile>*	Opens or creates the given workspace. When used with `/SINGLEINSTANCE`, an existing instance is selected only when it uses the given workspace file. Otherwise, a new instance is created.
/COMMANDFILE:*<CmdFile>*	Load commands from a given XML file.

Command	Description
/EXIT	Saves the TSD file, then closes Performance Explorer without saving changes to the workspace.
/NOEXIT	Performance Explorer stays open after the command-line execution.
/?	Displays Performance Explorer with allowed command-line parameters.

Command File

Besides submitting commands directly within the command line, you can also submit a command file in XML format. Specify the commands by using the following syntax.

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

The command file is designated for batch operation.

The structure of the XML file to use as a command file consists of a root node called PerfExpCommandFile and two optional sub-nodes called OverviewReport and Command.

PerfExpCommandFile

Possible specifications in the root node PerfExpCommandFile include the following examples:

- `NoGui="True" / NoGui="False"`

  Specifies whether Performance Explorer launches visibly or invisibly.
- `LogFile="<filename>"`

  Specifies the name and location of the log file.

Different Command and OverviewReport nodes can reside below the root node.

OverviewReport

The OverviewReport node specifies which TSD file Performance Explorer uses. The following attributes are available:

- `File`

  Specifies which TSD file to use. If omitted, the last file loaded by Performance Explorer is used.
- `Export`

  Specifies the output file to which the overview report is saved. Available formats include the following options:

  - HTML (`.htm`)
  - WebArchive (`.mht`)
- `Template`

  Specifies an optional template OVT file to use for the overview report.

Command

The text of a Command node is interpreted as a command-line parameter. Nodes are executed sequentially.

The following code shows an XML command file:

```
<?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>
```

Help command

Use the help command `/?` with the following syntax.

```
Perfexp.exe /?
```

Performance Explorer opens and displays the command-line help.

Index