## Starting a Recording
Starting a Recording ......................................................... 20

## Stopping a Recording
Stopping a Recording ....................................................... 22

## Replaying a Recording
Replaying a Recording ..................................................... 23

## Using Fast Record
Using Fast Record ........................................................... 23

### Chapter 4 Editing Scripts

**Introduction to Script Editing** ........................................... 25
**Navigation** .................................................................... 27
**Adding Instructions** ...................................................... 28
**Moving Instructions** ...................................................... 28

**Using Folders** ............................................................... 29
**Repeating Folders** .......................................................... 30
**Iterating Folders** ............................................................ 30
**Conditional Folders** ....................................................... 30
**Do Until Satisfied Example** ............................................. 31
**Creating Conditional Folders** ........................................... 31

**Folder Properties** .......................................................... 32

**Synchronization Points** ................................................... 33
**Match Synchronization Points** ......................................... 35
**Search Synchronization Points** ......................................... 36

**Editing Keyboard Input** ................................................... 37
**Keyboard Examples** .......................................................... 38

**Editing Mouse Input** ........................................................ 39

**Creating Variables** ........................................................ 41
**Importing Variables** ........................................................ 42

**Using Microsoft JScript** ................................................... 42
**Supported Built-in JScript** ............................................... 43
**User.SetSpeed() Example** ............................................... 44

### Chapter 5 Running a Load Test

**Creating a Load** ............................................................ 45
**Concurrency Control** ....................................................... 46
**Rate Control** ............................................................... 46
**Add a Load** ............................................................... 46
**Adding Load Control Rules** ............................................ 46

**Starting a Test** .............................................................. 48
**Viewing the Test Windows** .............................................. 48
**Replaying a Test in Debug Mode** .................................... 49
**Stopping a Test** ........................................................... 50
Scheduling Tests ................................................. 50

Chapter 6  Displaying Test Results
Displaying Script Performance ............................. 53
Connections Window .......................................... 55
Counters Window ............................................. 56
  Add Windows Counter ..................................... 56
  Add Xen Counter .......................................... 56
  Delete Counter ........................................... 57
Load Control Rule Window ................................. 57
  Delete Load Control Rule ................................. 58
Alarms Window .................................................. 58
  Add Alarm .................................................. 58
  Delete Alarm ............................................. 59
Measurement Window ....................................... 59
Monitor Window .............................................. 60
Messages Window ........................................... 60

Chapter 7  Chart Reports

Appendix A  Virtual Keys Reference

Appendix B  Creating an ICA File
Creating ICA Files for Versions Prior to Presentation Server 4.5 ........... 69
Creating ICA Files for Presentation Server 4.5 and Later .................... 69
Creating ICA Files from the APPSRV.ini File ................................ 70

Appendix C  Script Example
  Close Existing Doc Folder .................................. 74
  Word Task Folder ......................................... 75
  Exit App Folder ........................................... 76

Appendix D  Intelligent Load Control Example
Sample Load Control Rule .................................... 77
Sample Chart Report ............................................ 79
Load Control Rule Runtime Activation Messages .......................... 80
Welcome

EdgeSight for Load Testing is an automated load and performance testing solution for Citrix XenApp and XenDesktop environments. The product extends the application performance visibility that these environments provide by introducing pre-production application performance tools. The following figure shows the EdgeSight for Load Testing components.

The Controller is used to record and create virtual user scripts and define tests. When the test is ready for playback, the Controller instructs the Launchers to run the test with a certain number of virtual users for a certain period of time.

The Launchers receive the commands from the Controller and generate virtual user ICA sessions on the target XenApp or XenDesktop systems. The number of Launchers required will vary based on the target virtual user load.

The Launchers then report session information back to the Controller for run-time and post run-time analysis.

This load generating software solution enables administrators to predict how systems will cope with high levels of user load. By simulating hundreds of virtual users and monitoring the responsiveness of the system under test, it allows the administrator to determine how the current configuration and hardware infrastructure will support anticipated demand.
Document Organization

This document contains seven chapters and three appendices:

Chapter 1, this chapter, contains an introduction to the EdgeSight for Load Testing product and a list of related documentation.

Chapter 2, “Initial Configuration,” describes the tasks that must be completed before you can start using the Load Test software. These tasks include configuring servers, controllers, and launchers.

Chapter 3, “Recording a Script,” describes how to record a script and provides suggestions for creating reliable and maintainable scripts.

Chapter 4, “Editing Scripts,” describes how to navigate in the user interface and contains detailed information about the instructions EdgeSight for Load Testing supports.

Chapter 5, “Running a Load Test,” describes how to execute a load test.

Chapter 6, “Displaying Test Results,” describes the different options that you can use to view run-time performance statistics for load tests.

Chapter 7, “Chart Reports,” describes how to create, display, and save historical information for load tests.

Appendix A, “Virtual Keys Reference,” describes the supported virtual keys you use as keyboard input in scripts.

Appendix B, “Creating an ICA File,” describes how to create ICA files for different versions of EdgeSight for Presentation Server software.

Appendix C, “Script Example,” provides an example script that contains a description of some of the more complex editing techniques and use of instruction properties.

Related Documentation

You will find it helpful to have the following documentation available while using this product:

• Citrix EdgeSight for Load Testing Installation Guide
• Citrix EdgeSight for Load Testing Readme
• Citrix XenApp documentation
• Citrix XenDesktop documentation
• Citrix License Server documentation

In addition, online help is available from the Controller.
Chapter 2

Initial Configuration

This chapter describes the minimum configuration steps required to start developing scripts with the EdgeSight for Load Testing software. Additional configuration steps are required when you run load tests and are described in “Running a Load Test” on page 41.

The topics described in this chapter include:

• Configure Servers
• Configure XenDesktop Environment
• Configure Launchers
• Configure Controller

Configure Servers

The servers that load tests will be applied to require the following software and user account settings:

• Presentation Server 3 or later or XenApp must be installed on the servers.
• A user account must be created that allows the test users to log into the system. A single account can be configured so that multiple users log in to the same account simultaneously. See “Setting User Session Limit” on page 8.

Note For a complete list of the hardware and software requirements, see the Citrix EdgeSight for Load Testing Installation Guide.

The following sections describe connection settings that are required on servers.

Ending Sessions Automatically

During load testing, when virtual users disconnect from servers they must not leave remnant disconnected sessions. To ensure this, you must configure all servers to terminate when session limit is reached or the connection is broken:
1. Go to Start Menu > Administrative Tools > Terminal Services Configuration Tool.

2. Click on the Connections folder, and right click on the ICA-tcp connection.

3. In the resulting dialog, select Sessions > Properties.

4. For When session limit is reached or connection is broken, select End session and Override user settings.

Setting User Session Limit

If you are using Windows Server 2003 or later, the Restrict each user to one session setting must be disabled. Disabling this setting allows EdgeSight for Load Testing to use multiple copies of the same user.

1. Go to Start Menu > Administrative Tools > Terminal Services Configuration

2. Click on the Server Settings folder, and right click on Restrict each user to one session.

3. In the resulting dialog, disable Restrict each user to one session.

Setting Published Applications Settings

If you connect directly to a server (without using an ICA file), you must configure the server so that any user can run non-published applications.

1. Go to Start Menu > Administrative Tools > Terminal Services Configuration Tool

2. Click on the Connections folder, and right click on the ICA-tcp connection.

3. Select the ICA Settings tab

4. Disable Non-administrators only launch published applications.

Setting Seamless Logins

Although not a requirement, the recommended setup is to use EdgeSight for Load Testing with seamless user logins. This ensures that virtual users log in without human intervention.

1. Go to Start Menu > Administrative Tools > Terminal Services Configuration Tool

2. Click on the Connections folder, and right click on the ICA-tcp connection.
3. In the resulting dialog, select **Login Settings**
4. Select **Use Client-provided logon information**
5. Unselect **Always prompt for password**.

**Configure XenDesktop Environment**

If you are load testing XenDesktop systems, you should disable client drive mapping using an HDX user policy. When the Auto connect client drives setting is disabled via the Citrix Desktop Studio, the prompt to allow drive mapping access is not displayed and does not interfere with load testing.

**Configure Launchers**

Launchers, the systems that load test users are started from, require the following:

- ICA Client - Version 8.1 or later of the ICA Client software is required to run load tests.
- Ports - Launchers use port 18747 to communicate with the Load Test controller.

The Launcher is installed as a service (Citrix EdgeSight Launcher Service). The service is set to start automatically.

In addition to these required settings, you may want to perform one of the following optional configuration tasks.

**Disabling System Beep**

During load test execution, the system the launcher is installed on beeps each time a network connection is opened or closed.

Use the following command to temporarily disable the beep. When using this method, the beep is enabled when the system is rebooted.

1. Open a Windows command prompt screen. **Start > cmd**
2. Enter the following command: **net stop beep**

You can restart the beep service using the **net start beep** command.

Use the following command to permanently disable the beep:

1. From the Start menu, right click **My Computer > Manage**
2. Expand **System Tools** and select **Device Manager**.
3. From the **View** menu, select **Show hidden devices**.
4. Expand **Non-Plug and Play Drivers**.
5. Right click **Beep**, and select **Properties**.
6. Select the **Drivers** tab.
7. Click **Stop**. You can also change the Start-up type to Disabled so the beep service never starts.

## Configure Controller

This section describes configuration settings required on the system running the Controller prior to recording scripts.

The following ports are used by the Controller:

- Port 18747 must be available for the Controller to communicate with the Launcher
- Port 27000 must be available for the Controller to communicate with the License Server.

## Configure License Server

EdgeSight for Load Testing uses Version 4.5 of the Citrix License Server. See the *Getting Started with Citrix Licensing Guide* for more information.

The License Server Configuration dialog identifies the host and port of the license server. Use the following procedure to identify the License Server for your system:

1. From the Main menu, select **Options > License Server Configuration**
2. In the **Connect to** field, enter the License Server name as an IP address or fully qualified domain name.
3. In the **Connect on port** field, enter the port the Load Test Controller uses to communicate with the License Server. The default port is 27000.

When a test is replayed, the EdgeSight for Load Testing software checks for the presence of a XenApp Platinum or Enterprise license on the specified license server. If a valid license is detected, you can run as many users as required.

If a XenApp Platinum or Enterprise license is not detected, the EdgeSight for Load Testing software checks for the presence of a XenDesktop Platinum or Enterprise license on the specified license server. If a valid license is detected, you can run as many users as required.
If you do not have a XenApp or XenDesktop Platinum or Enterprise license on the specified license server, but have Concurrent User (CCU) licenses, the number of users you can test with is limited by the amount for which you are licensed. When you start a load test, the number of CCU licenses that are checked out corresponds to the number of users you have created. If you create a load of five concurrent users, but created a total of 100 users, then 100 CCU licenses are checked out when you start the test.

If no licenses of any kind are detected, the software displays the message (Could not find a valid license on the designated license server) and offers to run in 15 user evaluation mode.

**Create a Script**

A script must be created as a placeholder for the instructions you record. A script requires a name, the size of the screen that displays test execution back to the Launcher, and an algorithm that defines how load tests are executed across multiple launchers.

When you create a script it contains three components:

- **Load** - defining the duration of a load test, the number of users that will execute the test, and how many users are concurrently executing the test.
- **Connections** - defining how the launchers connect to servers.
- **Instructions** - the discrete components of the test.

These components are organized in the Test Tree as shown in the following figure.

![Test Tree Diagram]

When you save a script, the Load and Connections information is saved with the instructions.

Use the following steps to create a script:

1. From the Menu bar, select **Test > Add Script**
2. In the Script Properties dialog, enter a script name. There are no restrictions on the name that you use.
3. Enter the Client Size. The Client Size defines the size of the screen display for the session between the Launcher and the Server.

*Note* If you record a script in one window resolution and then replay it in another resolution, the script may fail. Changing the screen size may cause mouse instructions to fail due to corresponding changes in X and Y window coordinates.

4. Select the algorithm for the Concurrency Model.

**Concurrency Model**

The concurrency model allows selection from three algorithms that specify the order tests are run when multiple launchers are used in a load test. The following figures illustrate the different concurrency models: Balanced, Rotated, and Top down.

In the Balanced model, the controller directs an equal number of users on each launcher to execute tests.

In the Rotated model, the controller directs all users from Launcher A to run before the users on Launcher B start testing.
In the Top down model, the controller directs the first user in a launcher to execute the test whenever it is available to restart the test. In the following figure, Launcher A / User 1 is available to restart the test when User 2 completes the test. At the same time, Launcher B / User 1 is not available to restart the test until User 3 has completed testing.

Create a Connection
Connections define where users are launched from and the servers and applications to which they connect. A load test may have multiple connections defined. For example, a single load test may replicate a geographically dispersed Presentation Server or XenApp environment and require connections from three different locations.
Different types of connections are supported. Each launcher is limited to operating using a single type of connection during a load test. Connections can be made using the following methods:

- ICA file
- XML Service
- Web Interface
- Server Desktop

**Connecting with an ICA File**

EdgeSight for Load Testing can connect to published applications using ICA files.

You can create ICA files for published applications. The procedure for creating ICA files depends on the version of Presentation Server software you are running. See “Creating an ICA File” on page 63.

Once you have an ICA file, you can use the following procedure to define it as the connection method:

1. Select the Connection node and click the Add a Connection button in the scripts toolbar. The Connection Properties dialog is displayed.
2. In the **Connect From** field, enter the address of the Launcher using an IP address or a fully qualified domain name.
3. Click **ICA file** and the **Browse** button to identify the ICA file. The ICA file identifies the Server and applications to connect to and is entered in the **Connect To** field.
4. To use the same ICA file for all Launchers, click the **Distribute** check box. This causes a copy of the ICA file to be sent to each Launcher.

**Connecting With the XML Service**

The XML service uses a Web Interface Connector to access published resources. Users log on to the Web Interface and see links to the applications that they are authorized to run.

**Note** The Web Interface Connector is an optional feature that must be selected during the EdgeSight for Load Testing installation. If the feature was not installed, you can rerun the installation procedure to install it. To use the Web Interface Connector, you need the Microsoft Visual J# .NET Redistributable Package. See the *Citrix EdgeSight for Load Testing Installation Guide* for information about this requirement.

Use the following steps to connect using the XML service:
1. Select the Connection node and click the Add a Connection button in the scripts toolbar. The Connection Properties dialog is displayed.

2. In the Connect From field, enter the address of the Launcher.

3. Select XML Service and select the Browse button or enter the Server, port, and application in the Server field (server:port:application).

4. The XML Application browser is displayed. Enter the following fields:
   - Server Address and Port to connect to. The default port is 8080.
   - User Name, Password, and Domain of a valid user.
   - Select Search. The system displays available application.
   - Select the application you want to connect to.

Connecting With the Web Interface

The Web Interface allows you to access applications that are available through the XenApp or XenDesktop Web Interface. Before using this connection method, you must install Web Interface support using the EdgeSight Web Interface Support MSI file on the target system that you will be load testing. See the Citrix EdgeSight for Load Testing Installation Guide for more information.

- The installation places the Application Page (for example, Citrix/Accelerator/site/Swiss) on the Web Interface Server of the system you are load testing. You must correctly identify the location of the Application Page when configuring the Web interface.

- The installation creates the EdgeSightWISecurity group on the Web Interface Server. Users of the Controller and Launcher systems must be logged in with a user that is a member of this group. (Alternatively, you can specify Web interface access credentials as described in “Web Interface Access Credentials ” on page 16.)

- The Login Page differs depending on the target system. Depending on your configuration, you may need to change the default.

When you use the Web interface, the following authentication is performed:

1. Test users are authenticated against a list of users on the target system

2. The user logged on to the Controller or Launcher is authenticated, ensuring that they are a member of the EdgeSightWISecurity group. If configured, the Web Interface access credentials are used to authenticate the user.

Use the following steps to configure a connection using the Web Interface Application Browser:
1. Select the Connection node and click the Add a Connection button in the scripts toolbar. The Connection Properties dialog is displayed.

2. In the **Connect From** field, enter the address of the Launcher using an IP address or a fully qualified domain name.

3. Click the **Web Interface** radio button.

4. Click **Browse**. The system displays the Web Interface Application Browser dialog.

5. Enter the server address of the target system that you will load test.

6. Enter the Web interface login page you will be accessing. The default login page differs depending on the target system.

7. Enter the Application Page that contains the list of applications to be accessed by EdgeSight for Load Testing. Note that this page is loaded when the Web Interface Support MSI is run and is required to display applications in a format that is usable by EdgeSight for Load Testing.

8. Enter the Username, Password, and Domain of the valid EdgeSight for Load Testing user that will be logged in to the server during load testing.

9. Click **Search** to display the available applications.

10. Select the application to use and click **Select**.

   If you have incorrectly configured the connection or if the Controller fails to authenticate with the Web Server, an error message will be displayed when you attempt to search for applications.

### Web Interface Access Credentials

In order to successfully access the Web Interface, Controllers and Launchers must log in with an account that is a member of the EdgeSightWISecurity group on the Web Interface Server. If the account you are logged into is not a member of this group, you can use the Web Interface Access Credentials dialog to log in with the correct credentials.

Use the following steps to enter credentials for the Web Interface:

1. In the Connections Properties dialog box, click **Access**

2. Click the **Use the logged in users credentials** checkbox if the account you are logged into is a member of the EdgeSightWISecurity group

   or

   Enter the Username, Password, and Domain of a user that is a member of the EdgeSightWISecurity group.

3. Click **OK**.
Connecting to the Server Desktop

Connecting to the Server desktop allows users to run applications available from the Desktop of the server.

1. Select the Connection node and click the Add a Connection button in the scripts toolbar. The Connection Properties dialog is displayed.

2. In the Connect From field, enter the address of the Launcher

3. Select Server and enter the Server address in the Connect To field.

**Note** When connecting to the Server Desktop, you must configure the server to allow users to run unpublished applications. See “Setting Published Applications Settings” on page 8.

Create Users

Users are created for a specific connection. You must create a connection before creating users. See “Create a Connection” on page 12 for instructions on creating a connection.

To create users, select a connection and then add the users for that connection. In order for users to successfully open connections to servers and execute load tests, they must have accounts and the proper credentials on the servers. See “User Accounts” on page 8 for information about creating accounts on servers.

Use the following steps to create users:

1. Select the Connection for which you want to create users.

2. Select the Add Users to the Selected Connection button from the scripts toolbar.

3. In the Add Users to Connection dialog.
   A. Enter the number of users you want to add.
   B. Enter the number you want users to start at. Users are added as a single account (Tester). By numbering them, you can better identify each user (Tester1, Tester2...) in the Messages display. If a user times out during test execution, the specific user is identified. To enable this feature, you must click the check boxes next to the # character.
   C. Enter the User name for the user.
   D. Enter the Password for the user.
   E. Enter the Domain name for the user.
View and Copy Users

You can view, edit, and copy users in the Main window. To view users, select the connection that the users are associated with. The users are displayed in the Main editing window.

You can copy users using cut and paste operations. If you have numbered the users, the numbers do not get incremented.

The user properties can be edited in the Main Editing window. You can change the user name, password, or the Domain for each user. To make changes to any of these fields, select the text in the field and edit the text. Use the Enter key to complete the changes.
CHAPTER 3

Recording a Script

This chapter describes how to record a test, including starting, stopping, and restarting the recording. It also provides information that will be useful in developing strategies for creating scripts.

The topics described in this chapter include:

• Strategy for Creating Good Scripts
• Starting a Recording
• Stopping a Recording
• Replaying a Recording
• Using Fast Record

Strategy for Creating Good Scripts

In most cases, recording an entire script will be done in small increments. Larger scripts may require that you record parts of the script, debug and edit the script and replay it. When you have parts of the script that run error free, you add additional instructions to the script.

You can start or stop recording and add instructions at any point in the script. You can also replay selected portions of the script.

When you create large scripts, you can replay the scripts using the Fast Record option or JScript features to accelerate the replay.
Some of the practices that provide more reliable scripts include:

- Mouse instructions are usually the easiest way to record a script, but may be unreliable when running the load test. Replacing mouse instructions with keyboard instructions creates a more reliable test.

- Use folders to organize instructions. Group functional parts of the script into different folders. Use repeating and iterating folders to perform loops.

- Use wildcards in Synchronization Point captions. These allow you to identify screens with captions that change names. (For example, Microsoft Word Document 4, Microsoft Word Document 5, etc.)

- Use Search and Match instructions for windows that cannot be identified because the window changes position or the text inside the window changes position.

- Don’t save data to disk unless you require it as part of the test. Continuously saving data can cause storage problems, and saving and reopening files causes multiple dialog boxes to open and require response.

- Use the Messages screen to diagnose problems.

- Use Fast Record to reduce the time required to replay long scripts.

- Use Concurrency Control and Rate Control to avoid creating unrealistic network loads.

- Be aware of failures caused by exceeding timeout periods. Change the timeout periods for specific instructions.

- Use load control rules to automatically adjust test loads or change test formats when certain runtime conditions are met.

### Starting a Recording

Prior to recording a script, you must configure the Servers, Controller and Launchers as described in “Initial Configuration” on page 7.

When you create a recording, a single user must be selected to perform the instructions that are recorded. To select a recording user:

1. In the Test Tree pane, select Connections > connection (the connection that you created).

2. The list of users associated with this connection is displayed.

3. Select a recording user by clicking on the icon at the left of the user.

Start recording the test:
1. In the Test Tree pane, select **Instructions**
2. From the main toolbar, select the Record Test button

When the recording session starts, the following changes are visible:

- The Controller minimizes.
- A new window is created in the upper left hand corner of the screen.
- An ICA seamless host connection is made to the server.
- The Recording User is logged on.
- If you are connecting directly to a Server desktop, an ICA Seamless Host Agent dialog is displayed. You must click OK to continue.

Once you have acknowledged the ICA Seamless Host Agent dialog box, you begin recording instructions. Each mouse click, keyboard entry, and the windows that open during the recording session are recorded.
By starting a recording and then immediately logging out, you create a script that looks similar to the following. This is the simplest script you can create.

### Stopping a Recording

The recording session continues until you stop it, or log out of the ICA session. You stop the recording session using one of the following methods:

- Logging out of the ICA session, as shown in the script in the previous figure stops the recording process. When you exit the script by logging out you preserve all of the instructions that you have recorded in the script.

- From the main toolbar, select the Stop Test button to stop the recording and save the instructions that you have recorded. This may create a script that does not exit when you replay it. When you stop a script without logging out or closing the ICA session and want to record additional instructions, the script will replay to the point that it was stopped and then wait for additional instructions to be added.

- From the main toolbar, select the Cancel Test button to stop the recording and discard the instructions that have been recorded.

You can also use the Test > Stop and Test > Cancel menu items to stop the test.
Replaying a Recording
While you develop a script you may start, insert new instructions, stop, and replay
the script multiple times. There are number of different ways to replay a script.
Depending on the task you are trying to complete, use the following replay
methods:

- To replay the entire recording, select Instructions in the Test Tree pane (the
top level folder that you recorded the instructions under) and click the
Replay Test button in the main toolbar.

  The recording replays all of the instructions that you previously recorded
  and stops at the last instruction. If the last instruction logs out of the ICA
  session, the recording session ends.

- To replay a recording so that it stops at a certain instruction, allowing you to
  record new instructions into the script, select the last instruction you want
to execute and start the recording. The recording session plays back all of
  the existing instructions and stops at the selected instruction. At this point,
you can add instructions to the script.

Using Fast Record
Fast record allows you to replay a script at more than twice the speed it normally
takes to execute. This setting has no effect on the speed while you are running a
load test. This is helpful when you are constantly replaying a script during editing
or debugging.

Use the following steps to set the recording speed:
1. From the Main Menu, select Options > Test Configuration.
2. In the Test Configuration dialog, select the Use fast record check box.
3. Click OK.
This chapter provides information that is useful when editing scripts. Editing scripts allows you to change or tune the instructions that have been recorded. You can modify the properties of instructions, convert mouse instructions to keyboard instructions, add and delete instructions, and modify Synchronization Points using match and search features in the script.

In most cases, the script development process requires repeated recording and editing sessions to create an acceptable script.

The topics covered in this chapter include:

- Navigation
- Folders
- Synchronization points
- Mouse input
- Keyboard input
- Variables
- JScript
- Comments
- Idle time

**Introduction to Script Editing**

After you record a script, you can customize the script by changing properties that control how each of the instructions of the script function. You can also add instructions that are not created during the recording session. Instructions that you add could include folders, variables, comments, JScript instructions, and idle time.
You can edit scripts in the Test Tree pane (left hand side), or in the Script Editing screen (right hand side). The following figure shows how scripts are displayed in each of these screens:

Each instruction type is identified by an icon:

- The cog icon represents a Synchronization Point. Synchronization Points are instructions that cause a virtual user to wait for windows to appear or be in a predefined state before continuing. There are a number of settable properties for a synchronization point.

- The hourglass icon represents idle time in the script. The idle time is recorded when you create a script. Idle time can be changed and you can add idle time between instructions in a script.

- Folders can be used to organize multiple instructions. You can configure folders to repeat the instructions in them, to execute the instructions conditionally, or to not execute the instructions stored in the folder. There are a number of conditional folders:
  - Iterating folders execute the instructions in the folder repeatedly until the controller determines that the user should log off.
  - Repeating folders repeat the instructions in the folder based on a number of repetitions specified in the folder properties.
  - If Satisfied folders execute based on the success of a previous instruction.
  - If Not Satisfied folders executes based on the failure of a previous instruction.
Do Until Satisfied folders execute the contents of a folder until a designated instruction succeeds. The number of times to retry the folder is specified in the folder properties.

Do Until Not Satisfied folders execute the contents of a folder until it fails. The maximum number of times to retry the instruction is specified in the folder properties.

Do Not Execute folder never executes the instructions in the folder.

The keyboard icon represents keyboard instructions or a variable. Keyboard instructions include data entered into applications and keyboard commands.

The mouse icon represents mouse input. The input includes left and right clicks and mouse move operations.

The scriptlet icon represents an instruction that contains Microsoft JScript. The scriptlet allows you to execute custom JScript in a script.

Each instruction has settable properties. When you select an instruction in the Test Tree pane, the instruction properties are displayed and editable in the Main window. The following figure shows the properties for a Synchronization Point instruction.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Point</td>
<td>10</td>
</tr>
<tr>
<td>Type</td>
<td>Exists</td>
</tr>
<tr>
<td>Caption</td>
<td>Program Manager</td>
</tr>
<tr>
<td>Style</td>
<td>Any</td>
</tr>
<tr>
<td>Bitmap Code</td>
<td></td>
</tr>
<tr>
<td>ROI</td>
<td>232,354</td>
</tr>
<tr>
<td>Window ID</td>
<td>65632</td>
</tr>
<tr>
<td>Fail Mode</td>
<td>Logout</td>
</tr>
<tr>
<td>Timeout</td>
<td>30</td>
</tr>
<tr>
<td>Conditional Folder</td>
<td>None</td>
</tr>
<tr>
<td>Max Repeat</td>
<td>3</td>
</tr>
</tbody>
</table>

**Navigation**

Instructions are edited in the Test Tree pane and in the Main Window’s editing pane. When you select the top level folder of instructions, all instructions are displayed in both the Test Tree and the Main Window.
In both the Test Tree pane and the Instruction Editing window, you can add, delete, and move instructions. You can also create folders and organize instructions into the folders.

When you select (click) an instruction in the Test Tree pane, the instructions properties are displayed in the Main Window.

You can change any of the settable properties in the Main Window. Select the property you want to edit and either select an option from a drop-down menu or edit the value with the keyboard.

**Adding Instructions**

There are a number of ways to add instructions to an existing script.

- Right-click the instruction you want to add an instruction above. From the drop-down menu, select the instruction you want to add.
- Select the instruction that you want to add the instruction above, then select the instruction type to add from the toolbar:

```
<table>
<thead>
<tr>
<th>Type</th>
<th>Left Click</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen X Co-ordinate</td>
<td>176</td>
</tr>
<tr>
<td>Screen Y Co-ordinate</td>
<td>198</td>
</tr>
<tr>
<td>Window ID</td>
<td>65566</td>
</tr>
<tr>
<td>Window Input</td>
<td>Relative Top Left</td>
</tr>
<tr>
<td>Relative X Co-ordinate</td>
<td>176</td>
</tr>
<tr>
<td>Relative Y Co-ordinate</td>
<td>198</td>
</tr>
</tbody>
</table>
```

- In the Test Tree, select the top level folder of the script you are editing. Add instructions using one of the methods described previously. Instructions are added to the end of the script.
- In the Test Tree, select the folder you want to add an instruction into. Add an instruction to the bottom of any instructions in the folder using the menu or the toolbar. See “Moving Instructions” and “Using Folders” for more information on populating folders.

**Moving Instructions**

You may be required to move instructions after you have completed recording a script or when editing of the script. The majority of the time instructions are moved into folders.
In the Test Tree, you can drag and drop or cut and paste single instructions to perform moves within the Test Tree.

In the Main Window, you can select multiple instructions and drag and drop or cut and paste them into the Test Tree or within the Main Window.

- Each move operation results in the instructions being placed above the target instruction.
- When you move instructions into a folder, single instructions are added at the bottom of the folder, while multiple instructions preserve their order.
- Pasting an instruction on top of an instruction already in the folder places the new instruction above the original instruction.

Using Folders

Folders can be used to organize instructions, or to use the repeating or conditional features of the folder.

Use the following procedure to create and populate a folder:

1. In the Main Window, select the instruction that you want to create the folder above.
2. From the script toolbar, click the Add an Instructions Folder button.
3. Enter a name for the folder.
4. In the Main Window, select all of the instructions you want to move into the folder.
5. Drag and drop the instruction into the folder in the Test Tree pane.

Alternately, you can create a folder and move instructions into it in the Test Tree pane:

1. In the Main Window, select the instruction that you want to create the folder above.
2. From the script toolbar, click the Add an Instructions Folder button.
3. Enter a name for the folder.
4. In the Test Tree pane, drag and drop the instructions, one at a time, into the folder.
   A. Dropping an instruction on top of the Folder places the instruction at the bottom of the folder
   B. Dropping an instruction on top of an instruction already in the folder places the new instruction above the original instruction.
Note You can create folders in the Test Tree pane, but a folder created on top of an existing folder is created as a sub-folder. When you create a folder on top of a folder in the Main Window, it is created above the existing folder.

Repeating Folders
In a repeating folder, the instructions in the folder execute until the specified repetitions are completed. If the execution time is longer than the time specified in the Load properties, the test continues until complete, overriding the time specified by the load. See “Creating a Load” on page 45.

If the number of repetitions impacts the test duration specified by the load, an option to create a repeating test may be to use an Iterating Folder. See “Iterating Folders” on page 30.

To configure a repeating folder:
1. Right click on the folder and select **Folder Properties**. The Folder Properties dialog is displayed.
2. From the **Execute** drop-down menu, select **Repeat**.
3. In the Max Repeat dialog, enter the number of times you want the folder to repeat.
4. Click **OK**

Iterating Folders
Iterating folders are also repeating folders that continuously repeat the folder instructions until the Controller determines that the virtual user should log out.

Concurrency Control must be used with Iterating Folders to guarantee a certain numbers of users repeat the folder. Without setting concurrency control, the Controller will not execute the instructions contained in the folder.

Note Rate Control is not required but is helpful for iterating tests. With rate control you can eliminate performance spikes caused by a large number of users starting testing at the same time. In addition, if multiple users fail or time out during the test, Rate Control prevents them from restarting tests simultaneously.

Using iterating folders ensures that the script tests the application rather than the ability of the server to service connections.

Use the following steps to create an Iterating folder:
1. Right click on the folder, select **Folder Properties**. The Folder Properties dialog is displayed.
2. In the **Execute** drop-down menu, select **Iterate**.
3. Click **OK**
Set the Concurrency Control and the Rate Control for the folder using the following steps:

1. In the Test Tree pane, select **Load**.
2. In the Main Window, double-click the load that you want to modify.
3. In the Load Properties dialog, check the **Concurrency Control** box.
4. Enter the number of concurrent users you want running the test at start time and at completion time.
5. Check the **Rate Control** box and enter entry and exit rates.

See “Creating a Load” on page 45 for information about Rate Control and Concurrency Control.

**Conditional Folders**

Conditional folders execute folders based on a previous instruction passing or failing execution. There are a number of different conditional folders:

- **If satisfied** - execute the instructions in the folder if the specified instruction completed successfully.
- **If not satisfied** - execute the instructions in the folder if the specified instruction failed.
- **Do until satisfied** - execute the specified instruction until it is successful. The number of times to retry the execution is specified by the number specified in the Max Repeat field.
- **Do until not satisfied** - execute the specified instruction until it fails. The number of times to retry the execution is specified by the number specified in the Max Repeat field.

You can configure a conditional folder so that when it fails, execution of the script continues or the script exits with an error message.

**Do Until Satisfied Example**

Occasionally, when you click on OK in an ICA Seamless Host Agent dialog box to acknowledge the start of the session, the click is not recognized. An additional click must be entered to continue with the script.

To make certain that the mouse click is recognized, you can use a Do Until Satisfied conditional folder and enter the mouse click multiple times. The following steps describe how you configure the folder:
1. Create a folder above the first instruction in the script:

![Folder and Instruction Image]

2. Move the first three instructions into the new folder.
3. Modify the Folder Properties, selecting Do Until Satisfied for the Execute property. In the Max Repeat field, enter 3.

When the script starts, the instructions in the folder are repeated until the mouse click is recognized.

Creating Conditional Folders

To create a conditional folder, perform the following steps:

1. Create a folder (see “Using Folders” on page 29).
2. Right-click the new folder and select **Folder Properties**.
3. In the Folder Properties dialog:
   A. From the **Execute** menu, select the type of condition to create
   B. In the Synch No. field enter the number of the instruction that you are basing the condition on.
   C. In the Max Repeat field, enter the number of times to repeat the instruction before the loop fails (for Do Until Satisfied and Do Until Not Satisfied conditions).

Folder Properties

Folder properties define how the instructions in a folder are executed. There are three properties that can be set when configuring a folder:

- **Execute** - defines the type of operation the folder executes.
- **Synch No** - identifies an instruction that a conditional folder is based on.
- **Max Repeat** - defines how many repetitions are performed.
The following table describes the Execute properties and notes which require Synch No. and Max Repeat properties.

<table>
<thead>
<tr>
<th>Execute</th>
<th>Description</th>
<th>Requires...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>Execute the instructions in the folder</td>
<td>N/A</td>
</tr>
<tr>
<td>Iterate</td>
<td>Instructions in the folder are repeated continuously during the load test. The controller determines when the instructions exit the loop and log out of the test.Concurrency control must be used with iterating folders.</td>
<td>N/A</td>
</tr>
<tr>
<td>Repeat</td>
<td>Instructions in the folder are repeated the number of times specified in Max Repeat.</td>
<td>Max Repeat</td>
</tr>
<tr>
<td>If Satisfied</td>
<td>Instructions in the folder are executed if the Synchronization Point is satisfied.</td>
<td>Synch No</td>
</tr>
<tr>
<td>If Not Satisfied</td>
<td>Instructions in the folder are executed if the Synchronization Point fails.</td>
<td>Synch No</td>
</tr>
<tr>
<td>Do Until Satisfied</td>
<td>The instructions will be repeatedly run until the Max Repeat value is reached or the Synchronization Point is satisfied.</td>
<td>Synch No, Max Repeat</td>
</tr>
<tr>
<td>Do Until Not Satisfied</td>
<td>The instructions will be repeatedly run until the Max Repeat value is reached or the identified Synchronization Point fails.</td>
<td>Synch No, Max Repeat</td>
</tr>
<tr>
<td>Never</td>
<td>The instructions in the folder are not executed. This is useful for temporarily suspending execution when creating or debugging a script.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Synchronization Points**

Synchronization Point instructions cause virtual users to wait for an application window to be in a defined state. Once correctly defined, instructions are executed in the window. When the specified event does not occur in the specified timeout period, the instruction fails and a timeout error is generated.
The following tables provide a description of the Synchronization Point properties. The Type properties identify how the Synchronization Point is identified before input is applied. For Match and Search synchronization points, you must specify bitmap recording options. You can also specify the suppression of mouse input. Mouse input can alter the graphical state of a screen, making it difficult to match a screen graphically.

<table>
<thead>
<tr>
<th>Type Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Input synchronization is based on the screen existing. This is the default.</td>
</tr>
<tr>
<td>Does not exist</td>
<td>Input synchronization is based on the screen not existing.</td>
</tr>
<tr>
<td>Foreground</td>
<td>Waits for the window to be in the foreground.</td>
</tr>
<tr>
<td>Matches</td>
<td>A bitmap recorded synchronization point. The bitmap image is used to match the Synchronization Point that is always in the same position.</td>
</tr>
<tr>
<td>Does not match</td>
<td>Waits for the specified window's graphics around a point to not match the specified bitmap code.</td>
</tr>
<tr>
<td>Changed</td>
<td>Waits for the specified window's graphics to have changed since the last Synchronization Point. If a Bitmap code is specified then this instruction waits for a graphics change in the 10x10 pixel area in the window specified by ROI. If the ROI field is left empty then this instruction waits for any change in graphics in the entire window.</td>
</tr>
<tr>
<td>Search</td>
<td>A bitmap recorded synchronization point. During execution, a scan of the whole window is made to find the bitmap.</td>
</tr>
</tbody>
</table>

Style properties specify the style of the window to be matched. These properties are useful for differentiating between two windows with the same caption.

<table>
<thead>
<tr>
<th>Style Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Any style window or window property is accepted.</td>
</tr>
<tr>
<td>Menu</td>
<td>A window menu.</td>
</tr>
<tr>
<td>Dialog</td>
<td>A dialog box.</td>
</tr>
<tr>
<td>Child</td>
<td>The child window.</td>
</tr>
<tr>
<td>Maximized</td>
<td>A maximized window.</td>
</tr>
<tr>
<td>Minimized</td>
<td>A minimized window.</td>
</tr>
<tr>
<td>Significant</td>
<td>The window is of significant size.</td>
</tr>
<tr>
<td>None</td>
<td>No window style is specified.</td>
</tr>
</tbody>
</table>
The following table describes the remaining Synchronization Point properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>The window caption that virtual users wait for before proceeding with the script. You can edit this field. The caption appears when you view the script in the Test Tree pane and the script editing screen.</td>
</tr>
<tr>
<td>Bitmap Code</td>
<td>For search and match synchronization points, a code derived from a 10 x10 bitmap screen capture.</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest. Used with match synchronization. Contains the point in the window at which the match bitmap was captured.</td>
</tr>
<tr>
<td>Window ID</td>
<td>The ID of the window. Used by mouse instructions to deliver input to the correct window.</td>
</tr>
<tr>
<td>Fail Mode</td>
<td>Describes whether instruction execution continues when this instruction fails.</td>
</tr>
<tr>
<td>Timeout period</td>
<td>The default is 30. The default can be changed in the Test Configuration dialog (Options &gt; Test Configuration).</td>
</tr>
<tr>
<td>Conditional folder</td>
<td>The type of conditional execution the Synchronization Point performs. (See “Conditional Folders” on page 31)</td>
</tr>
<tr>
<td>Max repeat</td>
<td>For Do Until Satisfied and Do Until Not Satisfied conditional Synchronization Points.</td>
</tr>
</tbody>
</table>

**Match Synchronization Points**

Match synchronization points cause virtual users to wait for windows to be in a particular graphical state. When recording a match type synchronization point, a 10 x 10 pixel graphic image is captured in the area where the mouse clicks are applied. When virtual users execute a match type synchronization point, they compare the 10 x 10 pixel graphic image with the recorded image. The comparison is done at the same point, the ROI. If no match is found, the instruction fails.

Normally, you would use a Match Synchronization Point to replace a Synchronization Point that requires better detection of when the window is in the correct state for the script to proceed. Match Synchronization Points are useful for ensuring that a window is fully rendered on the screen before proceeding.

If you need to check for a graphical image that is not always in the same point in a window, use a Search Synchronization Point instead of a Match Synchronization Point. (See “Search Synchronization Points” on page 36.)

To create a match synchronization point, use the following steps:

1. From the Main Menu, select **Options > Test Configuration**.
2. In the Test Configuration dialog, select the **Match** and **Suppress Mouse Move** checkboxes.

3. Click **OK**.

4. Restart the script, stopping it at the point where you want to add the Match Synchronization Point.

5. Add the Match Synchronization Point.

6. Stop the script.

7. Display the properties for the Match Synchronization Point you just created.

8. Change the **Type** property to **Matches**

9. Replay the script, the Synchronization Point instruction should be successful.

**Note** You should reset the Test Configuration options if you intend to add additional Synchronization Points to your script. Failure to do so may cause problems with mouse input created while the Suppress Mouse Move option is set.

### Search Synchronization Points

Search Synchronization Points cause virtual users to wait for windows to be in a particular graphical state. When recording a Search Synchronization Point, a 10 x 10 pixel graphic image is captured in the area where mouse clicks are applied. When virtual users execute a Search Synchronization Point, they compare the 10 x 10 pixel graphic image to the entire window. If no match is found, the instruction fails.

Search Synchronization Points are CPU and disk intensive and should only be used when required.

To create a search synchronization point, use the following steps:

1. From the Main Menu, select **Options > Test Configuration**.

2. In the Test Configuration dialog, select the **Search** and **Suppress Mouse Move** checkboxes.

3. Click **OK**.

4. Restart the script, stopping it at the point where you want to add the Search Synchronization Point.

5. Add the Search Synchronization Point.

6. Stop the script.

7. Display the properties for the Search Synchronization Point you created.

8. Change the **Type** property to **Search**.
Using Search Synchronization Points is a good way of finding GUI items that are prone to move, such as icons on desktops or menu items that are subject to changing position in the menu. They can also be combined with mouse instructions as a method of targeting mouse input to a particular graphic.

To use targeted mouse input, use the following steps:

1. Create a Search Synchronization Point as described previously.
2. Edit the Window Input properties of the mouse instruction for the Search Synchronization Point. Set the properties to On Search Point.

This directs the mouse input to the graphic that was searched for. If the window or menu moves, the mouse input moves to where the graphic was found during the search.

The mouse Window Input properties can also be set to On Relative Search Point for Search Synchronization Points. This allows you to apply input to an area relative to, or offset from the actual search point. This is useful when the search point is a menu item that requires input to an area not directly on the searched graphic. Creating a relative search point requires that you manually configure the Relative X Coordinate and the Relative Y Coordinate properties.

### Editing Keyboard Input

Keyboard instructions include text, commands, and references to variables. The variable can be a text variable or a variable defined in a scriptlet instruction. Variables are described in “Creating Variables” on page 41 and “Using Microsoft JScript” on page 42.

Keyboard input is defined by the Type, Modifier, and the keys that are entered. The following tables describe the keyboard properties.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keys</td>
<td>Sends one or more characters. That can be a single character, multiple characters, or a virtual key.</td>
</tr>
<tr>
<td>Key downa</td>
<td>Sends a key down. Can only be a single character.</td>
</tr>
<tr>
<td>Key up</td>
<td>Sends a key up. Can only be a single character.</td>
</tr>
<tr>
<td>Variable</td>
<td>Contains a variable name.</td>
</tr>
</tbody>
</table>

*The Key down and Key up types are used in multiple-key input, such as CTRL+C*
Key modifiers include the following:

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>The Ctrl key is pressed.</td>
</tr>
<tr>
<td>Alt</td>
<td>The Alt key is pressed.</td>
</tr>
<tr>
<td>Ext</td>
<td>The Ext key is pressed.</td>
</tr>
<tr>
<td>None</td>
<td>No key modifier is applied.</td>
</tr>
</tbody>
</table>

The Keys field describes the input to the script:

<table>
<thead>
<tr>
<th>Keys</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>The text is entered into the window and represents text, virtual keys(^1), or a combination of virtual keys and text.</td>
</tr>
<tr>
<td>Variable</td>
<td>A variable name is entered.</td>
</tr>
</tbody>
</table>

\(^1\)The list of virtual keys is provided in Appendix A, “Virtual Keys Reference”.

**Keyboard Examples**

The following examples demonstrate use of keyboard input as commands in a script. These examples are of multiple key and single key commands.

Use the following keyboard input sequence to enter a Alt+F keyboard command (the hot key to the file menu in an application):

<table>
<thead>
<tr>
<th>Keys</th>
<th>Type</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>[VK_MENU]</td>
<td>Keys</td>
<td>None</td>
</tr>
<tr>
<td>f</td>
<td>Keys</td>
<td>None</td>
</tr>
</tbody>
</table>

For a CTRL+Alt+Del command entered using only virtual keys, use the following input:

<table>
<thead>
<tr>
<th>Keys</th>
<th>Type</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>[VK_CTRL]</td>
<td>Key down</td>
<td>None</td>
</tr>
<tr>
<td>[VK_ALT]</td>
<td>Key down</td>
<td>None</td>
</tr>
</tbody>
</table>
You can use the Microsoft Windows key in keyboard sequences. For example, to use the Windows key to open the Windows Run dialog box and start a WordPad session, use the following keyboard input:

<table>
<thead>
<tr>
<th>Keys</th>
<th>Type</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>VK_LWIN</td>
<td>Key down</td>
<td>None</td>
</tr>
<tr>
<td>R</td>
<td>Keys</td>
<td>None</td>
</tr>
<tr>
<td>VK_LWIN</td>
<td>Key up</td>
<td>None</td>
</tr>
<tr>
<td>wordpad</td>
<td>Keys</td>
<td>None</td>
</tr>
<tr>
<td>VK_RETURN</td>
<td>Keys</td>
<td>None</td>
</tr>
</tbody>
</table>

For many key commands, there is more than one way to enter the command. For example, to exit a WordPad session using keyboard commands, you would enter: Alt+F to access the File menu and the x key to exit the session. You can enter these commands using either of the two methods listed in the following table:

<table>
<thead>
<tr>
<th>Keys</th>
<th>Type</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>VK_ALT</td>
<td>Key down</td>
<td>None</td>
</tr>
<tr>
<td>f</td>
<td>Keys</td>
<td>None</td>
</tr>
<tr>
<td>VK_ALT</td>
<td>Key up</td>
<td>None</td>
</tr>
<tr>
<td>x</td>
<td>Keys</td>
<td>None</td>
</tr>
<tr>
<td>f</td>
<td>Keys</td>
<td>Alt</td>
</tr>
<tr>
<td>x</td>
<td>Keys</td>
<td>None</td>
</tr>
</tbody>
</table>

**Editing Mouse Input**

The Mouse Instructions send mouse input to screen or window coordinates. Each instruction contains a type of mouse action, coordinates, and the ID of the window that the operation is taking place in.
Mouse instructions make recording scripts easy, but can be an unreliable input method when executing a test. Mouse instructions fail when windows or menus are repositioned, causing the mouse coordinates to be invalid. For example, if you create a script that opens a window by clicking on a Desktop icon, that icon may move when a file is added to the Desktop. If the icon moves, the mouse instruction fails because it is no longer clicking on a valid coordinate. The same is true if a menu item moves or if the position of a dialog box changes in an application.

In many cases, the instructions you created with mouse input can be replaced with keyboard input. For example the File, Exit mouse clicks can be replaced with the Alt, f, and x keys.

Mouse properties include the following:

<table>
<thead>
<tr>
<th>Mouse Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of mouse operation and can include left click, left down, left up, left double click, right click, right down, right up, right double click, middle click, middle down, middle up, and middle double click. (Mouse up and down keys can be used in combination to simulate drag and drop)</td>
</tr>
<tr>
<td>Screen X coordinate</td>
<td>The X coordinate relative to the top left of the screen. Used when Window input is Relative top left.</td>
</tr>
<tr>
<td>Screen Y coordinate</td>
<td>The Y coordinate relative to the top left of the screen. Used when Window input is Relative top left.</td>
</tr>
<tr>
<td>Window ID</td>
<td>The ID of the window that the mouse input is sent to.</td>
</tr>
<tr>
<td>Window input</td>
<td>Identifies the method of window input: Relative top left - input is delivered to the X and Y coordinates relative to the top left of the window. If the window ID is 0, the screen coordinates are used. On search point - the mouse input is delivered to the coordinates where the graphic was found. Relative search point - the mouse input is delivered to coordinates relative to the point at which the graphic was found.</td>
</tr>
<tr>
<td>Relative X coordinate</td>
<td>The X coordinate relative to the window Used by relative top left and relative search types.</td>
</tr>
<tr>
<td>Relative Y coordinate</td>
<td>The Y coordinate relative to the window. Used by relative top left and relative search types.</td>
</tr>
</tbody>
</table>

a. When you drag the mouse you normally use key down and Window coordinates and then key up and Screen coordinates.

b. Used to enter text into a text field at coordinates relative to a search point.
Creating Variables

By default, all virtual users enter the same data throughout test execution. To create a more realistic test, you can use variables that allow each user to enter unique text into applications.

A User Variable is a data value specific to an individual virtual user that can be accessed from within the script. Multiple variables can be defined for each script so that simulated users can have access to an unlimited amount of ‘individual’ data.

Use the following steps to create a variable:

1. In the Test Tree pane, select Connections > connection to display the list of users for your test.
2. Right-click a user and select Add Variable. The system displays the Add Variable dialog box.
3. Enter a Variable Name
4. Enter the variable text

In the Main Window, the variable is shown being applied at all users:

<table>
<thead>
<tr>
<th>No.</th>
<th>Username</th>
<th>Password</th>
<th>Domain</th>
<th>VarTxt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>tester1</td>
<td>********</td>
<td>mydomain</td>
<td>Add unique text</td>
</tr>
<tr>
<td>2</td>
<td>tester2</td>
<td>********</td>
<td>mydomain</td>
<td>Add unique text</td>
</tr>
<tr>
<td>3</td>
<td>tester3</td>
<td>********</td>
<td>mydomain</td>
<td>Add unique text</td>
</tr>
<tr>
<td>4</td>
<td>tester4</td>
<td>********</td>
<td>mydomain</td>
<td>Add unique text</td>
</tr>
<tr>
<td>5</td>
<td>tester5</td>
<td>********</td>
<td>mydomain</td>
<td>Add unique text</td>
</tr>
</tbody>
</table>

In this example, VarTxt is the variable name and Add unique text is the variable. You must edit each user’s variable to create unique text:

1. Double-click the variable text you want to change.
2. Enter text and then press the Enter key to exit the editing session.

To use the variable in a script, add a keyboard instruction, setting the Type property to Variable and entering the variable name into the Keys field:

**Note** You cannot enter a return key into the variable text you are editing. In addition, if you cut and paste text into the Variable text box, return keys (multiple paragraphs) are not recognized. You can separate paragraphs using the [VK_RETURN] virtual key. For more information about virtual keys, see “Virtual Keys Reference” on page 63.

To delete a variable:

1. Select the variable name in the Main Window.
2. Right-click on the selected text.
3. From the options dialog box, select Delete Variable.

**Importing Variables**

You can import variables using a variable file. A variable file can be a comma-separated text file or a CSV file and include multiple variables. The first line contains variable names and each subsequent line contains the variable values. The first line of variable values is applied to user number 1.

Variable names can contain a-z, A-Z, 0-9, and _ (underscore).

Variable values can contain any characters except for a comma (,). Commas can only be used as the field separator. An imported file with N/A as a variable value preserves existing values.

```
Variable1, Variable2, Variable3
name1, address1, phone1
name2, address2, phone2
name3, address3, phone3
name4, address4, phone4
```

When you import variables, the variable names and the variable values are created for users. The number of users that variables are created for is limited by the number of lines in the file.

If the variable file contains less lines than there are users defined, variables are created for all users, but the variables created for the additional users contain empty value fields.

**Note** When you import variables, if the a variable with the same name exists, the variable data is overwritten by the imported file.

**Using Microsoft JScript**

Scriptlet Instructions allow you to add snippets of Microsoft JScript programming language into scripts. You can then leverage variables within the scriptlet instructions in subsequent keyboard input instructions. In the Test Tree pane, the instructions are implemented as follows:

```
22 // MyVariable=Math.round(Math.random() *45);
23 MyVariable
```
Scriptlets can be used to create variables, such as random numbers, that are entered by virtual users as part of a Keyboard Instruction. They can also be used to control the execution of scripts when conditional Synchronization Points and folders do not provide adequate coverage. For the scriptlet in the previous figure, the instruction properties are displayed in the following figures.

**Note**  Invalid scriptlet instructions may cause scripts to execute incorrectly.

The keyboard Type is set to Variable and the name of the Variable is entered in the keys field.

The Scriptlet instruction properties contain the variable name and the JScript code.

```
Scriptlet | 22
JScript   | // MyVariable=Math.round(Math.random() *45);
```

**Supported Built-in JScript**

Available methods to call from Scriptlets include the following:

<table>
<thead>
<tr>
<th>Scriptlet</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User.IsSatisfied(number instruction);</td>
<td>Returns true if the instruction was satisfied.</td>
</tr>
<tr>
<td>User.GetVariable(string variable name);</td>
<td>Returns the virtual users value for the specified variable name.</td>
</tr>
<tr>
<td>User.SetSpeed(number speed multiplier);</td>
<td>Multiplies the virtual users idle time and typing speed by the supplied value. For example, if speedMultiplier is set to 0.1 then the idle times will be one tenth of their original value.</td>
</tr>
<tr>
<td>User.GetRunCount();</td>
<td>Returns the number of times a virtual user has executed its script during a test.</td>
</tr>
<tr>
<td>User.Logout(Boolean error);</td>
<td>Forces a virtual user to logout. If error is set to true, then an error is generated with the error message supplied.</td>
</tr>
</tbody>
</table>
**User.SetSpeed() Example**

To speed up the execution of the script, or certain parts of the script, use the User.SetSpeed() scriptlet as shown following sample:

```
// User.SetSpeed (0.1);
Text entered in Wordpad
// User.SetSpeed (1.0);
```

- The User.SetSpeed(0.1) is used to set the input speed to ten times the default speed.
- Text is entered in the WordPad window at the increased speed.
- The User.SetSpeed(1.0) scriptlet is called to return the speed to the default value.
Running a Load Test

This chapter describes the steps required to start a load test. The topics described in this chapter include:

- Creating a Load
- Starting a Test
- Stopping a Test
- Scheduling Tests

Creating a Load

The load defines the duration of the test and how many virtual users are running at the beginning of the test and at the end of the test. Optionally, you can specify the rate that users enter and exit the test and specify the number of users running when the test starts and the number of users running when the test completes.

Without using Rate or Concurrency options, users are added to the test in a linear fashion. That is, for a test running one hour with 100 users, you can expect 50 users to be running after one half hour.

You can create multiple loads for one test. When multiple loads are used, the test executes one load after the other. This allows you to run tests with different amounts of users. For example, if you wanted to simulate an environment that changes over a 24 hour period, where the users may change from 100 to 40 and then to 20, you can create a load similar to the following:

<table>
<thead>
<tr>
<th>Duration (min)</th>
<th>Start Rate</th>
<th>End Rate</th>
<th>Start Users</th>
<th>End Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>480</td>
<td>60</td>
<td>60</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>480</td>
<td>60</td>
<td>60</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>480</td>
<td>180</td>
<td>180</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>
Concurrency Control

When the Concurrency check box is selected the system attempts to maintain a count of executing virtual users as specified by the At Start and At End fields. The target user count is ramped between the start and end values over the course of the load, in a linear fashion.

Rate Control

When the Rate check box is selected the system attempts to connect new virtual users at the rate specified by the Connection Interval fields for the load. This rate value represents the time (in seconds) to wait before connecting a new virtual user. Rate control reduces the network load created when multiple simultaneous connections to the server are made.

Rate Control determines not only how new users enter the test, but also control the rate that users attempt to reenter the test because of failure or timeout.

Add a Load

Use the following steps to configure the Load:

1. In the Test Tree pane, select Load and click the Add a Load to the Script button in the script toolbar. The Load Properties dialog box is displayed.

2. Enter the number of minutes that you want the test to run.

3. Click Concurrent if you want the users to run concurrently. If you do not enable this option, only a single user will execute the test at a time.

   • Enter the number of users you want running when the test starts.
   • Enter the number of user you want running when the test completes.

4. Click Rate and enter a rate at which you want the users to connect.

   • Enter the rate for users to enter in the beginning of the test.
   • Enter the rate for users to enter at the end of the test.

Adding Load Control Rules

You can enhance your script and implement intelligent load control by adding load control rules. A load control rule allows you to automatically adjust test loads or change test formats when certain runtime conditions are met. Each rule is associated with a specific part of a test, which must have an associated measurable value. For example, you can create a load control rule for a script, folder, instruction, or synchronization point. You can create multiple rules for a single entity.
For a sample load control rule and the resulting load test data, see Appendix D, “Intelligent Load Control Example.”

Each rule includes the following types of information:

- **Name**—A name for the rule.
- **Condition**—The condition which triggers the application of the rule, and the condition which causes the rule to be reset. For example, a rule can be triggered when the login failure rate exceeds 15 per minute, and then reset when the failure rate falls below 5 per minute. Both the triggering and resetting of the rule can be dependent on the condition occurring for a specified amount of time.
- **Effect**—The actions to be taken when the rule is triggered. Actions include stopping the creation of new users until a specified amount of time has passed, until the user count drops to a specified level, or until all active rules have been reset. The action can also be to terminate the test.
- **Affected script(s)**—One or more scripts to which the rule is applied and the resulting action taken.
- **Additional parameters**—You can specify a lag time from the start of the test to when the rule conditions are tested. You can also set a limit for the number of times that the rule can be triggered during a test.

To create a load control rule:

1. In the Test Tree pane, select **Display**.
2. Select the **Scripts** tab or the **Counters** tab.
3. Select the part of the test for which the rule will be created. This can be a script, folder, instruction, or synchronization point on the Scripts tab, or a Windows or Xen counter on the Counter tab.
4. Click the Add a Load Control Rule button in the display toolbar. The Load Control Rule dialog box is displayed.
5. Enter a name for the rule. Names should be unique and descriptive to allow you to easily distinguish rules when adding them to the display.
6. Specify the rule conditions:
   - Specify when the rule is to be activated by entering a value and choosing whether activation occurs when the actual value matches or exceeds (> or >=) or matches or falls below (< or <=) the value you supply. For example, if you want to trigger the rule when the Login Fail Rate measurement exceeds 15 per minute, choose > and enter 15.
   - Specify when the rule is to be reset by entering a value and choosing whether resetting occurs when the actual value matches exceeds (> or
>=) or matches or falls below (< or <=) the value you supply. For example, if you want to reset the rule when the Login Fail Rate measurement falls below 5 per minute, choose < and enter 5.

- Click the **Require parameter consistently over/below threshold** checkbox and enter a value in seconds that the condition must exist before activation or resetting occur. It is recommended that you use this feature to avoid implementing rules in reaction to brief spikes in activity.

7. Specify the rule effect by choosing one of the following options:

- Terminate test—When the rule is activated, the test is stopped.
- Prevent starting users for affected scripts until all active rules have been reset—When the rule is activated, new users are not started until all the active rules associated with the selected script(s) have been reset.
- Prevent starting users until user count for each affected script drops below \( n \)%—When the rule is activated, new users are not started until the user count for the selected script(s) falls below a specified percentage of the number of users at the time the rule was activated.
- Prevent starting users for the affected scripts for \( n \) seconds—When the rule is activated, new users are not started until the specified number of seconds has elapsed.

8. Select one or more test scripts and click **Add** to apply the rule.

9. Select additional parameters as required:

- Do not apply the rule for the first \( n \) seconds of the test—Setting this parameter allows the test script to get through any initialization phases without having the rule activated.
- Reactivate the rule no more than \( n \) times—Setting this parameter allows you to limit the number of times a rule is activated for each running of the test script.

10. Click **OK** to complete rule creation.

## Starting a Test

Once you have defined connections, users, loads, and (optionally) load control rules, you can start a load test. You cannot start a load test with more users than you are licensed for, nor can you start one with more users than you have defined.
To start the test:, select the Test node in the Test Tree pane and click the Replay Test button on the main tool bar. When you replay or debug a test, the test status and the running time are displayed in the script toolbar. Status messages and times are displayed in the Message pane at the bottom of the main window.

**Viewing the Test Windows**

When you start a test, a window for each user in the test is opened. By default, all of the windows are opened in the upper left hand corner of the screen. The windows are stacked on top of each other, making only one window visible.

You can view all of the user windows by displaying the windows tiled horizontally or vertically. To enable this display:

1. In the Windows Task bar, right-click the Citrix EdgeSight for Load Testing - TUser task
2. Select Tile Horizontally or Tile Vertically to display all of the test users.

This view of the test allows you to monitor the progress of the test by viewing each user.

**Replaying a Test in Debug Mode**

As you develop the test, you may find it useful to run the test in debug mode. This allows you to set break points within a script. and then step through the script, either step by step, or to the next breakpoint. As the script runs, the active step is highlighted.

**Note** You cannot currently edit a script which is being replayed in debug mode.

To debug a test:

1. Identify the steps in the script that will be used as breakpoints. In the Test Tree, open the **Instructions** folder and right click on an instruction and select **Set Breakpoint** to set a breakpoint. (To clear a break point, right click the step again and select **Clear Breakpoint**.)
2. From the main toolbar, click the Debug Test button. The Recording User message box is displayed.
3. Click **OK** to use the selected user as the debugging user. The TUser window is displayed. As instructions are executed, they are highlighted in the Test Tree and the caption of the TUser window is updated. Test execution continues until a breakpoint is reached.
4. To continue execution, click the Continue Test Execution or the Step to the Next Instruction button. Continue causes the test to be executed to the next breakpoint and Step Next causes the next instruction to be executed.
You may find the following keyboard shortcuts helpful when debugging tests:

- F5 – Test Debug
- Shift+F5 – Debug Continue
- F9 – Set instruction breakpoint
- Shift + F9 – Clear instruction breakpoint
- F10 – Debug Step Next

**Stopping a Test**

You can stop a test using the Main Menu or the main tool bar. There are two modes you can use to stop a test, Stop and Cancel.

When you Stop a test, the controller allows the users that are running scripts to continue to completion. After the scripts are completed, the clients are shut down and the licenses are returned to the license server. This is the most orderly method for stopping a test. The test may take some time to stop, based on the size of the script that is being executed, but this is the recommended way to stop a test.

When you Cancel a test the controller terminates all scripts that are running, shuts down virtual user sessions, and returns the licenses to the license server. This method is not recommended.

**Scheduling Tests**

You can schedule tests to automatically run at a designated date and time. To schedule a test:

1. In the main toolbar, click the Schedule Test button. The Test Scheduler dialog box is displayed
2. Enter the Date and Time you want the test to start. Click Schedule to add the test to a list of scheduled tests. You can add multiple test times to the schedule.
3. Select Start logs automatically to create a log file of this test. A log file must have been defined for the test previously. The file will not be created by the test scheduler.

You can remove scheduled test from the list by selecting a test and clicking the Remove button.
Displaying Test Results

The display screens provide runtime information about the tests you are executing. The screens provide the following information:

- Scripts - provides an overview of the overall performance of the test you are executing and allows you to get more detailed performance information about certain parts of the test.
- Connections - displays performance statistics for each Launcher used in a test and displays statistics for specific instructions across each launcher.
- Counters - displays the counters that have been defined to measure the tests.
- Load Control Rules - displays the Load Control Rules which have been defined for the test.
- Alarms - displays alarms that have been triggered during test execution.
- Monitor - provides a graphical display of selected measurements.
- Measurements - provides a list of selected measurements that can be displayed in the Monitor screen. Used to add and remove instructions from the Display screen.
- Messages - displays messages generated during test execution.
To access the display screens, select Display in the Test Tree pane. The different screens are available by selecting the corresponding tabs in the Main window. A portion of the display is shown in the following figure.

You configure the display properties using the Display Properties dialog. To open this screen, right-click Display and select Properties.

Using the Display Properties dialog box you can configure the measurements that are displayed and the format of the Monitor screen.

Quality of Service (QOS) levels are used to visually indicate where system stress is causing delays in the executing scripts.
The QOS levels are delays measured against the average time an instruction takes to execute. The QOS levels are configured to highlight measurements when the preset performance levels are exceeded. The default settings for the QOS levels are:

- Level 3 - 30 seconds
- Level 2 - 20 seconds
- Level 1 - 10 seconds

Warning levels are displayed on the Scripts Window. Measurements exceeding the QOS levels are displayed in colors: Red (level 3), Orange (level 2), and Yellow (level 1).

<table>
<thead>
<tr>
<th>Connect</th>
<th>41</th>
<th>2.1 (1.1)</th>
<th>65</th>
<th>2.5 (1.5)</th>
<th>39</th>
<th>7.4 (6.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>34</td>
<td>30.1 (0.9)</td>
<td>63</td>
<td>30.3 (1.0)</td>
<td>48</td>
<td>33.7 (4.4)</td>
</tr>
<tr>
<td>Disconnect</td>
<td>34</td>
<td>1.7 (0.9)</td>
<td>62</td>
<td>1.9 (1.0)</td>
<td>49</td>
<td>3.8 (2.9)</td>
</tr>
</tbody>
</table>

## Displaying Script Performance

The Scripts display reports the performance of the test as a whole. Performance statistics for the following are provided:

- User count - reports the average number of users currently executing the script.
- Connections - reports the average time required for each user to connect to the server.
- Disconnections - reports the average time required for a user to disconnect from the server.
- ICA Ping - reports the number of ICA Pings performed. High rates of ICA Pings may be an indication of connection problems between the users and the server.
- Script - displays performance of the currently defined scripts. Each script can be selected to display performance of individual instructions.
- Login Rate (/min) - displays the number of successful logins per minute.
- Login Fail Rate (/min) - displays the number of login failures and dropped connections per minute.
For each measurement in the Scripts window, averaged performance data is shown in cells to the right of the measurement name.

<table>
<thead>
<tr>
<th>Type</th>
<th>Script</th>
<th>Measurement</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>demo</td>
<td>demo</td>
<td>User Count</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>demo</td>
<td>demo</td>
<td>Connect</td>
<td>24</td>
<td>4.2</td>
<td>0</td>
</tr>
<tr>
<td>demo</td>
<td>demo</td>
<td>Instructions (1)</td>
<td>35</td>
<td>65.1</td>
<td>2</td>
</tr>
<tr>
<td>demo</td>
<td>demo</td>
<td>Disconnect</td>
<td>14</td>
<td>10.3</td>
<td>0</td>
</tr>
<tr>
<td>demo</td>
<td>demo</td>
<td>ICA Ping (ms)</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>demo</td>
<td>demo</td>
<td>Login Rate (/min)</td>
<td>20</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td>demo</td>
<td>demo</td>
<td>Login Fail Rate (/min)</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each data cell displays the following:

- The center value is the average response time for the selected measurement in seconds, or the average number of executing users in the case of a User Count measurement.
- A center value in parentheses indicates the average delay for users at this point in the script measured against the fastest execution.
- The value on the left indicates the number of virtual users that have successfully executed the measurement.
- The value of the right indicates the number of virtual users that have failed to execute the measurement.

The average delay, success, and failure values are optional and are set in the Display Properties dialog box.

As a test executes, the cells in the Scripts display are populated with data. The response time for each cell is averaged over the interval set for the cell. The default interval is two minutes. Pass and fail counts are totalled over the same interval.

You can change the start time and the interval length using the buttons on the script toolbar. Clicking the Move Left and Move Right buttons changes the start time and clicking the Zoom In and Zoom Out buttons changes the interval length.

Detailed performance information can be obtained for Script, Folder, and Synchronization Point measurements by drilling down on a measurement. Detailed performance information is provided for the immediate children of the measurement you select.
Instructions grouped together in a folder allow you to obtain performance data for specific transactions (the specific transactions inside the folder). Double-clicking on the Instructions cell results in a display similar to the following:

<table>
<thead>
<tr>
<th>Type</th>
<th>Script</th>
<th>Measurement</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>demo</td>
<td>ICA Seamless Host ...</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>demo</td>
<td>Program Manager (5)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>demo</td>
<td>Start Menu (8)</td>
<td>12</td>
</tr>
</tbody>
</table>

**Connections Window**

Use the Connections window to display the statistics for each connection to the server or to displays the status of a single measurement across multiple connections.

Use the following method to display all of the connections in a test:

1. In the Test Tree pane, select Display.
2. In the main window, double-click the Scripts tab.
3. From the Scripts listing, select Connect
4. Select the Connections tab. All of the connections statistics are displayed:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) localhost</td>
<td>8 3.9 0</td>
</tr>
<tr>
<td>(2) system2.my...</td>
<td>7 2.1 0</td>
</tr>
<tr>
<td>(3) system3.my...</td>
<td>6 4.7 0</td>
</tr>
</tbody>
</table>

You can select a measurement from the Scripts Window and display the measurement for each connection in the test. This display allows you to compare the performance of the instruction across different connections.

To add instruction measurements to the Connection Window:

1. In the Scripts Window, select the measurement that you want to display across all connections.
2. Select the Connections tab. The measurement is displayed for all of the connections in the test. The name and number of the instruction is displayed as the title of the display.

You can add measurements from the Connections window to the Monitor window. This allows you to graphically compare the measurement across multiple connections. To add connections measurements to the Monitor window:

1. Right-click the measurement in the Connections window.
2. Select Add Measurement to Display.
3. In the Measurements window, select the check box next to the measurement.

**Counters Window**

Selecting the Counters tab displays the current averaged values of performance counters you have selected to monitor. The performance counters are ones selected for monitoring from the Microsoft Performance counters or from the Xen Server counters.

**Add Windows Counter**

Use the following steps to add Windows performance counters to the test. Performance counters cannot be added to a test that is running.

1. In the Test Tree pane, select **Display > Counters**.
2. From the display toolbar, select the Add Windows Counters button.
3. In the Select a Counter to Add screen, click the Select Counters from Computer button.
4. Enter the network address of the server under test.
5. Select the Performance Object field. A connection is created to the server you specified.
6. Select the counter to monitor.
   **Note:** Each counter must be added individually, the Select All option does not work.

**Note** Selecting a Counter Detail increases and decreases the number of counters available to select from. The Wizard level provides the greatest number of counters, while the Novice level provides the fewest counters.

Counters are displayed as follows:

<table>
<thead>
<tr>
<th>Counter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>\BOSALD802\Process(Total)\Page Faults/sec</td>
<td>1133.0</td>
</tr>
<tr>
<td>\BOSALD802\Process(Total)\Working Set</td>
<td>1664163840.0</td>
</tr>
<tr>
<td>\BOSALD802\Processor(Total)% User Time</td>
<td>2.0</td>
</tr>
</tbody>
</table>

**Add Xen Counter**

Use the following steps to add Xen performance counters to the test. Performance counters cannot be added to a test that is running.
1. In the Test Tree pane, select Display > Counters.

2. From the display toolbar, select the Add Xen Counters button.

3. In the Select Xen Server Counters screen, enter the IP address for the Xen pool master in the Xen Pool Master’s IP Address field.

4. Edit the port number as required.

5. Enter a username and password used to access the Xen pool master.

6. Click Query. The Xen Systems list box is populated with the names of systems in the Xen pool. Both physical and virtual machines are displayed. Virtual machine names are indented under the associated physical machine.

7. Select a machine and then click the checkbox for each Xen counter to be collected for that machine. Continue selecting machines and counters as needed. As counters are added, they are displayed in the Added Counters list box. You can select one or more counters from the list and click Remove to delete the counter from the list, or click Remove All to delete all selected counters.

8. When you have finished selecting counters, click Add.

**Delete Counter**

You cannot delete a counter from the Counters display. Use the following procedure to delete counters.

1. In the Test Tree pane, select Display > Counters.

2. A list of counters is displayed in the Main window.

3. Select the counters you want to delete, right click on the selection and select Delete.

You can also delete all counters by right clicking in the Main windows and selecting Clear All Windows Counters or Clear All Xen Counters

**Load Control Rule Window**

Use the Load Control Rules window to display the load control rules which have been enabled and displays the percentage of time that they were activated for each time period. (To enable or disable a rule, go to Display > Load Control Rules, right-click the rule, and select Enable/Disable.) For information on creating a load control rule, see “Adding Load Control Rules” on page 46. For a sample load control rule and the resulting load test data, see Chapter D, “Intelligent Load Control Example.”
Delete Load Control Rule
To delete load control rules.
1. In the Test Tree pane, select Display > Load Control Rules.
2. A list of rules is displayed in the Main window.
3. Select the rules you want to delete, or select multiple rules to be deleted.
4. Right-click the rule(s) and select Delete.

Alarms Window
You can add alarms that are triggered when a measurement you have specified reaches a certain value. The alarms can be used to generate records of performance events and can be configured to provide email notification of events.

To configure alarms, your test must already contain performance counters.
Selecting the Alarms tab displays the Alarms window. This window lists alarms that have been triggered during the current execution of the load test.

Add Alarm
Use the following steps to configure alarms:
1. In the Test Tree pane, select Display > Counters.
2. In the Main window, select the Counter Tab.
3. From the list of counters, select the counter that you want to add an alarm to, right click the selection, and select Set Alarm.
4. Enter the Alarm specifications in the Alarm Properties: New Alarm dialog box:
   a. Specify an alarm type:
      Value alarms are triggered when the average value of a measurement reaches a specified level.
      Pass Rate alarms are triggered when the pass rate (in percent) of the measurement reaches a specified percentage value.
   b. Specify if the alarm is triggered by a greater-than or less-than value.
   c. Specify an alarm trigger value.
   d. Choose a sampling interval.
   e. Choose a reset interval.
   f. Enter any comments which might help users understand the context of the alarm or appropriate actions.
To save the alarm to a file:
1. Click the Write to File when triggered check box.
2. Enter the filename for the Alarm file.

To receive email notification of triggered alarms:
1. Click the Send SMTP email when triggered check box.
2. Enter the email message specification.

**Delete Alarm**

Use the following steps to delete an alarm:
1. In the Test Tree pane, select *Display > Alarms*.
2. In the Main window, select the Alarm you want to delete, right click the selection, and select *Delete*.

**Measurement Window**

The Measurement window is positioned below the Display window, and is used to show summary data for selected measurements. From this display, measurements can be selected for display in the Monitor window.

The measurements in this window can include Microsoft Performance counters, Xen Server counters, and counters selected from the Scripts display.

<table>
<thead>
<tr>
<th>Display</th>
<th>Scale</th>
<th>Script</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>10</td>
<td>demo</td>
<td>Login Rate (min)</td>
</tr>
<tr>
<td>✔️</td>
<td>10</td>
<td>demo</td>
<td>Login Fail Rate (min)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>demo</td>
<td>User Count</td>
</tr>
</tbody>
</table>

To add a measurement to the Measurement Console:
1. From the Scripts, Connections, Counters, or Load Control Rules window, select the measurement you want to add to the real time display.
2. From the display toolbar, select the Add a Monitor Measurement button.
The fields in the measurements display include the following:

- **Display** - contains an icon for the type of measurement, the color representing the measurement in the Monitor screen, and a check box to enable displaying the measurement in the Monitor screen.
- **Scale** - allows you to select a scale for the measurement in the Monitor screen.
- **Script** - identifies the script that the measurement is from.
- **Measurement** - identifies the measurement. An instruction contains an identification number.
- **Connection** - identifies the connection the measurement is from.
- **Last** - displays the most recent measurement.
- **Max.** - displays the highest measurement recorded.
- **Min.** - displays the lowest measurement recorded.

**Monitor Window**

The Monitor Window is used to plot the average values of selected test measurements against elapsed time. You can use it to examine the trends of multiple measurements simultaneously.

To view measurements in the graph view, you need to add it to the Measurement Console. Once added, click the check box alongside the measurement in the Measurement Console and then choose the Monitor tab. The average value of the measurement will be plotted on the graph against elapsed time.

You can change the start time and the interval length using Zoom Out and Zoom Into and the Move Left and Move Right buttons on the script toolbar. Clicking the Move Left and Move Right buttons changes the start time and clicking the Zoom Out and Zoom Into changes the interval length.

**Messages Window**

The Message Window displays status and error messages while tests execute. Messages displayed in this screen are cleared each time a test is started or restarted.
A sample from the Messages Window follows:

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>00:00:00</td>
<td>Starting test...</td>
</tr>
<tr>
<td>00:00:00</td>
<td>Looking for XenApp Platinum or Enterprise license...</td>
</tr>
<tr>
<td>00:00:00</td>
<td>Found XenApp Platinum license.</td>
</tr>
<tr>
<td>00:00:00</td>
<td>Connecting to launchers...</td>
</tr>
</tbody>
</table>
Chart Reports

You can view and save measurements displayed in the Monitor Window using the Chart Report function. Chart Reports are created as Microsoft Web Archive files (.mht) and saved in the default folder Citrix\Citrix EdgeSight for Load Testing\Reports. Each report includes the graphical data shown in the Monitor window, a list of measurement items and scales, and a table with the detailed measurement data.

To create a chart report:

1. In the Monitor Window, check the measurements you want to see in the report.
2. From the main toolbar, click the Generate a Chart Report button.
3. In the Chart Report Properties dialog, enter the Name of the chart report.
4. Enter a file name for the chart report in the Save As field.
5. If you want to save the chart report to a non-default location, click the Browse button and navigate to the selected folder. Click Save to save the new location.
6. Enter any comments you want included in the Chart Report.
7. Click OK.

Any measurements that you have highlighted are displayed in the Chart Display report. The resulting report includes the Monitor Window, the scale information for each measurement, and the data collected for each measurement.
Virtual Keys Reference

This Appendix describes the virtual keys supported as keyboard input. This information is provided as a supplement to the Keyboard Editing information in “Editing Keyboard Input” on page 37.

The following table lists the supported virtual keys and describes the actual keys they support.

<table>
<thead>
<tr>
<th>Virtual Key</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>VK_NUMLOCK</td>
<td>Numbers lock</td>
</tr>
<tr>
<td>VK_SCROLL</td>
<td>Scroll</td>
</tr>
<tr>
<td>VK_BACK</td>
<td>Backspace</td>
</tr>
<tr>
<td>VK_TAB</td>
<td>Tab</td>
</tr>
<tr>
<td>VK_CLEAR</td>
<td>Clear</td>
</tr>
<tr>
<td>VK_RETURN</td>
<td>Return</td>
</tr>
<tr>
<td>VK_CONTROL</td>
<td>Ctrl</td>
</tr>
<tr>
<td>VK_MENU</td>
<td>Alt</td>
</tr>
<tr>
<td>VK_PAUSE</td>
<td>Pause</td>
</tr>
<tr>
<td>VK_CAPITOL</td>
<td>Caps Lock</td>
</tr>
<tr>
<td>VK_ESCAPE</td>
<td>Esc</td>
</tr>
<tr>
<td>VK_PRIOR</td>
<td>Page Up</td>
</tr>
<tr>
<td>VK_NEXT</td>
<td>Page Down</td>
</tr>
<tr>
<td>VK_END</td>
<td>End</td>
</tr>
<tr>
<td>VK_HOME</td>
<td>Home</td>
</tr>
<tr>
<td>VK_LEFT</td>
<td>Left Arrow</td>
</tr>
<tr>
<td>VK_UP</td>
<td>Up Arrow</td>
</tr>
<tr>
<td>VK_RIGHT</td>
<td>Right arrow</td>
</tr>
<tr>
<td>VK_DOWN</td>
<td>Down arrow</td>
</tr>
<tr>
<td>VK_SELECT</td>
<td>Select</td>
</tr>
<tr>
<td>Virtual Key</td>
<td>Key</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>VK_EXECUTE</td>
<td>Execute</td>
</tr>
<tr>
<td>VK_SNAPSHOT</td>
<td>Print Screen</td>
</tr>
<tr>
<td>VK_INSERT</td>
<td>Insert</td>
</tr>
<tr>
<td>VK_DELETE</td>
<td>Delete</td>
</tr>
<tr>
<td>VK_HELP</td>
<td>Help</td>
</tr>
<tr>
<td>VK_F1</td>
<td>F1</td>
</tr>
<tr>
<td>VK_F2</td>
<td>F2</td>
</tr>
<tr>
<td>VK_F3</td>
<td>F3</td>
</tr>
<tr>
<td>VK_F4</td>
<td>F4</td>
</tr>
<tr>
<td>VK_F5</td>
<td>F5</td>
</tr>
<tr>
<td>VK_F6</td>
<td>F6</td>
</tr>
<tr>
<td>VK_F7</td>
<td>F7</td>
</tr>
<tr>
<td>VK_F8</td>
<td>F8</td>
</tr>
<tr>
<td>VK_F9</td>
<td>F9</td>
</tr>
<tr>
<td>VK_F10</td>
<td>F10</td>
</tr>
<tr>
<td>VK_F11</td>
<td>F11</td>
</tr>
<tr>
<td>VK_F12</td>
<td>F12</td>
</tr>
<tr>
<td>VK_F13</td>
<td>F13</td>
</tr>
<tr>
<td>VK_F14</td>
<td>F14</td>
</tr>
<tr>
<td>VK_F15</td>
<td>F15</td>
</tr>
<tr>
<td>VK_F16</td>
<td>F16</td>
</tr>
<tr>
<td>VK_F17</td>
<td>F17</td>
</tr>
<tr>
<td>VK_F18</td>
<td>F18</td>
</tr>
<tr>
<td>VK_F19</td>
<td>F19</td>
</tr>
<tr>
<td>VK_F20</td>
<td>F20</td>
</tr>
<tr>
<td>VK_F21</td>
<td>F21</td>
</tr>
<tr>
<td>VK_F22</td>
<td>F22</td>
</tr>
<tr>
<td>VK_F23</td>
<td>F23</td>
</tr>
<tr>
<td>VK_F24</td>
<td>F24</td>
</tr>
<tr>
<td>VK_LCONTROL</td>
<td>Left Ctrl</td>
</tr>
<tr>
<td>VK_RCONTROL</td>
<td>Right Ctrl</td>
</tr>
<tr>
<td>Virtual Key</td>
<td>Key</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>VK_LMENU</td>
<td>Left Alt</td>
</tr>
<tr>
<td>VK_RMENU</td>
<td>Right Alt</td>
</tr>
<tr>
<td>VK_SHIFT</td>
<td>Shift</td>
</tr>
<tr>
<td>VK_LWIN</td>
<td>Left Windows</td>
</tr>
<tr>
<td>VK_RWIN</td>
<td>Right Windows</td>
</tr>
</tbody>
</table>
Creating an ICA File

EdgeSight for Load Testing can connect to published applications and applications served through Web pages using ICA files.

You can create ICA files for published applications. The procedure for creating ICA files depends on the version of Presentation Server software you are running.

**Note** Creating ICA files using the following procedures may cause Load Tests to run with different default settings than those set on your system. These settings, which may include audio and window defaults, can be modified by editing the ICA files.

Creating ICA Files for Versions Prior to Presentation Server 4.5

Use the following steps to create an ICA file:

1. Open the Citrix Management Console and navigate to the published application you want to use.
2. Right click on the application you want to create an ICA file for.
3. Select Create ICA File.
4. Follow the steps presented by the Create ICA Wizard. By default, the ICA file is created in the Program Files\Citrix\Administration folder.
5. Copy the ICA file to the EdgeSight for Load Testing\ICA Files folder of the machine the Launcher(s) is running on.

When you create a Connection, use the ICA file you just created.

Creating ICA Files for Presentation Server 4.5 and Later

Use the following steps to create an ICA file on a Presentation Server version 4.5 System:

1. Launch the Internet Explorer Web browser and navigate to the Citrix Web Interface.
2. Log in to the Citrix Web Interface site.
3. Right-click the application that you want to create an ICA file for.
4. Select Save Target As....
5. Save the file to the EdgeSight for Load Testing\ICA Files folder.
6. Edit the file, deleting or commenting out the lines beginning with the following:
   • RemoveICAFile=
   • LogonTicket=
   • LogonTicketType=

**Creating ICA Files from the APPSRV.ini File**

You can use this procedure when using the XML service as a connection method. Use the following steps to create an ICA file from the APPSRV.ini file:

1. From the Start Menu, select All Programs > Citrix > Citrix Access Clients > Program Neighborhood.
2. Double-click Application Set Manager.
3. Double-click Custom ICA Connections.
4. Double-click Add ICA Connection.
5. In the Add New ICA Connection wizard:
   • Enter the connection type.
   • Enter a description (Name), the network protocol, the server name, and the server desktop or the published application to connect to.
   • Select the encryption level.
   • Enter the Username, Password, and Domain for the account that will use the connection.
   • Enter the display settings.
   • Leave the Application and Working directory fields blank.
   • Click Finish.
6. Edit the C:\Documents and Settings\username\Application Data\ICA\Client\APPSRV file.
7. Search for the a line containing [ApplicationServers].
8. Delete the lines below this line except for the one containing the ICA name you entered in the Wizard, for example:
   [ApplicationServers]
   test2=

9. Write these lines to a new file.

10. Search for the name of the ICA you created with the Wizard. The name should appear as [name].

11. Select the [name] field and all lines below (before you encounter another [name] field).

12. Append these lines to the new file. The file should look similar to the following:

   [ApplicationServers]
   test2=
   [test2]
   TransportDriver=TCP/IP
   BrowserProtocol=HTTPonTCP
   DoNotUseDefaultCSL=Off
   Description=test2
   Address=tload01.qalab.local
   IconPath=C:\Program Files\Citrix\ICA Client\pn.exe
   .
   .
   .
   UseLocalUserAndPassword=Off
   DisableCtrlAltDel=On
   UIFlags=12
   SSLEnable=Off
   SSLNoCACerts=0
   SSLCiphers=ALL
   CGPAddress=*.
   SSLProxyHost=:443

13. Copy this file to the Citrix EdgeSight for Load Testing\ICA Files folder.
This section contains a simple script that demonstrates the use of some of the editing functions that are described in this book:

- The script is organized into folders
- It uses an iterating folder
- Mouse instructions have been replaced by keyboard instructions
- Wildcards are used in the Microsoft Windows screen captions

Without expanding the folders, the script is displayed in following figure.
Close Existing Doc Folder

The first folder of the script closes any file that is open when Microsoft Word starts. The File > Close mouse commands have been replaced with keyboard input.
Word Task Folder

The Word Task folder is an iterating folder and is organized with three subfolders. The iterating folder executes the instructions in the folder continuously, until the controller stops the iteration and exits the test.

Open New Doc Folder

In this folder, a new Microsoft Word document is opened. Because Microsoft Word uses captions with incremented titles (Document\text{n} Microsoft Word), the caption for the Microsoft Word Synchronization Point uses a wildcard. This ensures that the caption always matches.
Enter Text Folder
The only instruction in this folder is the keyboard instruction that enters text into the Microsoft Word file.

Close Doc Folder
In this folder, the Microsoft Word file is closed without saving the file. A File > Close instruction is entered with keyboard instructions and a mouse instruction responds to the query to save the file.

Exit App Folder
The Exit App folder contains instructions that exit the Microsoft Word application and result in the connection to the server being closed. Note that the keyboard command is entered as a single command. In the Iterating folder, a similar keyboard command was entered as multiple commands.
Intelligent Load Control Example

This section contains a sample implementation of intelligent load control through the use of load control rules. The sample rule and chart report illustrates the usage and effect of a single, simple load control rule.

The intent of using load control rules is to allow the test to probe an area of interest in terms of system stress. Instead of a steady increase in user count which can stress the system beyond the point of producing meaningful data, intelligent load control allows EdgeSight for Load Testing to detect system stress and respond by reducing user count. This allows the system to recover to a point where the user count can start to increase again. This cycle of stress and adjustment produces repeatable and accurate data about system capacity.

Sample Load Control Rule

The following figure shows the sample load control rule parameters, including the following:

- When to activate and reset the rule. In this case, the rule is activated when the % disk time counter exceeds 90%, and is reset when the counter falls below 75%.

- A time period parameter required for rule activation. In this case, the rule is not activated until the % disk time counter exceeds 90% continuously for 90 seconds. This parameter helps prevent the rule from being triggered in response to transient spikes.

- The effect of the rule. All effects result in a reduction in user count. In this case, no new users are created until all active rules have been reset. (There can be multiple rules applied to a single script.)

- The script to which the rule will be applied. In this case, the Word Demo script is selected. The script has users open Microsoft Word, enter text, and close the application.
The metric of percent (%) physical disk time was chosen because it is a known bottleneck for the particular script on the system under test. Other potential candidates for metrics are average disk queue, average CPU, memory, and network utilization.

When defining the load, the user concurrency should be chosen to rise at a rate which does not trigger the rule prematurely, creating unnecessary false positives and artificially prolonging the test duration. In this case, the load is defined as 120 users over a period of 15 minutes.

A good test will include a rule that triggers at a predictable rate, cycling up and down at an even level.
Sample Chart Report

Rule activation in relation to various metrics can be shown by adding the rule and desired metrics to the Monitor tab display. As seen in the chart report example below, the threshold values of the selected counters are in sync. As user count increases, disk time, disk queue length, and processor time also rise. After the load control rule is activated, the user count decreases, along with the selected metrics. This indicates that each metric would render the same result when used as a rule parameter.

The alternate and repeated effect of reducing the load when performance parameters are out of normal operating ranges, and then increasing the load when the parameters return within the normal operating ranges provides fast and accurate capacity determination for the tested system.

The following represents the outcome of a test that benefits from using load control rules.

Load Test Chart Report

Report Name: LCR-Demo-PDT[90-75]-120u15m
File Name: C:\Documents and Settings\My Documents\Citrix EdgeSight for Load Testing\Reports\LCR-Demo-PDT[90-75]-120u15m.mht
Report Generated: May 13 2009 17:26:06
Comments: Load Control Rule Demo - Physical Disk Time [90-75] - 120 users over 15 minutes
Load Control Rule Runtime Activation Messages

The following represent runtime messages pertinent to rule activation.

00:00:00 Starting test...
00:00:00 Looking for XenApp Platinum or Enterprise license...
00:00:02 Found XenApp Platinum license.
00:00:02 Connecting to launchers...
00:00:03 Starting clients...
00:00:19 Executing test...
00:06:00 Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 100.0.
00:08:30 Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 37.6 25.4 27.1.
00:08:30 Rule % Disk Time - 90<->75 reset.
00:08:31 Load advanced 300 seconds, started 00:05:00, ends 00:20:00
00:10:01 Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 100.0.
00:12:01 Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 0.5 11.1 74.6.

<table>
<thead>
<tr>
<th>Item</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Disk Time - 90&lt;-&gt;75 - All</td>
<td>10</td>
</tr>
<tr>
<td>Word Demo\User Count - All</td>
<td>100</td>
</tr>
<tr>
<td>\test01\PhysicalDisk(_Total)% Disk Time - All</td>
<td>100</td>
</tr>
<tr>
<td>\test01\PhysicalDisk(_Total)\Avg. Disk Queue Length - All</td>
<td>10</td>
</tr>
<tr>
<td>\test01\Processor(_Total)% Processor Time - All</td>
<td>100</td>
</tr>
</tbody>
</table>
00:12:01  Rule % Disk Time - 90<->75 reset.
00:12:01  Load advanced 210 seconds, started 00:08:30, ends 00:23:30
00:14:01  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 92.3.
00:16:31  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 0.4 5.2 8.3.
00:16:31  Rule % Disk Time - 90<->75 reset.
00:16:31  Load advanced 270 seconds, started 00:13:00, ends 00:28:00
00:18:03  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 100.0.
00:20:03  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 10.4 5.1 57.2.
00:20:03  Rule % Disk Time - 90<->75 reset.
00:20:04  Load advanced 243 seconds, started 00:17:03, ends 00:32:03
00:22:03  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 91.5.
00:24:03  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 0.3 4.0 65.4.
00:24:03  Rule % Disk Time - 90<->75 reset.
00:24:05  Load advanced 218 seconds, started 00:20:41, ends 00:35:41
00:26:04  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 99.0.
00:28:04  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 0.7 5.2 45.6.
00:28:04  Rule % Disk Time - 90<->75 reset.
00:28:05  Load advanced 248 seconds, started 00:24:49, ends 00:39:49
00:30:04  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 100.0.
00:32:34  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 0.4 6.4 13.2.
00:32:34  Rule % Disk Time - 90<->75 reset.
00:32:34  Load advanced 239 seconds, started 00:28:48, ends 00:43:48
00:34:04  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 100.0.
00:36:34  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 14.7 5.3 3.5.
00:36:34  Rule % Disk Time - 90<->75 reset.
00:36:34  Load advanced 278 seconds, started 00:33:26, ends 00:48:26
00:38:30  Rule % Disk Time - 90<->75 activated, activation values are 100.0, 100.0, 100.0.
00:40:01  Rule % Disk Time - 90<->75 reset when parameter value = 75.0, current values = 58.3 66.9 68.1.
00:40:01  Rule % Disk Time - 90<->75 reset.
00:40:01  Load advanced 139 seconds, started 00:35:45, ends 00:50:45
00:42:01  Rule % Disk Time - 90<->75 activated, activation values are 100.0,
100.0, 91.2.
00:44:02  Rule % Disk Time - 90<->75 reset when parameter value = 75.0,
current values = 0.2 2.8 67.3.
00:44:02  Rule % Disk Time - 90<->75 reset.
00:46:41  Shutting down clients...
00:46:43  Test stopped.
Index

C
Chart Reports 63
Conditional folders 31
Connection
  XML service 14
Creating ICA files 14
Creating users 17
E
Editing instructions
  navigation 27
F
fast record 23
Folder
  iterate 30
  repeating 30
Folders 29
  conditional 31
  organizing instructions 29
I
ICA file
  connecting with 14
Iterating folders 30
Iteration 30
J
JScript 42
  built in 42
L
Load
  define 9
  repeating folders 30
Load control rule 46
example 77
M
Microsoft JScript
  See JScript 42
R
Record 23
Recording 23
  starting 20
  stopping 22
replaying 23
S
Script
  create 11
Scriptlet Instructions 42
Scripts
  creating 19
  navigation 27
U
Users
  create 17
  creating 17
  variables 41
V
Variables 41
  delete 41
  importing 42
X
XML service
  connecting with 14