MICRO FOCUS

Silk Central 20.0

Silk Central / Silk Performer Integration Guide

Contents

Introduction

This tutorial guides you through a typical Silk Performer / Silk Central integration scenario and provides you with the information you need to make the most of the integration.

Overview

Silk Performer is fully integrated with the test-planning and test-execution functionality of Silk Central. Silk Performer projects can be integrated into Silk Central test plans and executed directly from Silk Central. This allows for powerful test-result analysis and reporting. It also enables unattended Silk Performer tests, which are tests that are run automatically by Silk Central based on pre-configured schedules.

From Silk Central, Silk Performer projects can be opened directly in Silk Performer for the editing of scripts and settings. Edited Silk Performer projects can subsequently be uploaded back to Silk Central to make them available for future test executions.

Integration with Silk Central also enables access to your test plans directly from Silk Performer Workbench, allowing you to easily open, edit, and upload your Silk Central test projects from Silk Performer. Access to Silk Central is completely Internet-enabled, allowing access from a Web browser or Silk Performer Workbench.

The results of the most recent test runs in Silk Performer, excluding recent Try Script runs, can also be uploaded to Silk Central and associated with tests. To do this, Silk Performer searches its directories for the most recent directory that includes a baseline report file. The files in the directory are then uploaded to Silk Central.

Load Test Results

Silk Performer stores all data that are generated during load test runs (including time series data, TrueLog files, result files, and summary data) in the central Silk Central repository, which is easily accessible via the Silk Central Web GUI.

Cross Load-Test Analysis

Silk Performance Explorer offers a cross load-test comparison-report facility. Simply browse your Silk Central test plans with Performance Explorer and select up to four load-test runs for comparison. Or select local test results for comparison. Heat fields and rankings help you analyze the results of your optimization efforts across test runs.

Source Control Integration

Silk Performer integrates with the source control profiles of Silk Central. For additional information regarding Silk Central source control profiles, refer to *Silk Central Help.* Source-code control functionality, such as checking files in and out and getting the latest file versions, is available directly from Silk Performer and is executed automatically when you open test projects from Silk Central or upload edited projects to Silk Central.

Enhanced Test Asset Management

Silk Performer provides additional directories for custom user data and Silk Performer include files (BDH) enabling you to better organize your test scripts and test data and to share them across multiple projects.

Dynamic Workload-Assignment to Agent Clusters

Silk Performer workload delivered by Silk Central can make use of dynamic workload-to-agent assignment. The dynamic workload-assignment functionality in Silk Performer matches your specific test requirements to the replay capabilities of available agent computers at execution time.

Working with Performance Explorer

Performance Explorer can be used for in-depth analysis of test runs. Performance Explorer results analysis can be launched directly from Silk Central through execution runs on the **Runs** page. Results analysis can also be launched from Performance Explorer. For additional information, refer to *Performance Explorer Help*.

Silk Performer Integration with Silk Central

> **Note:** This tutorial assumes that you have a working knowledge of both Silk Central and Silk Performer.

This tutorial walks you through a typical integration scenario between Silk Performer and Silk Central, demonstrating the following:

- Configuring Silk Performer for use with Silk Central.
- Setting up Silk Central tests and execution plans.
- Executing Silk Performer tests from Silk Central.
- Analyzing test results in Performance Explorer.
- Generating cross load-test reports.

Configuring Access to Silk Central

To begin this tutorial, you must configure access to Silk Central from Silk Performer.

1. Select **Settings** > **System** from the Silk Performer menu bar. The **System Settings - Workbench** dialog box displays.
2. Click the **Silk Central** tab.
3. In the **Hostname** text box, type the host name or IP address of the host where the Silk Central front-end server is running.

   Do not include `http://` in the host name.
4. In the **Port** text box, specify the number of the port on which Silk Central is listening.

   This port is typically `80` when using an ISAPI Web server. The port is typically `19120` for standalone Web server configurations.
5. Type valid user credentials in the **Username** and **Password** text boxes.
6. Click **OK**.
7. *Optional:* To configure proxy settings for your server environment, click **Internet Options** to open the **Internet Properties** dialog box. Click the **Connections** tab, then click **LAN settings**.

Configuring Silk Performer Workbench

In addition to Silk Central access, other basic Silk Performer testing configurations can be set through **Workbench** system settings. If you already have a fully configured Silk Performer installation, you may not need to configure these settings to complete this tutorial.

Configuring Directory Settings

You can specify the directories where Silk Performer files are stored. The default directories are set up during installation. You can specify alternative or additional directories on the **Directories** page. You can also define locations for the following:

- Testing scripts
- Include files
- Test result files
- User-data files

- Certificate files
- Random-data files

Configuring Directory Settings

1. In the Silk Performer menu, click **Settings** > **System** .
2. Click the **Directories** tab.
3. Use the **File locations** area to specify the directories where various Silk Performer file types are stored. Default directories are set during installation.
4. In the **Projects** field, specify the directory where the project files are to be saved.
5. In the **Custom user data files** (`.pem, .rnd, .csv, .txt, .idl`) field, specify the directory where your self-created user data files (`.csv`), certificate files (`.pem`), random data files (`.rnd`), text files (`.txt`), and Interface Definition Language files (`.idl`) are located.

   In contrast to the specified **User data files** location, this location is used for your own user data files. As with Silk Performer pre-defined user data files, custom user data files in this directory can be shared across multiple projects.
6. In the **Custom include files** (`.bdh`) field, specify the directory where your self-created include files (`.bdh`) are located.

   In contrast to the specified **Include files** location, this location is used for your own include files. As with Silk Performer pre-defined include files, custom include files in this directory can be shared across multiple projects.

Source Code Control

If you utilize a source control system to store program sources, you will need to define a source control profile within Silk Central so that Silk Central execution servers can retrieve program sources for test executions. For additional information regarding source control settings, refer to *Silk Central Help*. For additional information regarding Silk Performer source code control (SCC), refer to *Silk Performer Help*.

Your project may require additional configurations. For additional information, please refer to *Silk Performer Help*.

Creating Your Silk Performer Project

Create a Silk Performer project of application type `Web business transaction (HTML/HTTP)`. Name the project `SampleWeb`. Within this project you will record simulated user interactions with ShopIt, the sample Web application that ships with Silk Performer.

> **Note:** This tutorial assumes that you have working knowledge of Silk Performer. For additional information regarding the use of Silk Performer, refer to *Silk Performer Help*.

Installing ShopIt V 6.0

The Silk Performer sample web application is ShopIt V 6.0. ShopIt V 6.0 simulates a simple e-commerce website with a catalog of camping merchandise that is available for simulated online purchase. Use this application to experiment with Silk Performer's web application capabilities. ShopIt V 6.0 is designed to generate errors, including missing links (due to merchandise being out of stock) and session errors.

Before you install ShopIt V 6.0, refer to the *Release Notes* to ensure that your system supports the use of ShopIt V 6.0.

You can download the ShopIt V 6.0 setup from the *product updates site*.

1. Double-click the file `ShopItV60.exe`

Note: IIS (Internet Information Server) must be installed on the computer. For IIS 7, also install Role Services ASP and ISAPI Extensions.

2. The **Welcome** page displays. Click **Next**.

3. The **Choose Destination Location** page displays. To change the default installation directory, click **Browse**, specify a folder, and click **OK**. Click **Next**.

4. Enter the name of the virtual directory for the web application. This is the name of the directory that will be created on the web server. Click **Next**.

5. Setup installs the files and configures IIS to run the ShopIt V 6.0 web application.

6. The **Installation Complete** dialog displays. Click **Finish**.

7. For IIS 7: Add the virtual directory to IIS manually.

   - Alias: ShopItV60
   - Physical path: Install directory of ShopIt.

     Note: Make sure that ASP is available in IIS.

The ShopIt V 6.0 web application is now ready for use. You can access ShopIt V 6.0 with a browser of your choice by entering the following URL:

```
http://<computer name>/<virtual directory name>/
```

If the name of your computer is `JohnSmith` and you have not modified the default value `ShopItV60` for the virtual directory, the URL is:

```
http://JohnSmith/ShopItV60/
```

Outlining a Project

1. Click **Start here** on the Silk Performer workflow bar.

   Note: If another project is already open, choose **File** > **New Project** from the menu bar and confirm that you want to close your currently open project.

   The **Workflow - Outline Project** dialog box opens.

2. In the **Name** text box, enter a name for your project.

3. Enter an optional project description in **Description**.

4. From the **Type** menu tree, choose the type of application that you want to use in your test.

   Note: If you are testing a Web application, choose the **Web business transaction (HTML/HTTP)** option to create simpler scripts while incorporating advanced functionality. Choose the **Web low level (HTTP)** option if you want to put the highest possible load on your application. Web low level scripts are more complex than Web business transaction scripts, which require more effort to customize. It is recommended that you use the **Web business transaction (HTML/HTTP)** instead of the **Web low level (HTTP)** scripts for browser based applications.

5. Click **Next**.

   Note: If you need to add additional resources to the project, right-click the project icon in the **Project** menu tree view. It is particularly important that all the user data files (`.csv`), random data files (`.rnd`), and `.idl` files needed by Silk Performer are set up for your project.

The **Workflow - Model Script** dialog box appears.

Recording the Sample Application

Record some interactions with the sample Web application, ShopIt, to flesh out a BDF script for your newly created `SampleWeb` project. For additional information on recording scripts, refer to the *Silk Performer Help*.

1. Click **Model Script** on the workflow bar. The *Workflow - Model Script* dialog box opens.
2. In the URL text box, type the URL of your sample application.

   For example `http://localhost/shopItV60/default.asp`.
3. Click **OK**. The Silk Performer Recorder starts.
4. Interact with the sample application.
5. In the Silk Performer Recorder, click **Stop** to stop the recorder.
6. Save the newly recorded script as `SampleWeb.bdf`.

Executing a Try Script Run

Evaluate the readiness of the test script that you have recorded by executing a Try Script run. Try Script runs are Silk Performer trial test runs that determine if recorded scripts run without error. If your script can not be run, use TrueLog Explorer to customize your script. For additional information, refer to *TrueLog Explorer Help*.

✎ **Note:** ShopIt is designed to generate errors, including missing Web links and session errors. If you experience a session error, proceed as explained in *Customizing Your Test Script in TrueLog Explorer*.

Executing a Try Script Run from Silk Performer

1. On the Silk Performer workflow bar, click **Try Script**. The **Try Script** dialog box opens.
2. Click **Run**.

   ✎ **Note:** To open TrueLog Explorer during the Try Script run, select the **Animated Run with TrueLog Explorer** check box on the **Try Script** dialog box.

Customizing Your Test Script in TrueLog Explorer

The ShopIt sample application is designed to generate errors, including missing Web links due to out-of-stock merchandise and session errors. Session errors result when session information embedded in Javascript (for example a session ID) is not detected by Silk Performer context management during script replay.

Your `SampleWeb.bdh` script must be customized before it can run correctly because it contains a static session ID. To customize the script, you need to utilize TrueLog Explorer, the Silk Performer script-customization tool.

✎ **Note:** For additional information on TrueLog Explorer and the customization of test scripts, refer to TrueLog Explorer Help.

Customizing Session Handling

You can create a new recording rule in TrueLog Explorer using values that you have specified in a parsing function. For example, creating a recording rule from a session-customization parsing function allows you to record an application while avoiding session customization issues entirely.

1. Select a TrueLog in the menu tree.
2. Click **Customize Session Handling** on the workflow bar. The **Workflow - Customize Session Handling** dialog box opens.
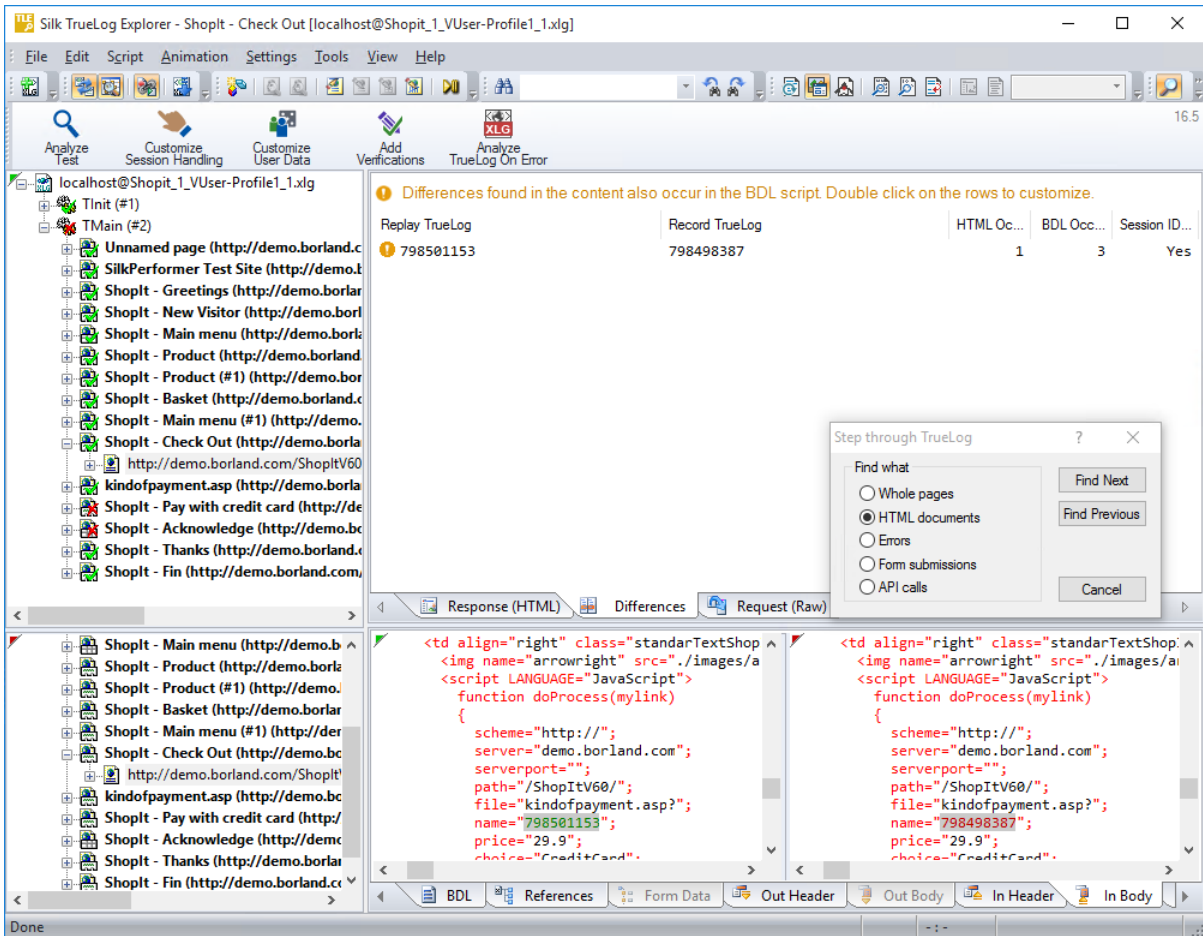3. Click the **Find differences** link to view a differences table on the **Differences** page.

   This step reveals instances in which the server responded with different (or dynamic) information during the replay session compared to of the record session. Such static information may need to be replaced with a variable.

Note: When the corresponding record TrueLog is not already open, a dialog box opens asking if you want to have the corresponding record TrueLog opened.

a) Click **OK**.

4. Step through the HTML server responses using the **Step Through TrueLog** dialog box.

   Recorded responses are displayed alongside the corresponding replayed responses. Only those differences that indicate that static information is included in the test script and being sent back to the server need to be parsed. For example, a difference between replay and record sessions might be due to an e-commerce site running out of stocked merchandise. Such a difference would not be appropriate for script customization because it is not session related.

Tip: The column headers on the **Differences** page offer helpful mouse-over tooltips that describe the elements contained in each column.

5. Double-click one of the errors listed on **Differences** page. The **Insert parsing function** dialog box opens with the boundary values already inserted. There is no need to locate and enter these values manually.

6. *Optional:* Click **Create Recording Rule** to create a recording rule based on the values of the parsing function. This enables you to record the application in the future without having to care about session customization again.

7. Back on the **Insert parsing function** dialog box, click **OK**.

8. Click **Customize Session Handling** on the workflow bar once the script has been successfully modified.

9. Click **Try Script Run** to see if the script runs correctly now that session handling has been modified. A new Try Script run is initiated.

10. Analyze the results of the subsequent test run to determine whether or not session handling customization was successful.

Uploading Your Project to Silk Central

After you have created a Silk Performer project, recorded a script based on the sample Web application, customized the session data in your recorded script using TrueLog Explorer, and confirmed that your test script can be run via a Try Script run, you need to create a Silk Central test to which you can upload your Silk Performer project. Name the new test *ShopItV_Increasing*.

> **Note:** For the purposes of this tutorial it is assumed that you are familiar with the functionality of Silk Central. For additional information regarding the use of Silk Central, refer to *Silk Central Help*.

Setting Up a Silk Central Project and Test Container

Before you can upload your Silk Performer project to Silk Central, you must configure a Silk Central project with a test structure to which you can upload your Silk Performer project.

1. From Silk Central, create a new Silk Central project called *Sample Web Test*.
2. Within the new project, create a new test container called *Silk Performer Tests*.

Uploading Projects to Silk Central

Before uploading a Silk Performer project to Silk Central, ensure that you have properly configured the project's workload through Silk Performer. Workload settings are subsequently specified along with uploaded projects in the Silk Performer **Test Properties** portion of the **Properties** page in Silk Central's **Tests** unit.

1. In Silk Performer, open the project that you want to upload and choose **File** > **Upload Project to Silk Central** .

   If the project has previously been uploaded to Silk Central, the association with a Silk Central test is already known and preselected. Click **Next** and go to step 3.

2. From the **Projects** list, select the Silk Central project to which you want to upload the Silk Performer project and click **Next**.

   You can select an existing test to replace or you can create a new test by right-clicking the appropriate folder or container choosing **New Child Test**.

   You can also create a new folder within an existing test container by right-clicking a container and choosing **New Child Test Folder**.

   > **Note:** Newly created tests and folders are displayed with bold text to indicate that they have not yet been written to the Silk Central database. If you click **Cancel**, the items you have created are not saved.

3. Check the **Enable Results Upload** check box to select specific test run results to upload with the project and then click **Next**.

   If you do not want to upload test run results at this time, do not check the **Enable Results Upload** check box. Instead, click **Finish**.

4. On the **Select results** page, check the appropriate check boxes in the **Results** list to select the results that you want to upload along with the project and then click **Next**.

5. Specify version and build numbers for the assigned product to which the uploaded results belong.

6. Click **Finish** to upload the project to Silk Central. The newly uploaded items appear in Silk Central's test and **Project** menu tree.

Associating a Test Script with a Test in Silk Central

To avoid associating a particular Silk Performer test script, which is stored in a BDF file, with the Silk Performer project, associate the test script with a Silk Performer test in Silk Central. When trying to execute the test script through Silk Central, assign the Silk Performer project to the test and then specify the correct Workload that you want to use. You do not directly point to the BDF script itself.

To associate a Silk Performer test script with a Silk Performer test in Silk Central:

1. In the menu, click **Tests** > **Details View** .
2. In the **Tests** tree, select a Silk Performer test.
3. Click the **Properties** tab.
4. Click **Edit**. The **Edit Test** dialog box opens.
5. Click **Next**. The **Silk Performer Test Properties - Select Project** dialog box opens.
6. Click **Browse** and select the Silk Performer project. The Silk Performer project is assigned to the test.
7. Click **Next**. The **Silk Performer Test Properties - Select Workload** dialog box opens.
8. In the **Workload** list box, select the workload that you want to use.

The Workload you choose contains the correct setting under which to run the script file. The users to be simulated are defined in the workload section of the test script. Within the Workload a user is described by the transactions to be called and the frequency of those transactions. Therefore you specify the workload file that contains the script that you want to execute.

Creating an Execution Plan

Once you have defined your test (*ShopItV_Increasing*) you need to create an execution plan for your Silk Performer test. Execution plans specify when and where tests are to be executed.

Creating an Execution Plan

1. In Silk Central, navigate to the **Execution Planning** unit.
2. Click **Details View**.
3. Select the *Sample Web Test* project in the **Execution** tree.
4. Click **New Child Execution Plan** on the toolbar. The **New Execution Plan** dialog box displays.
5. Type `ShopItV_Increasing_execution` in the **Name** text box.
6. *Optional:* Type a description for the execution plan in the **Description** text box.
7. Accept the test container that is selected by default in the **Test Container** list box.
   The only test container that exists in this project is *Silk Performer Tests*.
8. Accept the product version and build that are selected by default in the **Version** and **Build** list boxes.
   The version and build that are associated with the test container are automatically populated.
9. Accept the default priority for the execution plan, which is `Medium`.
10. Click **OK**. The **Execution** tree is updated with your newly created execution plan.

Assigning a Test to an Execution Plan

Assign your *ShopItV_Increasing* test to your *ShopItV_Increasing_execution* execution plan.

1. Navigate to **Silk Central** > **Execution Planning** .
2. Click the **Details View** icon.
3. In the **Execution** tree, select the *ShopItV_Increasing_execution* execution plan.
4. Click the **Assigned Tests** tab.
5. Click **Manual assignment**. All tests of the *Silk Performer Tests* test container are displayed in the right-hand **Test Plan** tree.
6. Click the assign arrow of the *ShopItV_Increasing* test.

Scheduling Your Execution Plan

After you have assigned a test to your execution plan, define the schedule upon which the execution plan is to be executed.

1. Navigate to **Silk Central** > **Execution Planning** .
2. Click the **Details View** icon.
3. In the **Execution** tree, select the *ShopItV_Increasing_execution* execution plan.
4. Click the **Schedule** tab.
5. Click the **Custom** option button. The calendar tool displays.
6. Click the **Edit** icon next to **From** to specify the date and time when the execution schedule is to begin.

   For the purpose of this tutorial, accept the default settings.
7. From the Interval list boxes, select the interval at which the execution plan is to be executed.

   To execute the execution plan daily, select `1` for day, and `0` for hour and minute.
8. In the **Run** section, specify the duration of the schedule.

   For the purpose of this tutorial, click the **Forever** option button.

   Note: The calendar tool also enables you to define *Exclusions* and *Definite Runs*. Exclusions are regularly occurring time periods during which executions are not executed, for example weekly planned system downtime or weekends. Definite Runs are executions that are executed independently of the configured schedule.

Assigning an Execution Server

Assign an execution server to execute your Silk Performer execution plan. The **Deployment** page in the **Execution Planning** unit of Silk Central displays all of the execution servers that are configured for the selected project and enables you to specify the execution servers on which the selected execution plan is to be executed.

   Note: New execution servers are set up at **Administration** > **Locations** . For additional information, refer to *Silk Central Help*.

1. Navigate to **Silk Central** > **Execution Planning** .
2. Click the **Details View** icon.
3. In the **Execution** tree, select the *ShopItV_Increasing_execution* execution plan.
4. Click the **Deployment** tab. The **Deployment** page displays.
5. In the **Execution environment** section of the page, click **Edit**. The **Assign Keywords** dialog box displays.
6. Select a keyword from or type a new keyword into the **Select or enter keywords** list box.

   The keyword must match the keywords of one or more execution servers.
7. Click **OK**. The keywords are assigned to the execution plan.

Assigning Workload to Agent Clusters

✎ **Note:** To use this optional part of the tutorial, you must have a test environment with multiple test agents that offer varying software capabilities.

You can distribute the workload that is delivered from Silk Central to Silk Performer with dynamic workload-to-agent assignment. The dynamic workload-assignment functionality of Silk Performer matches your specific test requirements to the replay capabilities of the available agent computers at execution time.

Assigning workload to clusters of agents is often preferable to assigning workload to individual agents. *Agent clusters* are groups of agents that can share the same capabilities, such as a Java Development Kit (JDK) installation, an ODBC client, or a Citrix client, or they may consist of agents that have entirely different capabilities, such as the ability to divide the machines in a lab into separate agent groups that can support different teams. With this approach, you select a cluster of agents that is capable of executing the required workload when you configure the workload of your test.

Within Silk Performer you specify the name of an agent cluster that should deliver the workload for your test. Silk Central provides the list of agent computers that are assigned to the specified cluster. The workload is assigned to specific agents at the moment of execution, based on the capabilities of the individual agents.

When a Silk Performer execution plan is executed, a Silk Central test agent-clusters XML file is checked out from Silk Central and used for dynamic workload-assignment.

Defining the Capabilities of Your Agent Computers

To facilitate dynamic workload-assignment you must tag each of your agent computers with one or more replay capabilities. For each agent capability, you also need to specify the maximum number of virtual users that the agent is capable of driving.

Replay Capability	Description and Prerequisites
General	Any replay feature that does not require an installation, for example Web, IIOP, COM, or ADO.
Java	A Java run-time environment is available for Java Framework, Oracle Forms, Oracle Applications, Jacada, and Jolt.
ODBC	An ODBC client is installed.
Oracle OCI	An Oracle client is installed.
SAPGUI	A SAPGUI client is installed.
Citrix	A Citrix client is installed.
Tuxedo	A Tuxedo client is installed.
.Net	The .Net Framework is installed.
GuiL Test	Terminal Services is installed and running on this agent. Silk Test and the system under test must be installed.

Defining the Capabilities of an Agent Computer

1. Navigate to **Settings** > **System** .
2. Click **Agents**.
3. Click the **Agent Pool** tab.
4. Select the agent for which you want to define capabilities.

5. Click **Properties**.
6. Click the **Capabilities** tab.

   The **Capabilities** page is pre-populated with suggested replay-capability values that are based on the specifications of the installed hardware.
7. Adjust the **Max. User** values for each capability type, depending on the agent.

   Values from `0` to `9999` are valid.
8. Click **OK** to save your settings.

Preparing an Agent Clusters File for Silk Central

This section describes how to prepare an agent clusters file for Silk Central.

Exporting Agent Pool

Before you can create a Silk Central test-agent cluster file, you need to output the connection properties, capabilities, and system information of the available agents in your agent pool.

1. Navigate to **System** > **Settings** > **Agents** > **Agent Pool** .
2. Click the **Export Agent Pool** button.
3. Specify an appropriate path and filename to save the contents of your agent pool as an XML file.

Example agent pool file

This XML file includes an `AgentPool` root element. Within the root element are `Agent` elements, one for each agent in the agent pool. Within each `Agent` element are elements that convey the connection properties, capabilities, and system information of that agent.

```
<?xml version='1.0' encoding='UTF-8'?>
<AgentPool name="LocalPool">
  <Agent id="LAB108">
    <ConnProperty name="ConnectPort">19200</ConnProperty>
    <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
    <ConnProperty name="IpAddress">192.168.1.108</ConnProperty>
    <ConnProperty name="LastConnectStatus">5</ConnProperty>
    <ConnProperty name="UseAuthentication">false</ConnProperty>
    <Capability maxVU="390" name=".Net"></Capability>
    <Capability maxVU="100" name="Browser-driven"></Capability>
    <Capability maxVU="39" name="Citrix"></Capability>
    <Capability maxVU="3900" name="General"></Capability>
    <Capability maxVU="39" name="GuiLTest"></Capability>
    <Capability maxVU="390" name="Java"></Capability>
    <Capability maxVU="390" name="ODBC"></Capability>
    <Capability maxVU="390" name="Oracle OCI"></Capability>
    <Capability maxVU="39" name="SAPGUI"></Capability>
    <Capability maxVU="390" name="Tuxedo"></Capability>
    <SysInfo name="AgentRAC">7803300</SysInfo>
    <SysInfo name="AgentVersion">7.8.0.3332</SysInfo>
    <SysInfo name="Memory">2039 MB</SysInfo>
    <SysInfo name="ProcType"></SysInfo>
    <SysInfo name="ProcessorCount">1</SysInfo>
    <SysInfo name="ProcessorSpeed">3200 MHz</SysInfo>
    <SysInfo name="ServicePack"></SysInfo>
    <SysInfo name="SysVersion">5.0</SysInfo>
    <SysInfo name="System">WinNT</SysInfo>
  </Agent>
  <Agent id="lab116">
    <ConnProperty name="AuthPassword"></ConnProperty>
    <ConnProperty name="ConnectPort">19200</ConnProperty>
    <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
```

```
      <ConnProperty name="EncryptionSSL">false</ConnProperty>
      <ConnProperty name="HTTPTunnel">:8080</ConnProperty>
      <ConnProperty name="IpAddress">192.168.1.116</ConnProperty>
      <ConnProperty name="LastConnectStatus">5</ConnProperty>
      <ConnProperty name="SOCKSTunnel">:1080</ConnProperty>
      <ConnProperty name="UseAuthentication">false</ConnProperty>
      <ConnProperty name="UseHttpTunnel">false</ConnProperty>
      <ConnProperty name="UseSocksTunnel">false</ConnProperty>
      <Capability maxVU="380" name=".Net"></Capability>
      <Capability maxVU="100" name="Browser-driven"></Capability>
      <Capability maxVU="38" name="Citrix"></Capability>
      <Capability maxVU="3800" name="General"></Capability>
      <Capability maxVU="38" name="GuiLTest"></Capability>
      <Capability maxVU="380" name="Java"></Capability>
      <Capability maxVU="380" name="ODBC"></Capability>
      <Capability maxVU="380" name="Oracle OCI"></Capability>
      <Capability maxVU="38" name="SAPGUI"></Capability>
      <Capability maxVU="380" name="Tuxedo"></Capability>
      <SysInfo name="AgentRAC">7803300</SysInfo>
      <SysInfo name="AgentVersion">7.8.0.3343</SysInfo>
      <SysInfo name="Memory">1983 MB</SysInfo>
      <SysInfo name="ProcType">Intel Pentium IV</SysInfo>
      <SysInfo name="ProcessorCount">2</SysInfo>
      <SysInfo name="ProcessorSpeed">3200 MHz</SysInfo>
      <SysInfo name="ServicePack">Service Pack 2</SysInfo>
      <SysInfo name="SysVersion">5.2</SysInfo>
      <SysInfo name="System">WinNT</SysInfo>
    </Agent>
    <Agent id="lab125">
      <ConnProperty name="ConnectPort">19200</ConnProperty>
      <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
      <ConnProperty name="IpAddress">192.168.1.125</ConnProperty>
      <ConnProperty name="LastConnectStatus">5</ConnProperty>
      <ConnProperty name="UseAuthentication">false</ConnProperty>
      <Capability maxVU="650" name=".Net"></Capability>
                  <Capability maxVU="100" name="Browser-driven"></
Capability>
      <Capability maxVU="65" name="Citrix"></Capability>
      <Capability maxVU="6500" name="General"></Capability>
      <Capability maxVU="65" name="GuiLTest"></Capability>
      <Capability maxVU="650" name="Java"></Capability>
      <Capability maxVU="650" name="ODBC"></Capability>
      <Capability maxVU="650" name="Oracle OCI"></Capability>
      <Capability maxVU="65" name="SAPGUI"></Capability>
      <Capability maxVU="650" name="Tuxedo"></Capability>
      <SysInfo name="AgentRAC">7803300</SysInfo>
      <SysInfo name="AgentVersion">7.8.0.3371</SysInfo>
      <SysInfo name="Memory">3318 MB</SysInfo>
      <SysInfo name="ProcType"></SysInfo>
      <SysInfo name="ProcessorCount">2</SysInfo>
      <SysInfo name="ProcessorSpeed">3000 MHz</SysInfo>
      <SysInfo name="ServicePack"></SysInfo>
      <SysInfo name="SysVersion">5.2</SysInfo>
      <SysInfo name="System">WinNT</SysInfo>
    </Agent>
    <Agent id="lab47">
      <ConnProperty name="ConnectPort">19200</ConnProperty>
      <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
      <ConnProperty name="IpAddress">192.168.1.47</ConnProperty>
      <ConnProperty name="LastConnectStatus">5</ConnProperty>
      <ConnProperty name="UseAuthentication">false</ConnProperty>
      <Capability maxVU="180" name=".Net"></Capability>
                  <Capability maxVU="100" name="Browser-driven"></
Capability>
```

```
        <Capability maxVU="18" name="Citrix"></Capability>
        <Capability maxVU="1800" name="General"></Capability>
        <Capability maxVU="18" name="GuiLTest"></Capability>
        <Capability maxVU="180" name="Java"></Capability>
        <Capability maxVU="180" name="ODBC"></Capability>
        <Capability maxVU="180" name="Oracle OCI"></Capability>
        <Capability maxVU="18" name="SAPGUI"></Capability>
        <Capability maxVU="180" name="Tuxedo"></Capability>
        <SysInfo name="AgentRAC">7803300</SysInfo>
        <SysInfo name="AgentVersion">7.8.0.3414</SysInfo>
        <SysInfo name="Memory">1007 MB</SysInfo>
        <SysInfo name="ProcType"></SysInfo>
        <SysInfo name="ProcessorCount">2</SysInfo>
        <SysInfo name="ProcessorSpeed">2593 MHz</SysInfo>
        <SysInfo name="ServicePack"></SysInfo>
        <SysInfo name="SysVersion">5.1</SysInfo>
        <SysInfo name="System">WinNT</SysInfo>
    </Agent>
</AgentPool>
```

You can now use the XML data in the exported agent-pool file to create a Silk Central test-agent clusters file.

Creating a Silk Central Agent-Clusters File

Before you can complete this task you must export the Silk Performer agent pool as an XML file (which includes the connection properties, capabilities, and system information of an agent) for each agent in your agent pool.

You only need to create one Silk Central agent-clusters file. The file may contain one or more agent clusters, each of which specifies its associated agents including their capabilities, connection properties and system information. The contents of the file you create will be available to all Silk Performer users on the **Workload Configuration** dialog when they select the **Dynamic assignment to Silk Central agent cluster** option.

You must create a Silk Central agent-clusters file if you intend to run your test against a Silk Central agent cluster (this is configured by clicking the `Dynamic assignment to Silk Central agent cluster` button on the **Workload Configuration** > **Agent Assignment** tab). Workloads that use a Silk Central agent cluster can be executed from within the Silk Performer Workbench and they can be scheduled as Silk Central tests.

1. Create an empty XML file on your local system.

   This file must be accessible by Silk Central. It can be placed under source control within your Silk Central directory structure.

2. Use the contents of an exported agent pool file to build the agent-clusters file as structured in the example below.

   The contents of an exported agent-pool file and the agent-clusters file are nearly identical, so this typically only involves enclosing the `<AgentPool/>` elements of the exported agent-pool file within a `<SctmAgentClusters/>` element.

   > **Example of a manually created Silk Central agent-clusters file**
   >
   > This manually created agent-clusters file includes a `SctmAgentClusters` root element. Within the root element are `AgentPool` elements, one for each agent pool in the cluster. Within each `AgentPool` element are `Agent` elements that convey the connection properties, capabilities, and system information of the individual agents.
   >
   > ```
   > <?xml version='1.0' encoding='UTF-8'?>
   >
   > <SctmAgentClusters>
   > ```

```xml
<AgentPool name="cluster_1">
  <Agent id="LAB100">
    <ConnProperty name="ConnectPort">19200</ConnProperty>
    <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
    <ConnProperty name="IpAddress">192.168.1.100</ConnProperty>
    <ConnProperty name="LastConnectStatus">5</ConnProperty>
    <ConnProperty name="UseAuthentication">false</ConnProperty>
    <Capability maxVU="390" name=".Net"></Capability>
                <Capability maxVU="100" name="Browser-driven"></
Capability>
    <Capability maxVU="39" name="Citrix"></Capability>
    <Capability maxVU="3900" name="General"></Capability>
    <Capability maxVU="39" name="GuiLTest"></Capability>
    <Capability maxVU="390" name="Java"></Capability>
    <Capability maxVU="390" name="ODBC"></Capability>
    <Capability maxVU="390" name="Oracle OCI"></Capability>
    <Capability maxVU="39" name="SAPGUI"></Capability>
    <Capability maxVU="390" name="Tuxedo"></Capability>
    <SysInfo name="AgentRAC">7803300</SysInfo>
    <SysInfo name="AgentVersion">7.8.0.3332</SysInfo>
    <SysInfo name="Memory">2039 MB</SysInfo>
    <SysInfo name="ProcType"></SysInfo>
    <SysInfo name="ProcessorCount">1</SysInfo>
    <SysInfo name="ProcessorSpeed">3200 MHz</SysInfo>
    <SysInfo name="ServicePack"></SysInfo>
    <SysInfo name="SysVersion">5.0</SysInfo>
    <SysInfo name="System">WinNT</SysInfo>
  </Agent>
  <Agent id="lab101">
    <ConnProperty name="AuthPassword"></ConnProperty>
    <ConnProperty name="ConnectPort">19200</ConnProperty>
    <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
    <ConnProperty name="EncryptionSSL">false</ConnProperty>
    <ConnProperty name="HTTPTunnel">:8080</ConnProperty>
    <ConnProperty name="IpAddress">192.168.1.101</ConnProperty>
    <ConnProperty name="LastConnectStatus">5</ConnProperty>
    <ConnProperty name="SOCKSTunnel">:1080</ConnProperty>
    <ConnProperty name="UseAuthentication">false</ConnProperty>
    <ConnProperty name="UseHttpTunnel">false</ConnProperty>
    <ConnProperty name="UseSocksTunnel">false</ConnProperty>
    <Capability maxVU="380" name=".Net"></Capability>
                <Capability maxVU="100" name="Browser-driven"></
Capability>
    <Capability maxVU="38" name="Citrix"></Capability>
    <Capability maxVU="3800" name="General"></Capability>
    <Capability maxVU="38" name="GuiLTest"></Capability>
    <Capability maxVU="380" name="Java"></Capability>
    <Capability maxVU="380" name="ODBC"></Capability>
    <Capability maxVU="380" name="Oracle OCI"></Capability>
    <Capability maxVU="38" name="SAPGUI"></Capability>
    <Capability maxVU="380" name="Tuxedo"></Capability>
    <SysInfo name="AgentRAC">7803300</SysInfo>
    <SysInfo name="AgentVersion">7.8.0.3343</SysInfo>
    <SysInfo name="Memory">1983 MB</SysInfo>
    <SysInfo name="ProcType">Intel Pentium IV</SysInfo>
    <SysInfo name="ProcessorCount">2</SysInfo>
    <SysInfo name="ProcessorSpeed">3200 MHz</SysInfo>
    <SysInfo name="ServicePack">Service Pack 2</SysInfo>
    <SysInfo name="SysVersion">5.2</SysInfo>
    <SysInfo name="System">WinNT</SysInfo>
  </Agent>
</AgentPool>
<AgentPool name="cluster_2">
  <Agent id="LAB200">
```

```xml
      <ConnProperty name="ConnectPort">19200</ConnProperty>
      <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
      <ConnProperty name="IpAddress">192.168.2.200</ConnProperty>
      <ConnProperty name="LastConnectStatus">5</ConnProperty>
      <ConnProperty name="UseAuthentication">false</ConnProperty>
      <Capability maxVU="390" name=".Net"></Capability>
                  <Capability maxVU="100" name="Browser-driven"></
Capability>
      <Capability maxVU="39" name="Citrix"></Capability>
      <Capability maxVU="3900" name="General"></Capability>
      <Capability maxVU="39" name="GuiLTest"></Capability>
      <Capability maxVU="390" name="Java"></Capability>
      <Capability maxVU="390" name="ODBC"></Capability>
      <Capability maxVU="390" name="Oracle OCI"></Capability>
      <Capability maxVU="39" name="SAPGUI"></Capability>
      <Capability maxVU="390" name="Tuxedo"></Capability>
      <SysInfo name="AgentRAC">7803300</SysInfo>
      <SysInfo name="AgentVersion">7.8.0.3332</SysInfo>
      <SysInfo name="Memory">2039 MB</SysInfo>
      <SysInfo name="ProcType"></SysInfo>
      <SysInfo name="ProcessorCount">1</SysInfo>
      <SysInfo name="ProcessorSpeed">3200 MHz</SysInfo>
      <SysInfo name="ServicePack"></SysInfo>
      <SysInfo name="SysVersion">5.0</SysInfo>
      <SysInfo name="System">WinNT</SysInfo>
    </Agent>
    <Agent id="lab201">
      <ConnProperty name="AuthPassword"></ConnProperty>
      <ConnProperty name="ConnectPort">19200</ConnProperty>
      <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
      <ConnProperty name="EncryptionSSL">false</ConnProperty>
      <ConnProperty name="HTTPTunnel">:8080</ConnProperty>
      <ConnProperty name="IpAddress">192.168.2.201</ConnProperty>
      <ConnProperty name="LastConnectStatus">5</ConnProperty>
      <ConnProperty name="SOCKSTunnel">:1080</ConnProperty>
      <ConnProperty name="UseAuthentication">false</ConnProperty>
      <ConnProperty name="UseHttpTunnel">false</ConnProperty>
      <ConnProperty name="UseSocksTunnel">false</ConnProperty>
      <Capability maxVU="380" name=".Net"></Capability>
                  <Capability maxVU="100" name="Browser-driven"></
Capability>
      <Capability maxVU="38" name="Citrix"></Capability>
      <Capability maxVU="3800" name="General"></Capability>
      <Capability maxVU="38" name="GuiLTest"></Capability>
      <Capability maxVU="380" name="Java"></Capability>
      <Capability maxVU="380" name="ODBC"></Capability>
      <Capability maxVU="380" name="Oracle OCI"></Capability>
      <Capability maxVU="38" name="SAPGUI"></Capability>
      <Capability maxVU="380" name="Tuxedo"></Capability>
      <SysInfo name="AgentRAC">7803300</SysInfo>
      <SysInfo name="AgentVersion">7.8.0.3343</SysInfo>
      <SysInfo name="Memory">1983 MB</SysInfo>
      <SysInfo name="ProcType">Intel Pentium IV</SysInfo>
      <SysInfo name="ProcessorCount">2</SysInfo>
      <SysInfo name="ProcessorSpeed">3200 MHz</SysInfo>
      <SysInfo name="ServicePack">Service Pack 2</SysInfo>
      <SysInfo name="SysVersion">5.2</SysInfo>
      <SysInfo name="System">WinNT</SysInfo>
    </Agent>
    <Agent id="lab203">
      <ConnProperty name="ConnectPort">19200</ConnProperty>
      <ConnProperty name="ConnectSecurePort">19201</ConnProperty>
      <ConnProperty name="IpAddress">192.168.2.203</ConnProperty>
      <ConnProperty name="LastConnectStatus">5</ConnProperty>
```

```
      <ConnProperty name="UseAuthentication">false</ConnProperty>
      <Capability maxVU="650" name=".Net"></Capability>
                <Capability maxVU="100" name="Browser-driven"></
Capability>
      <Capability maxVU="65" name="Citrix"></Capability>
      <Capability maxVU="6500" name="General"></Capability>
      <Capability maxVU="65" name="GuiLTest"></Capability>
      <Capability maxVU="650" name="Java"></Capability>
      <Capability maxVU="650" name="ODBC"></Capability>
      <Capability maxVU="650" name="Oracle OCI"></Capability>
      <Capability maxVU="65" name="SAPGUI"></Capability>
      <Capability maxVU="650" name="Tuxedo"></Capability>
      <SysInfo name="AgentRAC">7803300</SysInfo>
      <SysInfo name="AgentVersion">7.8.0.3371</SysInfo>
      <SysInfo name="Memory">3318 MB</SysInfo>
      <SysInfo name="ProcType"></SysInfo>
      <SysInfo name="ProcessorCount">2</SysInfo>
      <SysInfo name="ProcessorSpeed">3000 MHz</SysInfo>
      <SysInfo name="ServicePack"></SysInfo>
      <SysInfo name="SysVersion">5.2</SysInfo>
      <SysInfo name="System">WinNT</SysInfo>
    </Agent>
  </AgentPool>
</SctmAgentClusters>
```

Once you have created an agent clusters file, you must configure Silk Central to reference the file. Silk Central will copy the file to the execution servers so that whenever a Silk Performer project with dynamic workload-assignment is executed, the project will read the file to determine how workload should be allocated to the agents within the cluster.

Uploading Your Agent-Cluster File to Silk Central

After you have created the agent-cluster file for your project, you need to upload the file from your source control system to Silk Central. When your Silk Performer project with dynamic workload-assignment is executed, the project will read the file to determine how workload should be allocated to the agents, which are the execution servers, within your cluster.

1. Navigate to **Administration** > **System Settings** .
2. Click the **Load Test Agent Clusters** tab.
3. Click **Upload**. The **Upload Agent Clusters File** dialog box opens.
4. Click **Browse** to browse to and select the agent-clusters file.
5. Click **OK** to confirm your selection.

You have now completed configuration of dynamic workload-assignment for your project.

Assigning Agents to Workload

This task can only be performed after you have configured workload for your project.

1. Click **Run Test** on the workflow bar. The **Workflow - Workload Configuration** dialog box appears.
2. Click the **Agent Assignment** tab.
3. Select the **Assignment type**:

   - **Static assignment to project agents** : Use this method to statically assign specific agent computers (rather than clusters of agents) to your project. No agent-availability check is performed with this method and agent locking is disabled. Select this method if you want to use Agents deployed in the cloud.
   - **Dynamic assignment to project agents** : With this method, workload is delivered using dynamic agent-assignment at execution time against the project's agents. Workload delivery is enhanced with

agent-capability specifications to create optimized workload-to-agent assignments based on the capabilities of each agent. Agent locking at execution time is enabled with this method. Only responding agents that are not currently used by another controller are used with this method.

- **Dynamic assignment to Silk Central agent cluster** : Silk Performer workload delivered by way of Silk Central can also use dynamic workload-to-agent assignment. Within Silk Performer you choose the name of the agent cluster (from the drop list) that should deliver your test's workload. Silk Central then provides the list of agent computers that are assigned to the cluster. Workload is then assigned to specific agents at the moment of execution based on the capabilities of the individual agents. After you connect to Silk Central, you are presented with the list of available agent clusters. In the right-most window, you can view the agents that are currently associated with the selected agent cluster.

4. Define the agents that are to deliver the workload for your test.

- If you selected **Static assignment to project agents** , you can check the **Even user distribution** check box to distribute all existing user types evenly across all agents, depending on each agent's general replay capabilities. To use agents that run as virtual machines in the cloud, check the **Use cloud agents** check box. Click **Cloud Agent Manager** to manage your agents in the cloud.
- If you selected **Dynamic assignment to project agents** , workload is delivered automatically using dynamic agent-assignment at execution time against the project's agents.
- If you selected **Dynamic assignment to Silk Central agent cluster** , you are asked to log in to Silk Central. When you are logged in, you can select the available agent cluster.

The **Agents** list box displays the available agents.

5. Check the **Agent resource utilization** check box to assign a maximum percentage of total virtual users that each agent can run based on the agent's replay capabilities.

6. Check the **Balance load across agents** check box to apportion workload across agents.

7. If you selected **Static assignment to project agents** , use the lower window of the **Agent Assignment** page to define workload assignments for user groups.

   Note: Available options vary depending on the selected workload model.

8. Click **User Distribution Overview** to view the assignment of virtual users to the agent computers that are currently available and then click **Close**.

9. Click **OK** to save your settings.

Workload will be assigned to agents based on the agent-assignment settings you have configured. If there are not enough agents with the required capabilities to deliver the required workload, you will be presented with an error message and details regarding the user types that did not receive an agent assignment.

Successfully Running a Silk Performer Project in Silk Central

To successfully run your Silk Performer project you need to first run the test in Silk Performer. This generates user group information which is stored in the project file and is needed by Silk Central. To run the test in Silk Performer, click **Run Test** in the Silk Performer Workflow bar. When the test completes upload it again to Silk Central.

If you do not run the test in Silk Performer, an error message box with the following text might display:
```
Silk Performer Execution Error – LoadtestController: 3109 – No usergroup(s)
specified".
```

Running Your Test

If you have successfully completed all the preceding procedures, you can now run your Silk Performer test.

1. Navigate to **Silk Central** > **Execution Planning** .
2. Click the **Details View** icon.
3. In the **Execution** tree, select the *ShopItV_Increasing_execution* execution plan.
4. Click **Run** on the toolbar. Your execution plan is queued on the execution server that you have specified and the **Run** dialog box opens.
5. Check the **Go to Activities page** check box to open the **Activities** page.

Viewing Test Run Details

After you execute a test in Silk Central, the test-activity statistics are available on the **Activities** page. The page includes details about the recently run execution plans, the currently running execution plans, and the execution plans that are scheduled for execution.

1. Navigate to **Silk Central** > **Tests** .
2. In the **Tests** tree, select the *ShopItV_Increasing* test.
3. Click the **Runs** tab.
4. Click the **Run ID** of the last run to display the **Test Run Results** dialog box.

   The **Details** page of the dialog box includes the status of the run, Silk Performer specific details, error messages, and other details of the run.

How Silk Central Manages Silk Performer Test Results

When Silk Performer test runs are executed via Silk Central, the tests are run and the result files are generated in the usual way, with Silk Performer agents hosting the virtual users that are run during the tests. Silk Central only schedules the tests and invokes the Silk Performer replay engine.

Once a test is complete, Silk Performer results are passed to and stored within the Silk Central database. The metrics used to generate Silk Performer overview reports cannot be called directly from the Silk Central database.

> **Note:** To download the results of a Silk Performer test run via Silk Central, select the test in **Tests** > **Details View**. Click the **Runs** tab. Click **Download Results** in the **Actions** column of the desired test run.

Analyzing Results in Performance Explorer

Performance Explorer enables in-depth analysis of Silk Performer test results. The **Analyze Results** option in Silk Central downloads the selected results to Performance Explorer where you can evaluate the results of your optimization efforts. Performance Explorer also enables you to compare statistics from multiple test runs side-by-side in cross load-test reports. For additional information, see *Creating a Cross Load-Test Report*.

> **Note:** For additional information regarding the use of Performance Explorer and the integration of Performance Explorer with Silk Central, refer to *Performance Explorer Help*.

1. Navigate to **Silk Central** > **Tests** .
2. In the **Tests** tree, select the *ShopItV_Increasing* test.
3. Click the **Runs** tab.
4. In the **Actions** column, click the **Analyze Results...** icon. A download dialog box opens, showing you the name of the Performance Explorer command file (`.sppecmd`) that you are about to download.
5. Click **Open** to open the results in Performance Explorer.

Performance Explorer opens, connected directly to your Silk Central installation, and fetches the results of the selected execution run. The results are displayed in an overview report. For additional information regarding overview reports, refer to the *Performance Explorer Help*.

Running an Attended Test

You may need to run Silk Performer tests that are not tied to Silk Central schedules. For this reason the integration of Silk Performer with Silk Central supports *attended tests*. Attended tests are Silk Performer tests that are executed manually in Silk Performer and are not executed automatically based on a predefined schedule in Silk Central.

When you **open** a Silk Central project, you need to check out the project from a source control profile, edit the project, and check the project back in to Silk Central. When you import a project, you only need to download a copy of the project and work with the project independently of Silk Central. Any changes you make to an imported project have no effect on Silk Central. To import a Silk Central project, you only need a Web connection.

Once you have executed an attended Silk Performer test, upload the test results to Silk Central and associate the results with a Silk Performer test. Finally, analyze the results of the test in Performance Explorer, as described in *Analyzing Results in Performance Explorer*.

Adding Another Test

To continue with this tutorial you need to add another Silk Performer test to Silk Central, as described in *Uploading Your Project to Test Manager*. In the *Silk Performer Tests* container that you created earlier, create a Silk Performer test called *ShopIt_Modem*. Upload your *SampleWeb* project to this new test.

Executing an Attended Test Run

1. Navigate to **Silk Central** > **Tests** .
2. In the **Test Plan** tree, select your new `ShopIt_Modem` test.
3. Click the **Properties** tab.
4. In the **Silk Performer Test Properties** section, click the **Run Attended Test** icon. A file download dialog box opens, asking you to confirm that you wish to run the specified Silk Performer command file, `.spwbcmd`.
5. Click **OK** to open the file in Silk Performer. Silk Performer is invoked. The **Run Attended** dialog box opens.
6. Select the target directory for your Silk Performer project and click **OK**. The **Workload Configuration** dialog box opens with the worload settings that are associated with the `SampleWeb` project..
7. Click **Run** to execute the Silk Performer test in exactly the same way as if it were executed from Silk Central as an unattended test.

   Note: For the purposes of this tutorial, vary your workload settings. For example, you might select a dynamic workload type and a user profile that simulates 56k modem users. After editing the workload settings, click **Run** to begin the test and monitor the test through Silk Performer.

Uploading Test Results to Silk Central

1. Choose **Results** > **Upload Results to Silk Central** .

   Alternatively, you can right-click a results node and choose **Upload Results to Silk Central** .

   If the project has previously been uploaded to Silk Central, the association with a Silk Central test is already known and preselected. Click **Next** and go to step 4.

2. From the **Projects** list, select the Silk Central project to which you want to upload the Silk Performer test results and click **Next**.

3. From the menu tree, select the test to which you want to upload the results and click **Next**.

   Alternatively you can right-click in the tree and choose various menu items to create a new test, child test, test folder, or child test folder to which the results are saved.

4. On the **Select results** page, check the appropriate check boxes in the **Results** list to select the results to upload along with the project.

5. Click **Next**.

6. On the **Specify product information and result status** page, specify the version and build numbers for the assigned product to which the uploaded results belong.

7. Check the checkbox **Enable automatic test result upload** to let Silk Performer upload test results automatically after each test run.

   > **Note:** You can click the **Upload path** link to start Silk Central and go directly to the selected test. Check the **Open Upload path on finish** check box to automatically perform this action after clicking **Finish**.

8. Click **Finish** to upload the results. A confirmation dialog box asks if you want to delete the local copies of the uploaded results.

9. Click **Yes**.

Creating a Cross Load-Test Report

Performance Explorer offers a cross load-test comparison report facility. You can select up to four load-test runs for comparison. Heat fields and rankings help you analyze the results of your optimization efforts across runs.

Cross load-test reports are structured like standard baseline reports. They offer summary reports and statistics regarding transactions, page timers, Web forms, rankings, and errors. For additional information regarding cross load-test reports, refer to *Performance Explorer Help*.

To compare the results of the *ShopItV_Increasing* and *ShopIt_Modem* test executions, follow the steps described in *Analyzing Results in Performance Explorer*.

> **Note:** If you have followed the steps in this tutorial, the execution results for the tests will be loaded into the **Silk Central** tab in Performance Explorer. If they are not, download the results from the **Run** tab in Silk Central, as described in *Analyzing Results in Performance Explorer*. Alternatively, you can download the results from Performance Explorer by right-clicking within the project tree and selecting **Add Silk Central Results**.

Creating Custom Comparison Reports

1. Open Performance Explorer and add results. The results are loaded into Performance Explorer and display in the tree.

2. Select a project node in the tree and click **Custom** in the **Comparison Report** group on the **Reports** tab. The report is created with all load tests that are loaded in Performance Explorer. These are the load tests that appear under the project node in the tree.

3. You can add further load test results and drag them from the tree onto the report. You can even add load tests as well as user type or region results from various projects to one custom comparison report. A maximum of eight data sets can be included in a single custom comparison report.

   > **Tip:** You can also right-click a node in the tree and use the commands in the context menu.

4. To define the results of a certain load test as the baseline, right-click the load test in the baseline pane at the bottom of the report and click **Set as baseline**.

Analyzing Cross Load-Test Reports

In cross load-test reports, the results of multiple test runs are compared side-by-side. To facilitate comparison, one run is defined as the *baseline* to which the other runs are compared. Any test run can be set as the baseline. 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.

Example Analysis of a Cross Load-Test Report

In this example, *Loadtest 1* is a load test with a workload of 56k modem users and a dynamic workload model. The test is derived from the *ShopIt_Modem* test. *Loadtest 1* is tagged as the baseline execution. *Loadtest 2* is a load test with a workload of high-speed connection users and an increasing workload model. If the heat field next to *Loadtest 2* in the cross load-test report is green, then *Loadtest 2* has performed better than *Loadtest 1*.

Move the cursor over any heat field in the cross load-test report to see the details of the heat field setting.

If *Loadtest 2* were defined as the baseline, then the heat field for *Loadtest 1* would be red, indicating that *Loadtest 1* performed significantly worse than *Loadtest 2*.

Index