



HP LoadRunner® and Citrix®

Advanced Techniques for Performance Engineers



Table of Contents

Overview	3
Citrix® Administration and Setup	3
<i>Reset Connections on Disconnect</i>	3
<i>Allow Multiple User Instances</i>	5
<i>Using Terminal Services on Generators</i>	7
<i>IP Spoofing With Citrix</i>	10
Advanced Citrix® Scripting.....	11
<i>Execute on Window</i>	11
<i>Dynamic Synchronization on Bitmap Changes</i>	12
<i>Error Handling</i>	14
<i>Change Timeout</i>	15
<i>Remotely Logout Hanging Sessions</i>	17
Summary	21
About Northway Solutions Group	22
About HP Software	22
About Citrix®	23

Overview

This document is intended to educate IT managers and assist performance engineers who are interested in performance testing Citrix-deployed applications using HP Software's LoadRunner product. In 2006, Northway published a initial whitepaper on the topic (see "Performance Testing Citrix Applications Using LoadRunner" by Tim Chase on www.northwaysolutions.com for more information). This document expands on that information and explores Citrix server configuration settings, as well as more advanced scripting techniques within LoadRunner Vugen. This knowledge is the result of lessons learned from experience load testing Citrix-deployed applications over the last several years.

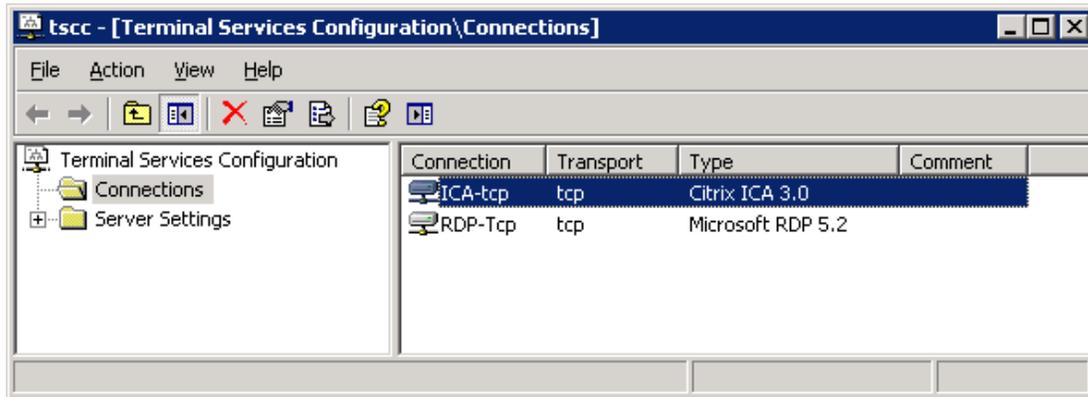
Citrix® Administration and Setup

Depending on the Citrix expertise available, the performance engineer may need to become a part-time Citrix server administrator to get projects done on time. There are several settings that must be addressed by the Citrix administrator for successful test execution.

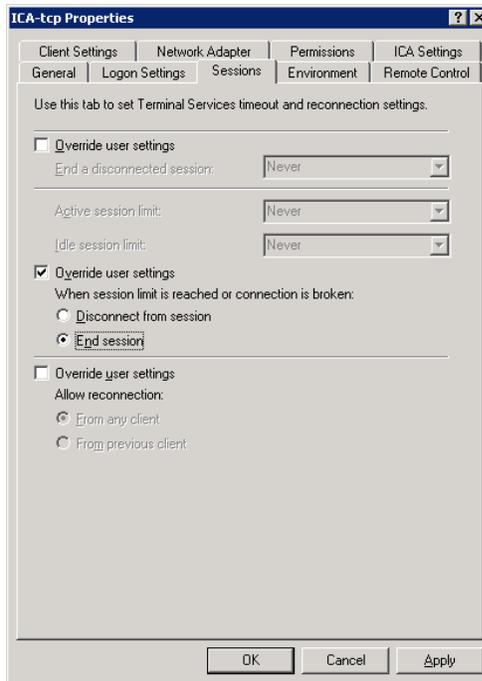
Reset Connections on Disconnect

Any session that disconnects should be reset to login, instead of leaving it open after a disconnect. Leaving sessions open will hinder Vugen script debugging. By default, the ICA TCP connections are set to disconnect sessions with broken or timed out connections, but these have usually been modified to allow users to disconnect and reconnect to the last state of their session. The connection can be configured to reset, or log off sessions with broken or timed out connections. When a session is reset, attempting to reconnect initiates a new session; rather than restoring a user to the same place in the application in use, the application is restarted.

To ensure that each user instance receives a clean Citrix session, the following modifications should be made within the ICA configuration settings in the Terminal Services Configuration MMC snap-in (tscc.msc) under the ICA Settings and Environment tabs. See the image below:



1. Select the ica-tcp connection, right-click and select **Properties**. Alternatively, double-click on the connection. The "Edit Connection" dialog box will open.
2. Select the Sessions Tab.
3. Enable the "Override user settings" check box
4. Select the "End Session" option, and click OK to save this configuration. See image below:



Prior to version 4.5, the way to do this is by using the Connection Configuration Tool for the Citrix Presentation Server (no screenshots are available):

Open the "Citrix Connection Configuration" dialog box (Choose Programs > Citrix > Administration Tools > Citrix Connection Configuration Tool).

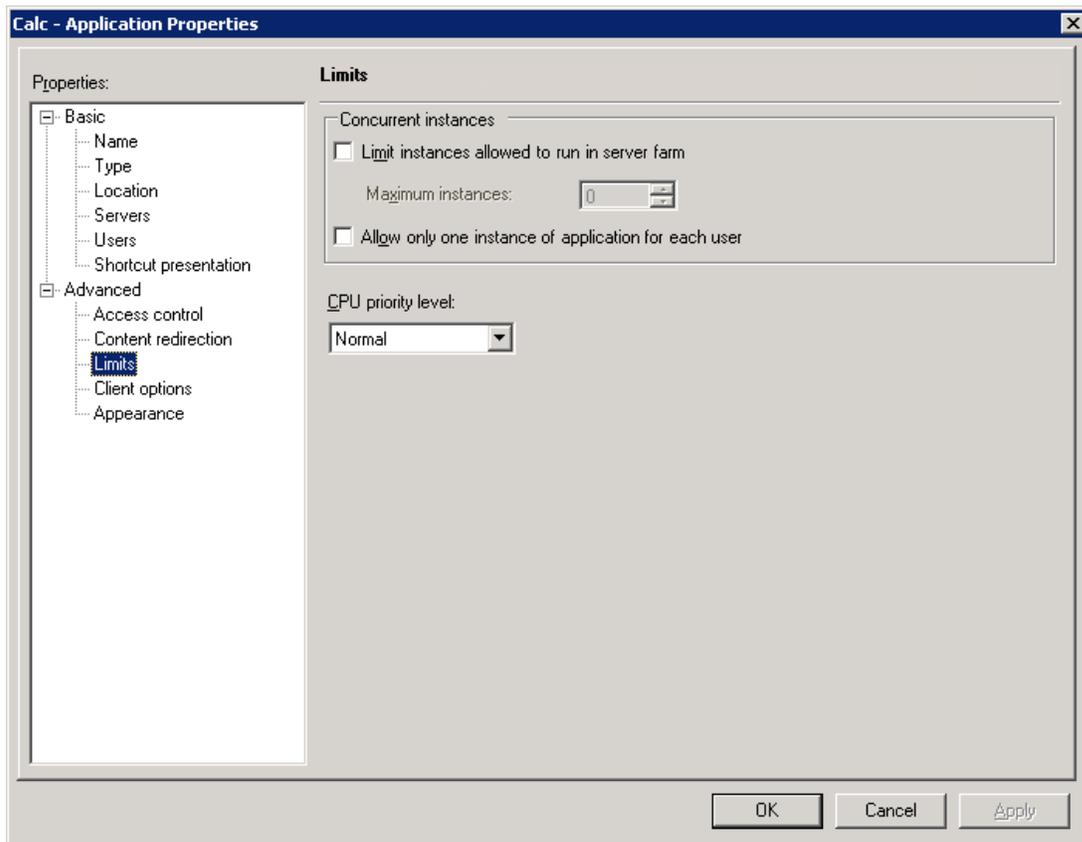
5. Select the ica-tcp connection name and choose Connection > Edit. Alternatively, double-click on the connection. The "Edit Connection" dialog box will open.
6. Click the Advanced button. The "Advanced Connection Settings" dialog box will open.
7. In the bottom section of the dialog box, clear the "inherit user config" check box adjacent to the "On a broken or timed-out connection" list box.
8. Change the entry for this list box to "reset", Click OK."

Allow Multiple User Instances

Many times, there is a limited amount of data to work with for a load test, and the number of login ids is restricted. In those cases, some logins may have to be reused, causing some

logins to be on the system multiple times concurrently. To ensure that this is allowed, the following modifications should be made in the Access Management Console on the Citrix Presentation Server:

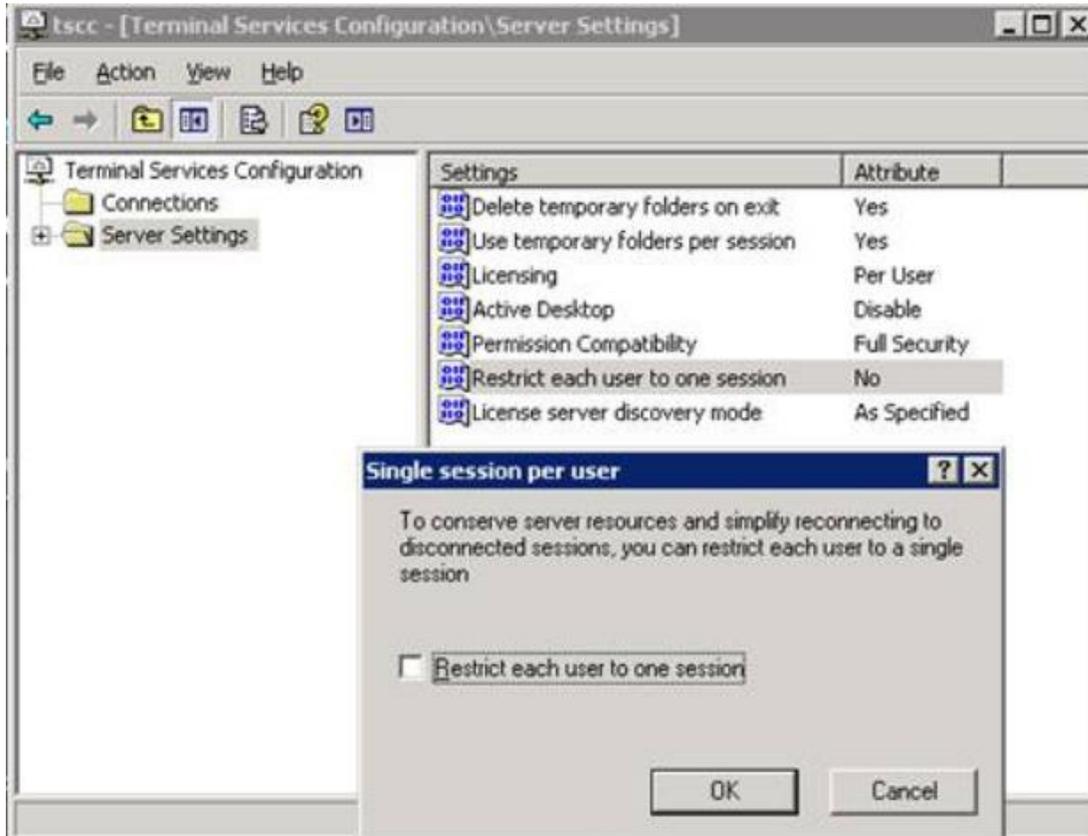
1. Expand the Citrix Resources/Presentation Server/<SERVER FARM NAME>/Applications options on the left panel of the Access Management Console and right-click on the application being tested.
2. Select **Properties**. The Application Properties dialog box is displayed.
3. Expand the Advanced section (if needed) and select the **Limits** option on the left panel. See the image below:



4. Ensure the following options in the **Limits** section are unchecked:
 - a. Limit instances allowed to run in the server farm.

- b. Allow only one instance of application for each user

An additional setting may need to be addressed in the Microsoft Terminal Services configuration. See the image below:

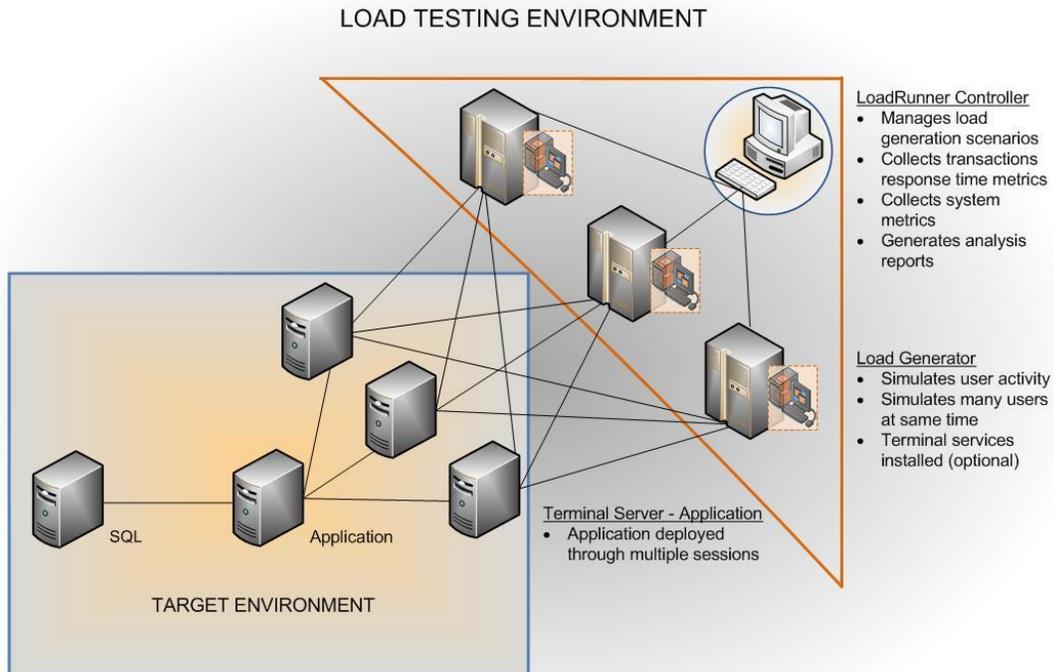


"Restrict Each User to one session", should be unselected (or "no" in the Attribute).

Using Terminal Services on Generators

If the OS supports it, Windows Terminal Services can be installed on Load Generators. This enables the use of multiple Terminal Sessions with a specified number of virtual users in each one. This can increase the scalability of the Generator, allowing many more virtual users to run on the machine.

This can become confusing for those configuring it for the first time, especially when the target application also uses Terminal Services as the application deployment method. The diagram below illustrates how a typical configuration could look:

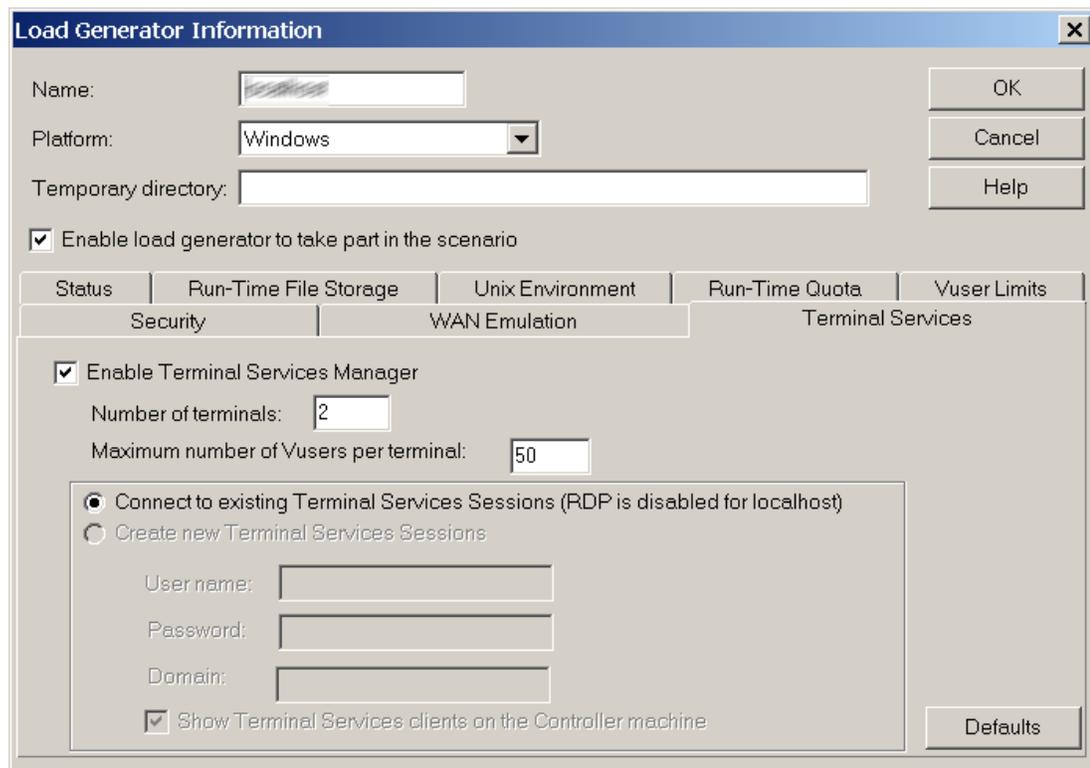


The target environment may be a two-tier client/server application where the Application server connects to a database back end. In the past, the application might have had a client portion of the application installed on individual workstations. Terminal Servers or Citrix servers are placed in front of the Application server to provide an alternative to installing the client on individual machines. The Terminal Server allows multiple users to connect and run the application in their own session/desktop space. The obvious benefit to this deployment is that it makes upgrades much easier by not having to update the client for each individual machine.

The LoadRunner lab depicted in the image above has the typical Controller and Load Generators, but in this case the Generators ALSO has Terminal Services installed. This

allows the Generator Agents to run a specified number of virtual users within multiple terminal sessions on the same machine, enabling more vusers overall to run on the machine and making the Generators much more scalable.

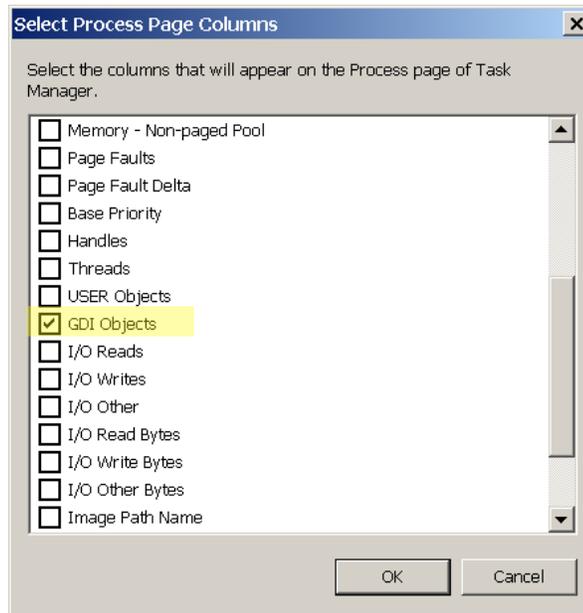
On the LoadRunner Controller, select the Load Generator dialog box and the Generator to configure. Select "Details", and click on the "Terminal Services" tab. Check the box for "Enable load generator to take part in the scenario". See the image below:



The default is two (2) Terminal Server Sessions (Number of terminals), and 50 Vusers per terminal session. The amounts entered into these fields depend on the hardware specifications of the Generator, the number of GDI Resources, and Memory that each session takes up for the application.

For Windows 2000, the GDI resource limitation is a little over 16,000 and in Windows 2003 it is approximately 65,000. However, the maximum number of GDI handles that can be

opened per session is usually lower, since it is also affected by available memory. Enable the “GDI Objects” column in Task Manager on the Generator to view how many objects are being used at any given time. In Task Manager, select “View” and “Select Columns...” from the menu options. Check the “GDI Objects” choice, and click OK.



Windows XP has a default limit of 10000. According to Microsoft, “GDI objects support only one handle per object. Handles to GDI objects are private to a process. That is, only the process that created the GDI object can use the object handle.” To change this limit, set the following registry value:

**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows\GDIPProcessHandleQuota**

IP Spoofing With Citrix

Some applications require a unique IP address, or restrict users to only one IP address. This can be for session pooling, security, or any number of reasons. The “IP Spoofing” feature in LoadRunner is not supported for the Citrix protocol. However, each Citrix session can be configured to use a different IP address by making configuration changes on the Citrix Presentation Server. For version 4.5, this information can be found in Chapter 4, “Using

HP LoadRunner® and Citrix® - Advanced Techniques for Performance Engineers

Virtual IP Addresses with Published Applications”. It requires registry changes on the server, so make sure the Citrix administrator is involved in these kinds of configuration changes.

Advanced Citrix® Scripting

Execute on Window

The `ctx_execute_on_window` function allows the script to handle windows that might appear anywhere along the execution of the script. This is especially useful with warning message or popup dialogue boxes. `ctx_execute_on_window` identifies a specific function to execute if a pre-identified window appears.

The syntax of this function is:

```
ctx_execute_on_window( char *window_name, long handler);
```

For example, if an application temporarily loses connectivity to the servers, it may have a popup window that allows the user to click on a button to reconnect to the server. By using the `ctx_execute_on_window` function, each time the application suddenly disconnects and the reconnect window appears, the script will automatically recognize this and perform the necessary steps (in this case, to tab to the reconnect button and select it). This will allow the script to reconnect at any time during the execution of the script, no matter if it is out of the original sequence scripted.

Example Vugen Code:

```
int reconnect(char win_title[]) {  
  
    // Note: the parameter win_title is not used in the handler  
    ctrx_key("TAB_KEY", 0); // tab through to "Connect" button  
    ctrx_key("ENTER_KEY", 0); // select the "Connect" button  
  
    return 0;  
}  
  
vuser_init () {  
  
    // If the "Disconnect" popup window appears then reconnect  
    ctrx_execute_on_window("Disconnected", reconnect);  
  
    ctrx_set_connect_opt(APPLICATION, "#My Application");  
    ctrx_set_connect_opt(NETWORK_PROTOCOL, "TCP/IP");  
    ctrx_connect_server("server", "username", "password", "domain", CTRX_LAST);  
    ctrx_wait_for_event("LOGON");  
    ctrx_set_window("Client info");  
    ctrx_type("This is a test");  
    lr_think_time(3);  
  
    // Navigate to a new client  
    ctrx_mouse_click(28, 38, LEFT_BUTTON, 0, "Client info");  
  
    return 0;  
}
```

Dynamic Synchronization on Bitmap Changes

Synchronization is very important when scripting against the Citrix protocol. Identifying when a bitmap changes allows the script to wait until an application process is finished before continuing to the next step of the script. The function which allows you to monitor when a bitmap value changes is: "ctrx_sync_on_bitmap_change".

The syntax of this function is:

```
ctrx_sync_on_bitmap_change ( x_start, y_start, width, ht, <extra_args>,  
[CONTINUE_ON_ERROR,] CTRX_LAST);
```

This can be very useful during the execution of a script to control synchronization. When executed, this function sets the value of the defined bitmap and waits for it to change. In some circumstances, it is possible for a bitmap to change before the call is made to wait. In that case, the script should set the value of the bitmap before initiating the command that will cause it to change. At that step in the script, insert the original bitmap value into the `ctrx_sync_on_bitmap_change` function. If the bitmap has already changed, the script will continue. If the bitmap has not changed, `ctrx_sync_on_bitmap_change` will wait until the bitmap changes. When including a predetermined bitmap value for `ctrx_sync_on_bitmap_change` to look for, ALL arguments for the function must be included. The syntax of the function when looking for a predefined bitmap value is:

```
ctrx_sync_on_bitmap_change(x_start, y_start, width, ht, initial wait time, timeout, bitmap value, CTRX_LAST);
```

Vugen Code Example:

```
char buffer[64];  
  
//Grab the bitmap value  
ctrx_get_bitmap_value(100, 125, 50, 100, buffer);  
  
//Click on save  
ctrx_mouse_click(573, 61, LEFT_BUTTON, 0, "* Microsoft Internet Explorer");  
  
//Wait for the screen to change  
ctrx_sync_on_bitmap_change(100, 125, 50, 100, 0, 10, buffer, CTRX_LAST);
```

In this example, the script retrieves the current value of the bitmap and saves it to the buffer. It uses that value in the `ctrx_sync_on_bitmap_change` function. A static value could have been used in the function. However, it is recommended to retrieve the bitmap value using this method because the value could change, and this makes the scripts more resilient to changes made to the application.

```
int ctrx_save_bitmap(long x_start, long y_start, long width, long height, const char
*file_name );
```

Error Handling

To obtain accurate test results, the script must correctly execute each step of the script. When a step returns something unexpected, the script must be able to handle this gracefully. Utilizing output codes returned by Citrix ICA functions, the scripter can write code that can react to change, and log unforeseen inconsistencies in the business process (i.e. the user could not login). Error handling is a valuable tool for debugging, and ensures proper test execution. Every function in LoadRunner returns an output code. To access the output code for a function in the script, declare a variable and set it equal to the function in question. Once the output code has been obtained, use script logic to perform specific activities.

Vugen Code Example:

```
int rc;

//Connect to the application
ctrx_set_connect_opt(APPLICATION, "MY APPLICATION");
ctrx_set_connect_opt(NETWORK_PROTOCOL, "TCP/IP");
ctrx_connect_server("server", "username", "password", "domain");
ctrx_wait_for_event("LOGON");

//log in screen appears
ctrx_set_window("My Application Login");

//input username
ctrx_mouse_click(34, 325, LEFT_BUTTON, 0, "My Application Login", CTRX_LAST);
ctrx_type("username", CTRX_LAST);

//input password and press enter
ctrx_mouse_click(573, 61, LEFT_BUTTON, 0, "My Application Login", CTRX_LAST);
ctrx_type("password", CTRX_LAST);
ctrx_key("ENTER_KEY", 0, CTRX_LAST);

//Synch on the welcome page
rc = ctrx_sync_on_bitmap(137, 211, 51, 17, "18d962dcbf373e948e83489a19419512", CONTINUE_ON_ERROR, CTRX_LAST);

//Ensure that the user is logged in to before continuing
if (rc!=0) {
    lr_message("Application login is successful");
}

else {
    lr_error_message("User failed to lon in");
    lr_abort();
}

return 0;
```

In the example above the variable `rc` is declared as an integer and set equal to a Citrix sync on bitmap. This ensures that the Vuser has reached the login page. By adding logic, the script will let only successfully logged in users continue. Users that failed to login will abort the script and send an error message to the output window with text stating why the user aborted. Notice the `CONTINUE_ON_ERROR` argument in the `ctrx_sync_on_bitmap` function. Without this argument, the script will error out and stop running if the sync is not successful. This would make the script logic void.

Change Timeout

Some applications have steps in their business processes which are expected to take longer (i.e. Streaming Video or a Smart Client download). During execution, these steps can timeout and fail to synchronize. To avoid step failure, manually adjust the waiting time. This can be configured two ways, with different results for each. In the Vugen run-time settings, the wait time can be set by selecting “timing” under the Citrix section. By increasing the “waiting time” property, it will increase the amount of time that the script will attempt to complete each step of the script (i.e. Sync on a bitmap or open a new window).


```
int rc;

//Connect to the application
ctrx_set_connect_opt(APPLICATION, "#My Application");
ctrx_set_connect_opt(NETWORK_PROTOCOL, "TCP/IP");
ctrx_connect_server("server", "username", "password", "domain");
ctrx_wait_for_event("LOGON");

//log in screen appears
ctrx_set_window("My Application Login");

// input username
ctrx_mouse_click(34, 325, LEFT_BUTTON, 0, "My Application Login", CTRX_LAST );
ctrx_type("username",CTRX_LAST);

// input password and press enter
ctrx_mouse_click(573, 61, LEFT_BUTTON, 0, "My Application Login", CTRX_LAST );
ctrx_type("password",CTRX_LAST);
ctrx_key("ENTER_KEY", 0, CTRX_LAST );

// Synch on the welcome page
rc = ctrx_sync_on_bitmap(137, 211, 51, 17, "18d962dcbf373e948e83489a19419512", CONTINUE_ON_ERROR, CTRX_LAST);

// Ensure that the user is logged in to before continuing
if (rc=0) {
    lr_message("Application login is successful");
}

else {
    lr_error_message("User failed to log in");
    lr_abort();
}
```

In the example above, the script saves the current wait time settings to a variable called **waitTime**. The script will allow 120 seconds for the application window to open. Once the window is open (after the `ctrx_set_window` function), another `ctrx_set_waiting_time` is executed, and the script will set the waiting time back to the original setting.

Remotely Logout Hanging Sessions

While debugging your script or abruptly stopping and restarting, some of the sessions may exist within the Citrix server in an unknown state (known as “hung sessions”). As explained earlier in this document, a timeout can be set in the terminal services manager options to reset hanging sessions and cause them to expire. However, it may not be possible for the performance engineer to make these configuration changes. There is another way: Create a batch file and use the command prompt window to reset them. This can be a lot faster when debugging Vugen scripts.

There are two commands that allow a user to manipulate Terminal Server users:

- **QWINSTA** (Query Windows Station)
- **RWINSTA** (Reset Windows Station)

To perform either of these commands on a remote host administrator privileges on the remote machine are required.

The batch file will need a separate command for every server you wish to terminate all ica sessions on. Each separate command should be a single line in the batch file. See the template below.

Template:

```
for /f "tokens=2" %%i in ('qwinsta /server:<Hostname or IP > ^| find /i "ica-tcp#") do echo y | rwinsta /server:<Hostname or IP> %%i
```

Example:

If you have 3 Presentation servers named citrixserver1, citrixserver2 and citrixserver3 your batch file should look like the following:

```
for /f "tokens=2" %%i in ('qwinsta /server:citrixserver1 ^| find /i "ica-tcp#") do echo y | rwinsta /server:citrixserver1 %%i
for /f "tokens=2" %%i in ('qwinsta /server:citrixserver2 ^| find /i "ica-tcp#") do echo y | rwinsta /server:citrixserver2 %%i
for /f "tokens=2" %%i in ('qwinsta /server:citrixserver3 ^| find /i "ica-tcp#") do echo y | rwinsta /server:citrixserver3 %%i
```

For further information on the QWINSTA and RWINSTA you can reference there command line help files below.

QWINSTA:

QINSTA [<SessionName> | <UserName> | <SessionID>] [/server:<ServerName>] [/mode]
[/flow] [/connect] [/counter]

<SessionName>	Specifies the name of the session that you want to query.
<UserName>	Specifies the name of the user whose sessions you want to query.
<SessionID>	Specifies the ID of the session that you want to query.
/server:<ServerName>	Identifies the terminal server to query. The default is the current server.
/mode	Displays current line settings.
/flow	Displays current flow-control settings.
/connect	Displays current connect settings.
/counter	Displays current counters information, including the total number of sessions created, disconnected, and reconnected.
/?	Displays help at the command prompt.

RWINSTA:

RWINSTA {<SessionName> | <SessionID>} [/server:<ServerName>] [/v]

<SessionName>	Specifies the name of the session that you want to reset. To determine the name of the session, use the query session command.
<SessionID>	Specifies the ID of the session to reset.
/server:<ServerName>	Specifies the terminal server containing the session that you want to reset. Otherwise, the current terminal server is used.
/v	Displays information about the actions being performed.
/?	Displays help at the command prompt.

```
for /f "tokens=2" %%i in ('qwinsta ^| find /i "ica-tcp#") do echo y | rwinsta  
/server:<Hostname or IP> %%i
```

If the credentials that you need to logon to the remote machine with are different than those are currently logged in, execute a NET USE command to create a new connection to the remote machine with the proper credentials. The NET USE command connects a computer to or disconnects a computer from a shared resource, or displays information

about computer connections. The command also controls persistent net connections. Used without parameters, net use retrieves a list of network connections.

Syntax Examples for NET USE:

```
net use [{DeviceName | *}] [\\ComputerName\ShareName[\volume]] [{Password | *}]  
[/user:[DomainName\]UserName]  [/user:[DottedDomainName\]UserName]  [/user:  
[UserName@DottedDomainName] [/savecred] [/smartcard] [{/delete | /persistent:{yes  
| no}}
```

```
net use [DeviceName [/home[{Password | *}] [/delete:{yes | no}}]
```

```
net use [/persistent:{yes | no}]
```

Summary

Using LoadRunner to execute performance testing against Citrix deployed applications requires the performance engineer be knowledgeable about Citrix server settings and how they affect the behaviors of virtual users. Some configuration settings may need to be changed during the debugging phase of scripting, and changed again when actual test execution begins. Citrix administrators will need to work closely with the performance engineer to ensure the configuration settings are verified in all phases of the test project.

Vugen scripting against the Citrix protocol means paying attention to screen synchronizations and visual clues within the business process and using code logic to control the process to account for unexpected behavior. Graceful error handling is crucial. Performance engineers need to understand the business side of the application as well as the technical challenges of the Citrix protocol within Vugen.

This paper illustrates some of the complexities involved in planning and configuration setup, as well as the overall skill sets required to test successfully. Northway helps companies ensure their applications deployed via Citrix are scalable, and meet the performance objectives required for service level agreements. Our world-class consultants have the in-depth knowledge of both LoadRunner and Citrix required to get reliable test results every time.

About Northway Solutions Group

Northway Solutions Group is a technology solution provider that provides consulting services around HP's Business Technology Optimization (BTO) software products. The company is a HP Elite Partner and reseller for HP Software products in the Application Lifecycle Management (ALM) space, which includes Project and Portfolio Management, Quality Center, Performance Center, Application Security Center, and Business Availability Center. Northway employs certified consultants with real-world experience who provide long-term solutions to the toughest business challenges. This includes providing training and implementation services for all the products in our portfolio. Visit www.northwaysolutions.com to find out more information about our services and offerings.

About HP Software

HP is a technology solutions provider to consumers, businesses and institutions. HP software helps IT organizations make sure that every dollar delivers positive business outcomes. Their business technology optimization (BTO) products automate key processes across IT functions, and their OpenCall platforms enable people to communicate and access services. As the focus of IT shifts from systems management to business results, companies are investing in HP software to lower costs, improve delivery time and reduce risk.

LoadRunner® is a registered trademark of HP Software. All rights reserved.

About Citrix®

Citrix desktop virtualization, application virtualization, server virtualization, application networking, cloud computing and Software as a Service offerings radically simplify computing for millions of users, delivering applications as an on-demand service to any user, in any location on any device.

Citrix customers include the world's largest Internet companies, 99 percent of Fortune Global 500 enterprises, and hundreds of thousands of small businesses and prosumers worldwide. Citrix partners with over 10,000 companies worldwide in more than 100 countries. Founded in 1989, annual revenue in 2008 was \$1.6 billion.

Citrix Presentation Server™ is a registered trademark of Citrix® software. All rights reserved.