Micro Focus®
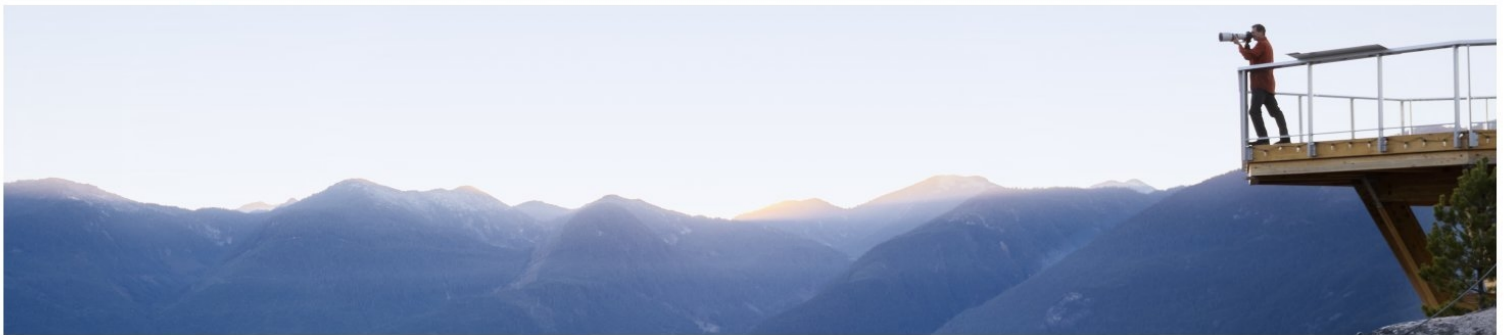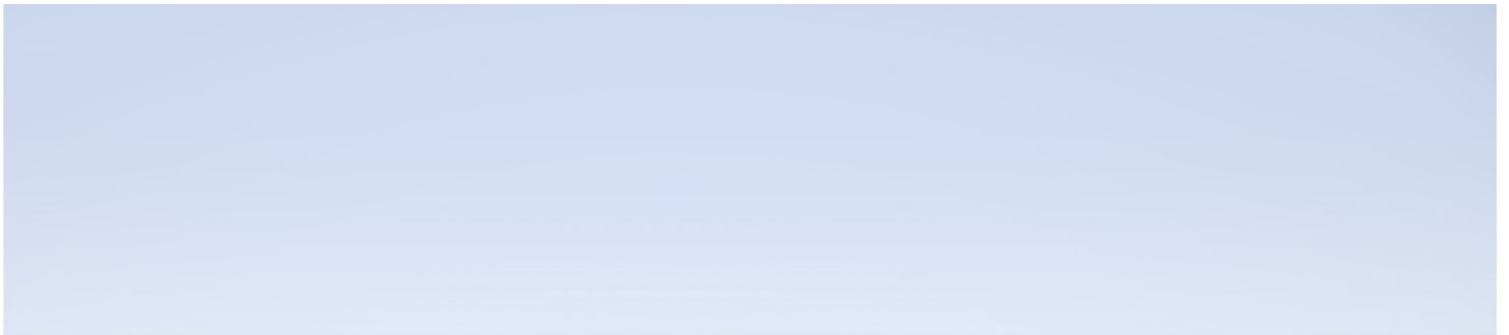
Diagnostics
Version 9.51, Released November 2018

LoadRunner and Performance Center - Diagnostics Integration Guide
Published November 2018

Legal Notices

Disclaimer

Certain versions of software and/or documents ("Material") accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. As of September 1, 2017, the Material is now offered by Micro Focus, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Java is a registered trademark of Oracle and/or its affiliates.

Oracle® is a registered trademark of Oracle and/or its affiliates.

Acknowledgements

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes software developed by the Spice Group (http://spice.codehaus.org).

For information about open source and third-party license agreements, see the *Open Source and Third-Party Software License Agreements* document.

Contents

Welcome To This Guide

Welcome to the LoadRunner/Peformance Center-Diagnostics Integration Guide. This guide describes how to set up and verify an integration of Micro Focus Diagnostics with Micro Focus LoadRunner or Performance Center.

Chapter 1: Introduction

Diagnostics integrates with LoadRunner and Performance Center to provide QA teams the power of load testing with the added advantage of developer-ready reporting that facilitates collaboration across silos.

This chapter provides the following sections:

- "How You can Use Diagnostics with LoadRunner" below
- "How You Can Use Diagnostics with Performance Center" on the next page

How You can Use Diagnostics with LoadRunner

LoadRunner diagnostics modules and the integration with Diagnostics provide detailed performance information in LoadRunner to help you rapidly identify and pinpoint performance problems in Siebel, Oracle, SAP, J2EE and .NET environments. During a load test, you can drill down to Diagnostics data for the whole scenario or for a particular transaction. After you have run your scenario, you can use LoadRunner Analysis to analyze offline Diagnostics data generated during the scenario.

Once you set up LoadRunner integration with Diagnostics you can view Diagnostics data in LoadRunner in a number of ways:

- "Diagnostics UI Views for a LoadRunner Scenario" below
- "Drill Down to the Diagnostics UI for Details on a Transaction" on the next page
- "LoadRunner Analysis J2EE and .NET Diagnostics Graphs" on the next page

Diagnostics UI Views for a LoadRunner Scenario

For an active LoadRunner load test scenario, you can open the Diagnostics UI from LoadRunner and get detailed performance data for the whole scenario. Select the Diagnostics for J2EE/.NET tab at the bottom of the scenario window and the Diagnostics UI opens displaying diagnostic data for the scenario:
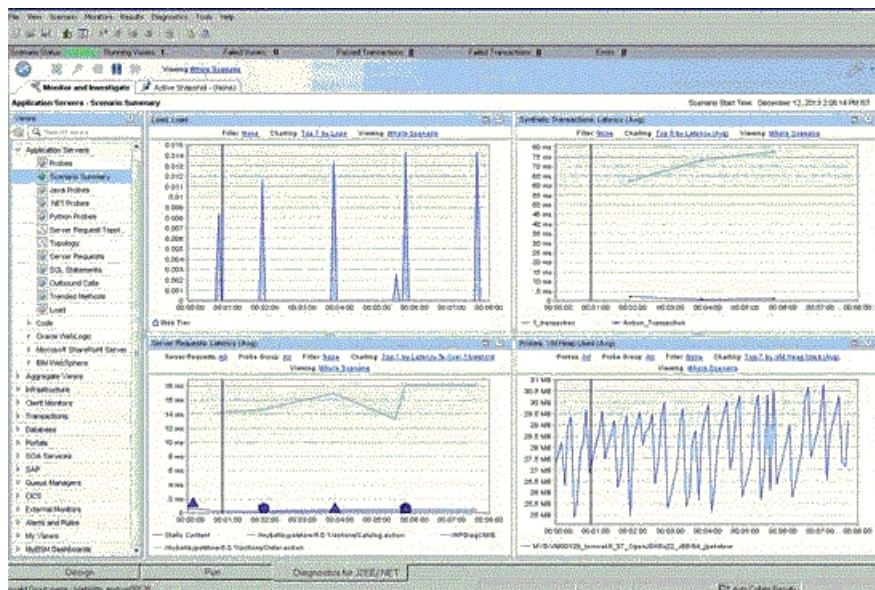
Once you are in the Diagnostics UI you can navigate to other views to identify, isolate, analyze, and solve performance problems detected during the run.

Drill Down to the Diagnostics UI for Details on a Transaction

In LoadRunner you can drill down to the Diagnostics UI for a particular transaction to get diagnostics data for that transaction.

In LoadRunner select one of the Transactions graphs (such as Transaction Response Time) and from the graph you drill down to the Diagnostics UI which opens displaying the Transactions view. From here you can navigate to other views in the Diagnostics UI to troubleshoot problems related to the transaction.
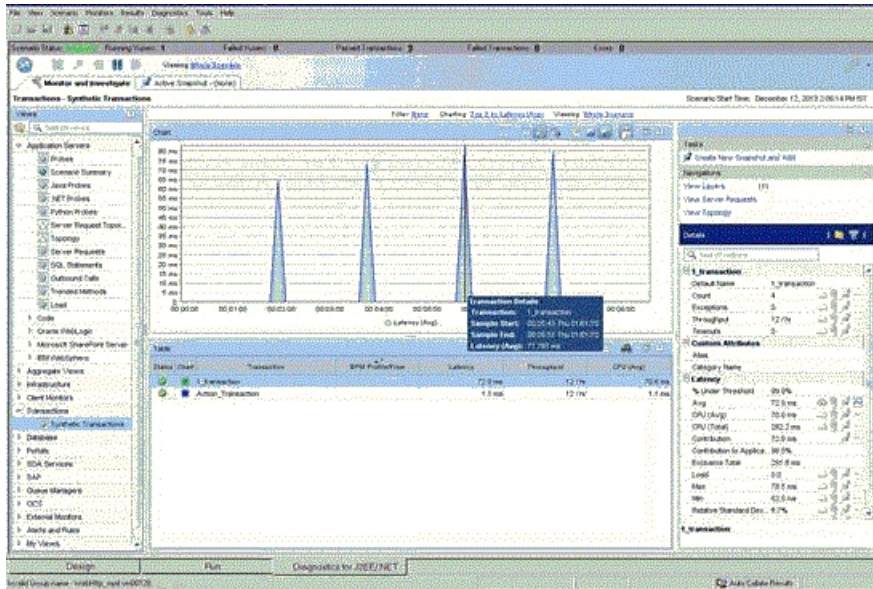
LoadRunner Analysis J2EE and .NET Diagnostics Graphs

In LoadRunner Analysis the **J2EE and .NET Diagnostics graphs** are based on data provided by Diagnostics. These graphs enable you to trace, time and troubleshoot individual transactions and server requests through J2EE and .NET web, application and database servers. you can also quickly pinpoint problem servlets and JDBC calls to maximize business process performance, scalability and efficiency.

The J2EE & .NET Diagnostics graphs in LoadRunner are comprised of two groups: J2EE & .NET Diagnostics Graphs, J2EE & .NET Server Diagnostics Graphs.

Refer to the LoadRunner Controller User Guide and the LoadRunner Analysis User Guide for information on viewing Diagnostics data in LoadRunner.

# How You Can Use Diagnostics with Performance Center

Performance Center diagnostics modules and the integration with Diagnostics provide detailed performance information in Performance Center to help you rapidly identify and pinpoint performance problems in Siebel, Oracle, SAP, J2EE and .NET environments.

In Performance Center, the **J2EE/.NET diagnostics** functionality is provided by Diagnostics allowing you to monitor, analyze and solve complex performance problems in your J2EE and .NET application test environments.

Once you set up Performance Center integration with Diagnostics you can view Diagnostics data from Performance Center.

For a Performance Center load test run you can drill down into the Diagnostics UI and get detailed performance data for the whole load test or for a particular transaction. After the load test run, you can use LoadRunner Analysis to analyze offline diagnostics data generated during the load test.
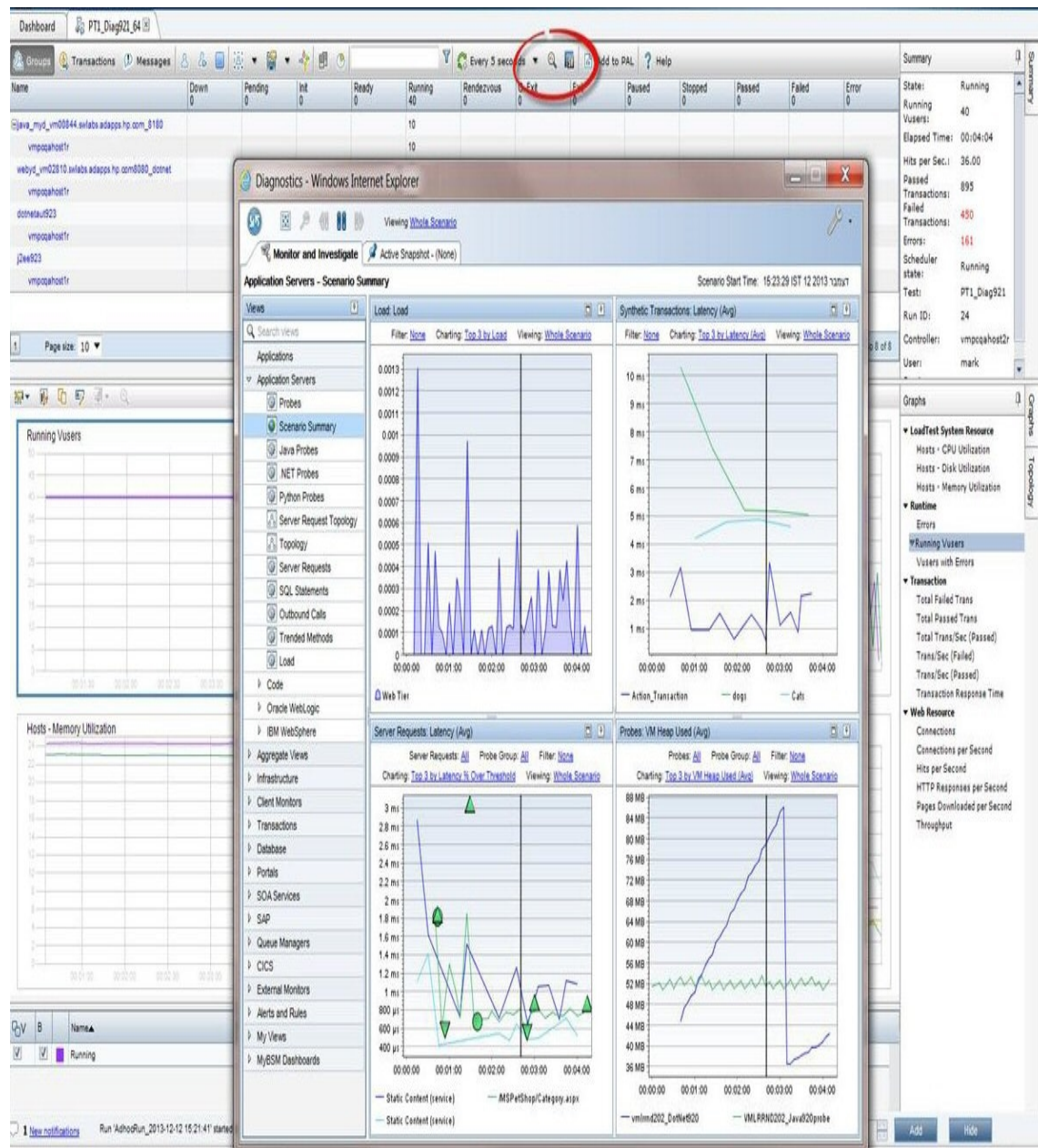
Chapter 2: Setting Up LoadRunner to Use Diagnostics

General information is provided on setting up LoadRunner and Diagnostics integration in load testing runs and offline analysis.

This chapter includes:

Diagnostics-LoadRunner Integration Overview

The high-level procedure for setting up an integration is as follows:

1. Identify the Diagnostics commander server and the LoadRunner installation to be integrated.

   The Diagnostics components have special configuration and licensing requirements when they are to be integrated with LoadRunner. See the Diagnostics Server Installation and Administration Guide and LoadRunner Installation Guide for more information about licensing and setting up the Diagnostics components.

   If a firewall separates the Diagnostics commander server and the LoadRunner components, special configuration is required. See "Configuring for a Firewall Environment" on page 17.

2. Install the LoadRunner Diagnostics Add-in on the LoadRunner Controller host machine.

3. Configure LoadRunner to access the Diagnostics Server. See the LoadRunner Controller User's Guide for details.

4. Configuring LoadRunner Scenarios to use Diagnostics. See the LoadRunner Controller User's Guide for details.

5. Select the Probe Metrics to Include in the Offline Analysis File. See "Selecting Probe Metrics to Include in the Offline Analysis File" on page 10.

Installing the LoadRunner Diagnostics Add-in

The LoadRunner Diagnostics add-in uses a small bootstrapper to dynamically download most of the software required from the Diagnostics server when it is first executed. If Diagnostics is updated, the new Diagnostics files should be picked up automatically at the next LoadRunner execution.

The Diagnostics 9.10 LoadRunner add-in is for Diagnostics versions 8.0x, 9.0x, 9.10 and later.

To install the LoadRunner Diagnostics Add-in:

1. Download the LoadRunner Diagnostics Add-In file from the Software Support web site (https://softwaresupport.softwaregrp.com/).

2. Log on to the LoadRunner Controller host machine.

   If LoadRunner is already running, close the Controller and main LoadRunner window.

3. Run **setup.exe** from the LR_AddIn directory of the Diagnostics installation disk. The setup installation program is launched.

4. The software license agreement is displayed. Read the agreement and click **Yes** to accept it.

   In the displayed Registration Information dialog box, type your name, the name of your company, and your LoadRunner maintenance number. You can find the maintenance number in the maintenance pack shipped with LoadRunner.

   Click **Next** to start the installation process. The installation process begins.

5. When the installation process is complete, the installation wizard displays a confirmation message.

   Click **Finish** to complete the installation process.

   In certain cases when related LoadRunner processes are running on your computer (such as the LoadRunner Agent), you will be required to restart your computer to complete the LoadRunner Add-in installation process.

> **Note:** No uninstall utility is provided for the LoadRunner Diagnostics add-in.

> **Caution:** The time and time-zone settings of the host machines for the Diagnostics and LoadRunner components must be consistent. You will encounter time-difference problems if the time is not properly set.

Configuring LoadRunner Scenarios to use Diagnostics

Each time you want to capture Diagnostics metrics in a load test scenario, you must configure the Diagnostics parameters for the scenario and select the probes that will be included in the scenario. You configure your scenario for Diagnostics from the LoadRunner Controller. See the *LoadRunner Controller User Guide* for details.

> **Note:** If you saved a scenario with the Diagnostics settings already configured, you do not need to reconfigure the Diagnostics parameters each time you run that scenario.

To send the ELT events from LoadRunner to Diagnostics server, set the following in the **etc/webserver.properties** file:

```
lr.trustAll=true
lr.elt.whitelist=<hostname or ip>,<hostname or ip>
```

This property adds trusted hosts that can send ELT events from LoadRunner. Values must be in a comma separated string with the hostname or IP address.

Selecting Probe Metrics to Include in the Offline Analysis File

Diagnostics probe metric data can be included or used in LoadRunner Offline Analysis.

By default, only *HeapUsed, GC Collections/sec* and *GC time Spent in Collections* metric data is included in offline Analysis. To select additional probe metrics to be included in offline Analysis (.eve file), you use the Diagnostics configuration file, **etc/offline.xml**.

Configure the offline.xml files on the Diagnostics mediators to specify the Diagnostics probe metrics that you want to be included in the offline analysis. You would configure this file for all mediators that have probes participating in runs.

Probe metrics are only available from the Java probe, not the .NET probe.

> **Note:** Make sure that the clocks are synchronized between the Diagnostic servers and LoadRunner.

A list of all the possible Diagnostics metrics can be found in **metrics.config** in the Diagnostics probe install directory. This file contains all metrics, although some of these metrics might not be available for offline analysis, depending on the platform (Java probe only), application server type, and version being monitored.

In general, all the metrics that you can see under a Java probe in the Diagnostics UI Details pane can be used in the offline.xml file for the relevant mediator. The following collectors that are included in the **metrics.config** file cannot be used in the offline.xml file: "system" and "Mercury System".

An example of the **offline.xml** file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<probeMetrics xmlns="http://mycompany.com/diagnostics/offline/1.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="offline.xsd">
  <metric>
    <name>HeapUsed</name>
  </metric>
  <metric>
    <name>GC Collections/sec</name>
  </metric>
  <metric>
    <name>GC Time Spent in Collections</name>
  </metric>
</probeMetrics>
```

The <metric> element specifies a match condition. Matching is possible on metric name (<name>), category (<category>) and collector (<collector>). The <name>, <category> and <collector> elements can be combined, in which case all elements must match.

Matching examples:

Match and include a metric named "HeapUsed"

```
<metric>
```

```
    <name>HeapUsed</name>
</metric>
```

Match and include any metric that the "JVM" collector exposes.

```
 <metric>
<collector>JVM</collector>
</metric>
```

Match and include a metric named "HeapFree" that the "JVM" collector exposes.

```
<metric>
   <name>HeapFree</name>
   <collector>JVM</collector>
</metric>
```

By default, the matching is performed on a substring, meaning the specified text between <name>, <category> and <collector> needs to be within (or contain) the actual metric name, category or collector. For example, specifying "HeapFree" text for <name> would match any metric that has "HeapFree" in its name (for example "MyHeapFreeMetric", "YourHeapFreeMetric").

It is further possible to specify a regular expression for <name>, <category> and <collector> via the match attribute on <metric> (for example <metric match="regex">). Note, regular expressions are expensive and will impact the time it takes to write out .eve files.

Specify sending all probe metrics to LoadRunner as follows:

```
<metric match="regex">
   <collector>.*</collector>
   </metric>
```

Note: Adding lots of metrics in offline.xml will increase the size of the offline (.eve) file, which in turn impacts the time it takes to analyze the offline file in the LoadRunner Analysis application.

Changes to the **offline.xml** file are automatically detected and applied every 15 seconds. Configuration errors (validated against **offline.xsd**) are logged to the **server.log** file.

Troubleshooting and Tuning

Improving Transfer of Large Offline Analysis Files

The offline analysis files (.eve) generated by Diagnostics during a LoadRunner or Performance Center test run can get quite large. At the end of the runs these files are transferred from the Diagnostics servers to the LoadRunner/Performance Center controller for collation and analysis. You can improve the transfer time and load time of the offline analysis files that include Diagnostics data by lowering the resolution of the .eve files.

Use the **bucket.lr.offline.duration** property and the **bucket.lr.offline.sr.duration** properties in the **server.properties** file on the Diagnostics server to increase the aggregation period (for example, from 5s to

15s). These properties enable you to define how many five second trend points are to be aggregated together to produce a single sample for offline analysis.

Out of Memory Issue in LoadRunner Controller's Diagnostics UI

If you find OutOfMemory errors in the LoadRunner UI log (Mercury_Diagnostics_UI.log) this may be caused by a memory limit imposed on the Diagnostics applet displaying data in the Diagnostics tab of the LoadRunner Controller.

To fix this you can increase the heap memory of the Diagnostics applet by defining the system environment variables in the OS of the LoadRunner system: APPCRITIC_MAX_MEM=256m and JAVA_TOOL_OPTIONS=-Xmx256m (in case 256MB Max VM Heap is also needed).

Offline Data Transfer from Diagnostics to the LoadRunner Controller Fails

If the **Secure Host Communication** option is turned on between the LoadRunner Controller and the **Remote LoadRunner LoadGenerator**, the offline data transfer from Diagnostics to the LoadRunner Controller fails.

Also, when Diagnostics runs in secure mode, the SSL mode Diagnostics Commander integration to the LoadRunner Controller is not supported.

Authentication and Authorization for LoadRunner Users

When you set up both LoadRunner for integration with Diagnostics, you specify the Diagnostics Server details within LoadRunner. These details include the username and password with which you log on to Diagnostics.

When you access Diagnostics from LoadRunner, you are logged into Diagnostics with that same username and password that you specified during the integration setup.

Users accessing Diagnostics from within LoadRunner, will therefore have the privileges that are associated with the username that was specified during the integration setup.

Chapter 3: Setting Up Performance Center to Use Diagnostics

General information is provided on configuring Performance Center to enable Diagnostics for use in a load test.

This chapter includes:

- "Diagnostics-Performance Center Integration Overview" below
- "Configuring Performance Center Load Tests to Use Diagnostics" on the next page
- "Managing Performance Center Offline Files" on the next page
- "Troubleshooting" on page 15

Diagnostics-Performance Center Integration Overview

Performance Center and Diagnostics are integrated products that are designed to work together to provide information to help you understand and improve the performance of your applications.

To make Diagnostics accessible from Performance Center,the following Diagnostics Server details are specified in Performance Center.

- **Server Name.** The name of the machine that is host to the Diagnostics commander server.
- **Port Number.** The port number used by the Diagnostics commander server. The default port number is **2006**.
- **Login Name.** The user name with which you log on to Diagnostics. The default user name is `admin`.

  The user name that you specify should have **view**, **change** and **execute** privileges. For more information about user privileges, see "Understanding User Privileges" in the Diagnostics Server Installation and Administration Guide.
- **Password.** The password with which you log on to Diagnostics. The default password is `admin`.
- **Communication**. The communication protocol with which Performance Center accesses the Diagnostics Server.

If **HTTPS** is the communication protocol, additional configuration steps are required. For more information about the steps required, see "Enabling HTTPS Between Components" in the Diagnostics Server Installation and Administration Guide.

If a firewall separates the Diagnostics commander server and the Performance Center components, special configuration is required. See "Configuring for a Firewall Environment" on page 17.

> **Note:** You only need to specify these details the first time you use Performance Center with Diagnostics. You provide this information on the Diagnostics page of the Performance Center Administration Site.

Configuring Performance Center Load Tests to Use Diagnostics

Each time you want to capture Diagnostics metrics in a load test, you must configure the Diagnostics parameters for the load test and select the probes that will be included in the load test.

For complete instructions on how to configure Performance Center to integrate with Diagnostics see the *Performance Center User's Guide* section about Diagnostics integration with Performance Center.

If there is a firewall between the Performance Center Controller and the Diagnostics Server involved in a load test, you must configure the Controller and the Diagnostics Server to use the MI Listener utility to enable the transfer of the offline analysis file. Also you must specify the IP address of the MI Listener machine in Performance Center.

And you must configure the Diagnostics Server in Mediator mode so that it can work across a firewall. See "Configuring for a Firewall Environment" on page 17.

The benefit of enabling the "Monitor server requests" functionality in the integration is that calls into a back-end VM can be captured even in the case where:

- the probe is not capturing RMI calls.
- RMI calls cannot be captured (perhaps because an unsupported application container is being used).
- the application uses some other mechanism for communications between multiple VMs.

> **Note:** If you configure the integration to monitor server requests this functionality imposes an additional overhead on the probe.

To investigate any issues that you have with the connections between the Diagnostics components, use the System Health Monitor accessible from Performance Center.

Managing Performance Center Offline Files

Performance Center offline files are kept by default. To manage offline files, you must configure the Diagnostics Servers in Mediator mode so that they delete these files.

You do this by setting the property **distributor.offlinedelivery.preserveFiles** to true in **<diag_server_ install_dir>/etc/server.properties**. When set to true, this property causes the run-specific "offline" files stored in the server's data directory to be retained for the amount of time specified in the **facade.run_delete_ delay property** in the server's **webserver.properties** file (default period is 5 days).

During this retention period, the run can be successfully collated. Sometime after the retention period has ended, the associated offline files will be deleted from the system.

Authentication and Authorization for Performance Center Users

When you set up both Performance Center for integration with Diagnostics, you specify the Diagnostics Server details within Performance Center. These details include the username and password with which you

log on to Diagnostics.

When you access Diagnostics from Performance Center, you are logged into Diagnostics with that same username and password that you specified during the integration setup.

Users accessing Diagnostics from within Performance Center, will therefore have the privileges that are associated with the username that was specified during the integration setup.

Troubleshooting

Some deployments have problems in collating the J2EE results data through a firewall and using the MI Listener.

Problem

The Diagnostics Commander server is located inside the company firewall and in the same network as the PC servers and Host Controller. Load Generators and the Diagnostics Mediator are located outside the firewall and communication to these machines is made through an MI Listener. All the machines are running Windows.

The TCP ports 2612, 2006 and 35000-35100 are open on the firewall to enable communication through the firewall between the Diagnostics Commander, the Mediator and Probes. This is as described in "LoadRunner and Performance Center Communication" on page 19.

The PC Controller should collate the J2EE results at the end of a load test by communicating with the Diagnostics Mediator through the MI Listener. The Mediator must be configured work with the MI Listener as described in "Configuring a Mediator Server to Work Across a Firewall" on page 20.

The PC loadtest is then configured to access J2EE Diagnostics and to collate over a firewall. While the load test runs the J2EE Diagnostics data is viewable from the PC load test page. The problem is the J2EE results are not transferred from the Mediator to the Controller when the load test ends and the results collation is started.

At the PC Event Log level the error " Collating Results is Stuck" may be displayed. After some time, possibly be up to 30 minutes, the results.zip may be available in PC but a subsequent Analysis report on these results will show there is Load Generator data but no J2EE data.

Cause

The collation failure in this specific case is caused by the configuration of Probes that are referenced in the PC loadtest Diagnostics, and are installed on both the Diagnostics Commander and the Mediator. In this case the Diagnostics Commander is also performing a Mediator role. The collation of J2EE results then fails as the MI Listener is only aware of one Mediator while the PC Collate process expects to communicate with two Mediators.

Fix

There are a number of possible workarounds to this problem:

- Manually Transfer the J2EE results file

  The J2EE results can be manually transferred from the Mediator to the PC Controller and then the results re-collated. This can be used in an emergency to allow an urgent Analysis of load test data where the J2EE data is critical. See https://softwaresupport.softwaregrp.com/km/KM1281953 for details.

- The Firewall Workaround

  a. On the Diagnostics Mediator machine go to **<diag_server_install_dir>\nanny\windows\bin** and run the **AgentConfig.exe**.

b. Check that the Enable Firewall Agent is clear (disabled).

c. Restart the Diagnostics Server to ensure the new magent runs.

d. On the firewall ensure that TCP Port 23472 is open to allow communication to this port from the IP address of the PC Controller(s).

e. In the PC loadtest design, open the Diagnostics page, then open the **J2EE/.NET Diagnostics Configure…** tab.

f. Ensure the **There is a firewall between the mediator and the Controller** option is clear (disabled).

g. Save the load test, run it and test the J2EE Collation.

- The MI Listener Workaround

If for network security or administration reasons it is not possible to open TCP port 23472 on the firewall then the following workaround can be used.

a. On the Diagnostics Commander machine go to **<diag_server_install_dir>\nanny\windows\bin** and run the **AgentConfig.exe**.

b. Ensure that the Enable Firewall Agent is set. Even though this machine is actually inside the firewall you want it to use the MI Listener channel.

c. In Settings configure the MI Listener hostname or IP address and for the Local Machine Key enter the hostname of the Diagnostics Commander.

d. Restart the agent and verify it connects to the MI Listener.

This workaround ensures the MI Listener has two Mediators connected to it when the PC J2EE collate process runs over a firewall, and therefore the request for J2EE data from probes running on both the Mediator and on the Commander is satisfied.

Note the MI Listener workaround is also effective for PC runs that do not collate over a firewall. In this case the PC collate process will use TCP port 23472 which remains open and in LISTENING state on the Diagnostic Mediator and Commander even when the agent has been configured to run over a firewall.

Problem

Offline data transfer from Diagnostics to the Performance Center Controller fails.

Cause

If the **Secure Host Communication** option is turned on between the Performance Center Controller and the **Remote LoadRunner LoadGenerator**, the offline data transfer from Diagnostics to the Performance Center Controller fails.

Also, when Diagnostics runs in secure mode, the SSL mode Diagnostics Commander integration to the Performance Center Controller is not supported.

# Chapter 4: Configuring for a Firewall Environment

General information is provided on configuring Performance Center and Load Runner communication over a firewall.

This chapter includes:

Communications with LoadRunner and Performance Center

The following diagram shows the default ports used by the Diagnostics and LoadRunner/Performance Center components.
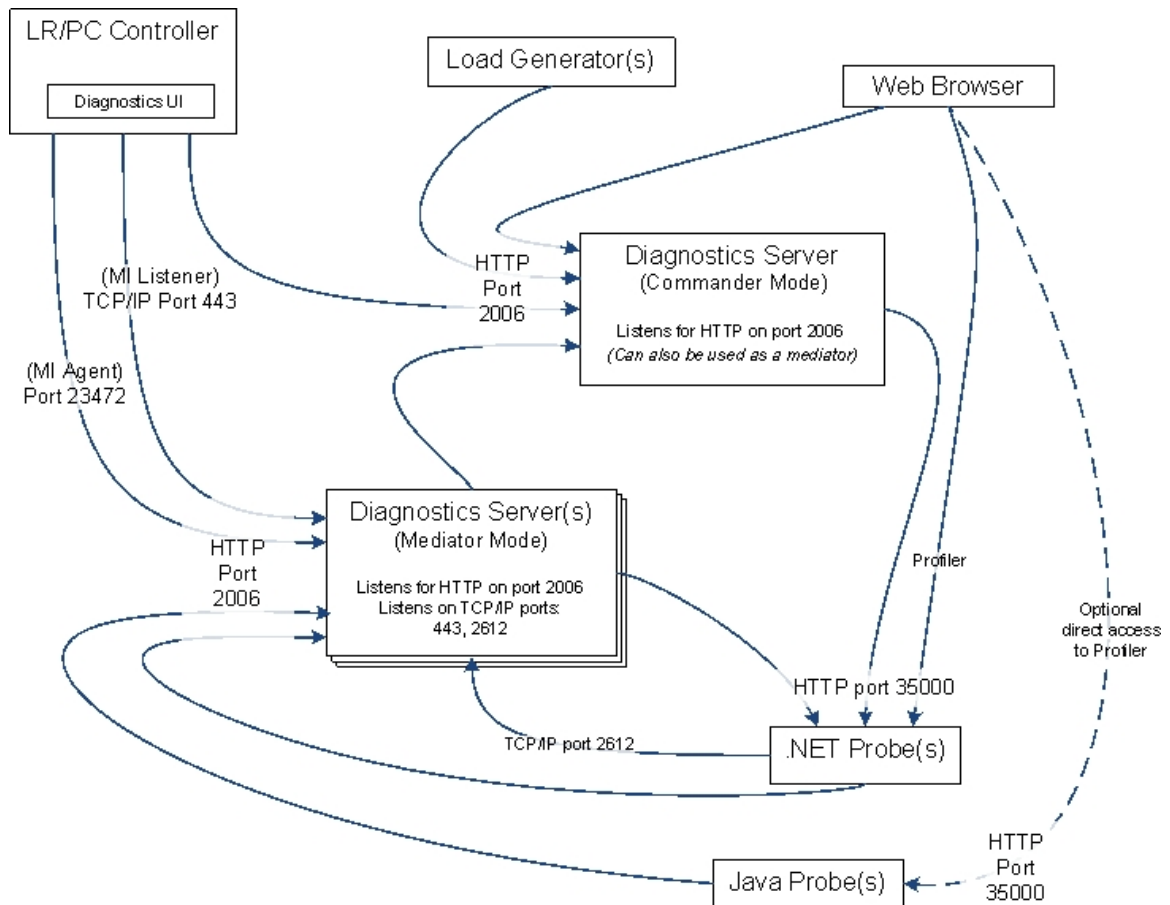
LoadRunner and Performance Center Communication

The diagram below shows a typical Diagnostics topology where a firewall separates the probe from the other Diagnostics and LoadRunner components.
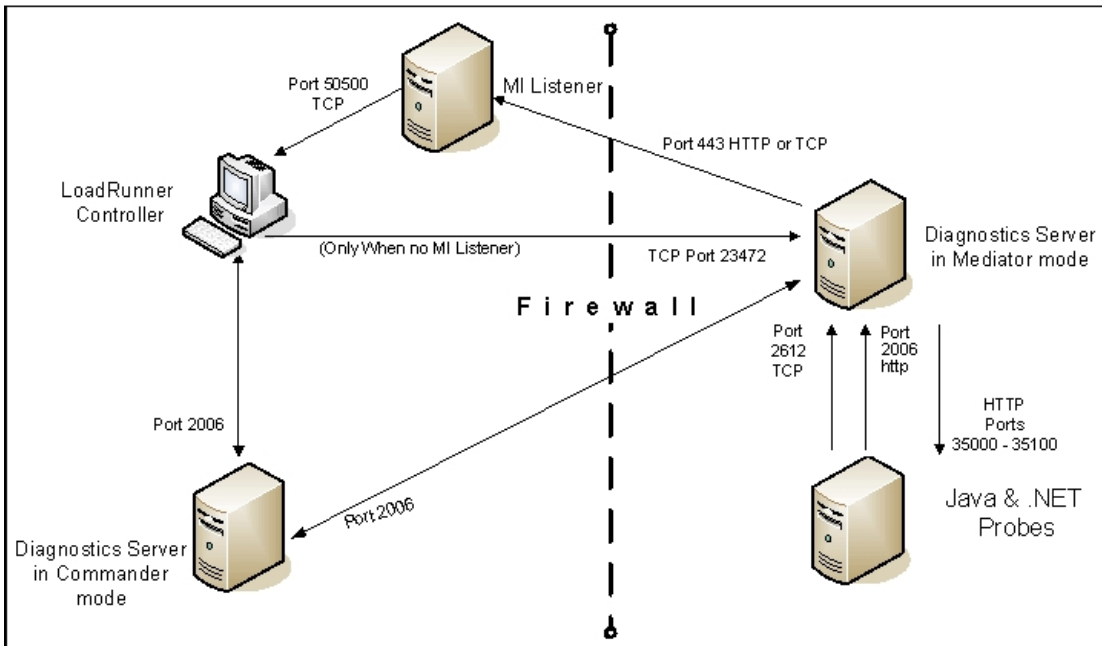
Note: LoadRunner is used in this diagram for illustrative purposes. The same information would apply to Performance Center.

You must configure the firewall to allow the Diagnostics components to communicate with each other.

To change the port used from the default (443), repeat the following steps for all three agents: Controller Machine, Load Generator Machine (the Diagnostics Mediator) and the MI Listener.

On the Controller and MI Listener <LaunchServiceDir> is the <LR_install_dir)\launch_service directory. On the Diagnostics Mediator <LaunchServiceDir> is the <diag_server_install_dir>\nanny\<platform>.

1. Go to the <LaunchServiceDir>\dat\mdrv.dat [launcher] section and add "OFWPort=<port>" (where port is the port number you want to open).

2. Go to <LaunchServiceDir>\dat\channel_configure.dat [General] section and add "OFWPort=<port>" (where port is the port number you want to open).

3. Restart the agent or mediator.

To configure the firewall to enable the communications between the Diagnostics components, open the ports that will allow:

- HTTP requests from the Diagnostics mediator server to the Diagnostics commander server on port 2006.

- TCP requests from the probe to the Diagnostics mediator server on port 2612.

- HTTP requests from the probe to the Diagnostics mediator server on port 2006.

Note: In addition to the above topology, if you are using the LoadRunner Analysis Tool to view offline J2EE results, see "Collating Offline Analysis Files over a Firewall" below to properly configure the Controller and the Diagnostics Servers in Mediator mode for offline file retrieval.

Collating Offline Analysis Files over a Firewall

During a LoadRunner / Performance Center load test, the Diagnostics Servers that have probes reporting to them generate an offline analysis file on their host machine. The offline analysis files is retrieved by LoadRunner / Performance Center when it collates the results of a load test.

If there is a firewall between the LoadRunner / Performance Center Controller and the Diagnostics Server involved in a load test, you must configure the Controller and the Diagnostics Server to use the MI Listener utility to enable the transfer of the offline analysis file. The MI Listener utility comes with LoadRunner / Performance Center and should be installed on a machine inside your firewall as shown in the following diagram.
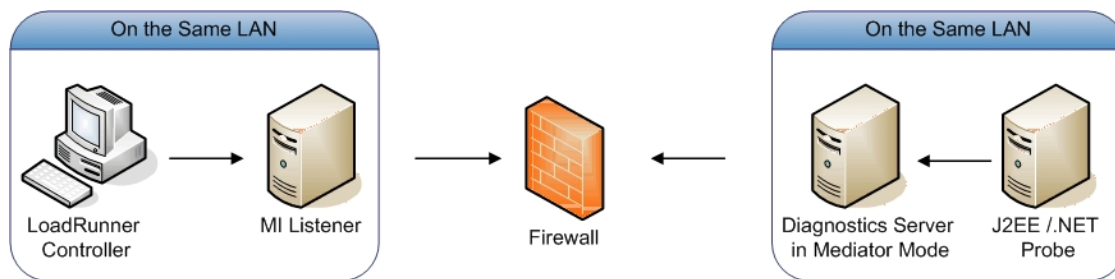
To configure the Controller to access Diagnostics Servers that are behind a firewall see the following sections:

- "Installing and Configuring the MI Listener" below.
- "Configuring a Mediator Server to Work Across a Firewall" below.
- "Configuring LoadRunner and Performance Center to Work with Diagnostics Firewalls" on page 24.

Installing and Configuring the MI Listener

The MI Listener component is the same component that is used to serve Load Generators that are outside of a firewall. For more information about how to configure the MI Listener for LoadRunner, see the *LoadRunner Controller User Guide*. For more information about how to configure the MI Listener for Performance Center, see the *Performance Center Administrator's Guide*.

Configuring a Mediator Server to Work Across a Firewall

To configure the Diagnostics mediator server so that it can work across a firewall, you must complete the following additional configuration steps. If you did not install and configure the Diagnostics mediator server you must do so before attempting these steps. For instructions on installing the Diagnostics mediator server, see the Diagnostics Server Installation and Administration Guide.
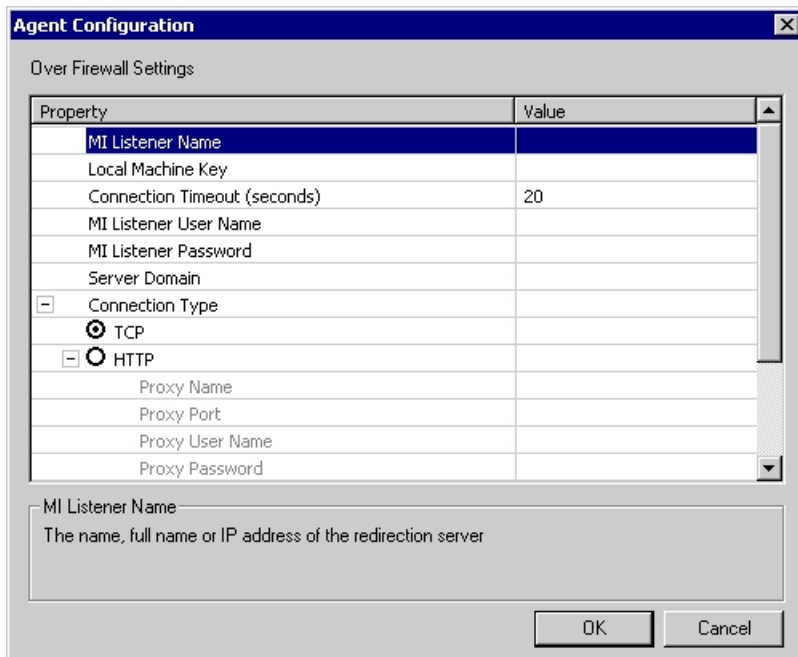
To configure the Diagnostics mediator server for a firewall on a Windows machine:

1. Launch the Agent Configuration by running **<diag_server_install_ dir>/nanny/windows/bin/AgentsConfig.exe**.

   The Agent Configuration dialog box opens:
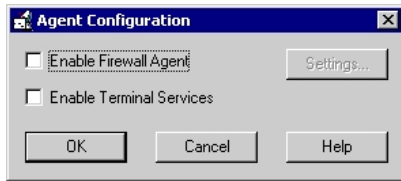
2. Select **Enable Firewall Agent**. The **Settings** button becomes enabled.

3. Click **Settings**. The Agent Configuration process opens the Agent Configuration Settings dialog box.

4. In Value column of the MI Listener Name property, enter the host name or IP address of the machine where the MI Listener was installed.

5. For the **Local Machine Key** property, enter the machine name of the host of the Diagnostics mediator server.

   > **Note:** When entering the host name of a Diagnostics component you must use the fully qualified host name, that is, the machine name and the domain name.
   >
   > Use the System Health view in the Diagnostics UI to determine the machine name for the host of the Diagnostics mediator server. For more information on the System views, see the Diagnostics Server Installation and Administration Guide.
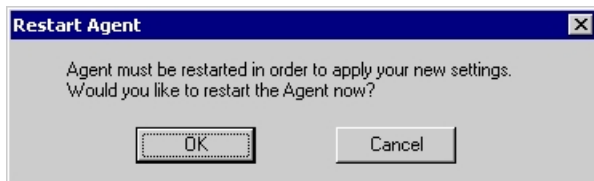
6. Click **OK** to close the dialog box.
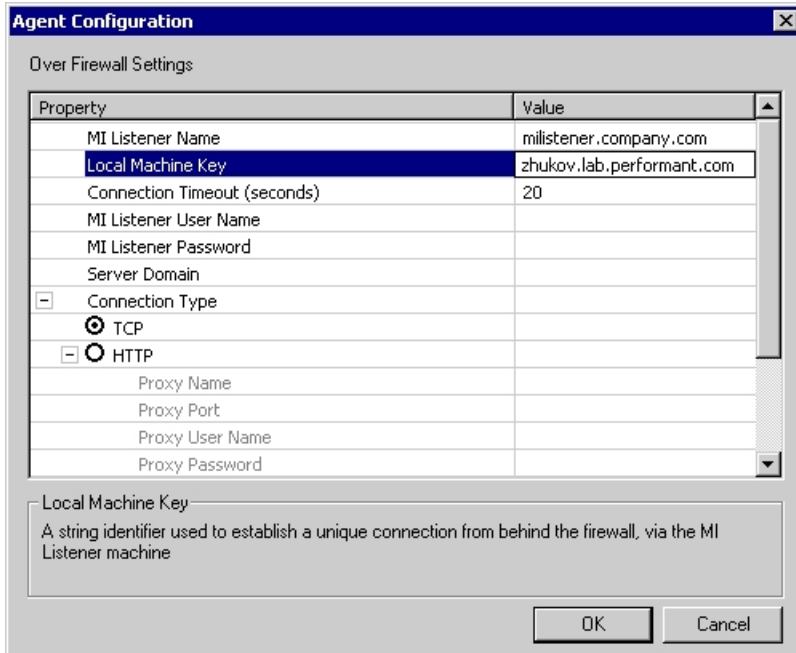
7. Click **OK** again to close the Agent Configuration dialog box.
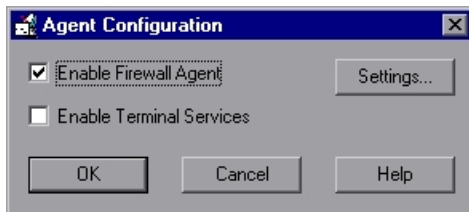
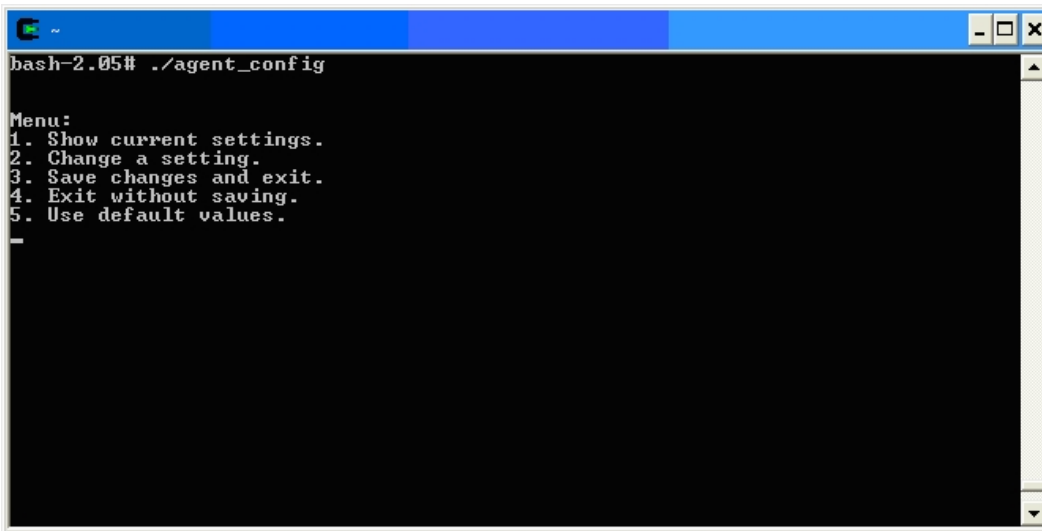8. The Restart Agent dialog box opens. Click **OK** to restart the Agent.

To configure the Diagnostics mediator server for a firewall on a UNIX/Linux machine:

1. Modify the **<diag_server_install_dir>/nanny/<platform>/dat/br_lnch_server.cfg** file.

   Change the value of the **FireWallServiceActive** property to 1.

2. Run the following commands to launch the Agent Configuration utility.

   For Solaris and Linux, where <platform> is either solaris or linux:

   ```
   export LD_LIBRARY_PATH=.
   export M_LROOT=<diag_server_install_dir>/nanny/<platform>
   cd $M_LROOT/bin
   ./agent_config
   ```
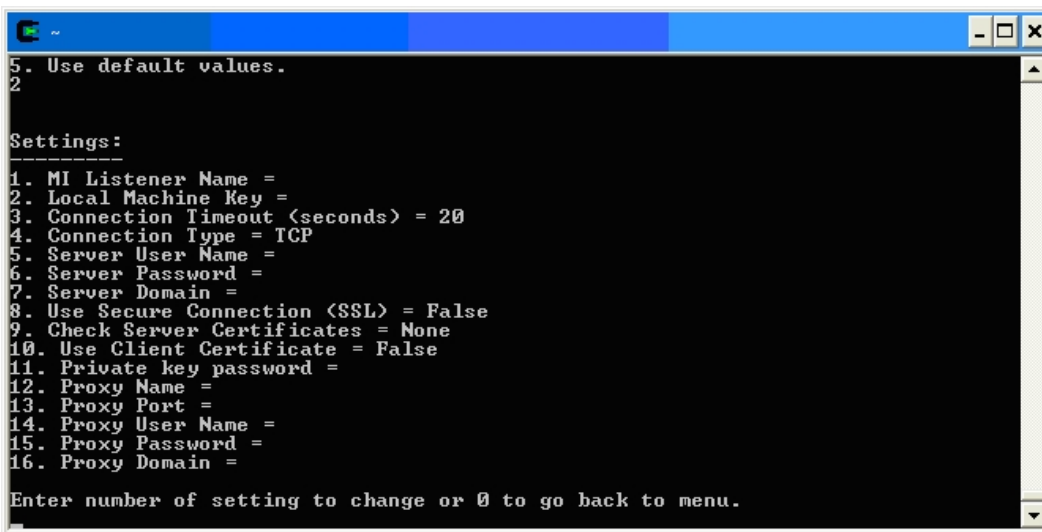
3. In the Agent Configuration Utility window, type **2** to select **Change a Setting**.

```
bash-2.05# ./agent_config

Menu:
1. Show current settings.
2. Change a setting.
3. Save changes and exit.
4. Exit without saving.
5. Use default values.
_
```

4. A list of settings appears.

Type **1** to select **MI Listener Name**, and enter the machine name or IP address of the MI Listener host.
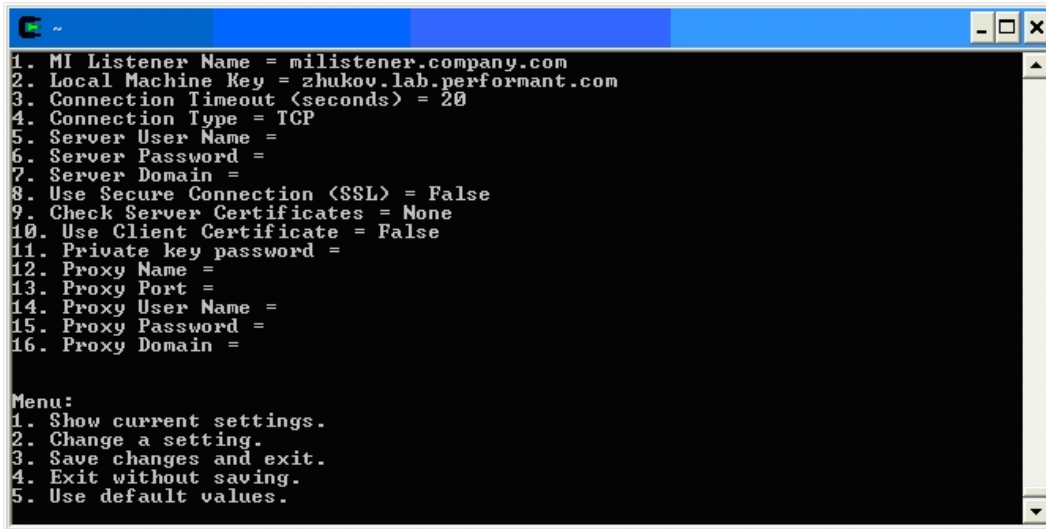
```
5. Use default values.
2


Settings:
--------
1. MI Listener Name =
2. Local Machine Key =
3. Connection Timeout (seconds) = 20
4. Connection Type = TCP
5. Server User Name =
6. Server Password =
7. Server Domain =
8. Use Secure Connection (SSL) = False
9. Check Server Certificates = None
10. Use Client Certificate = False
11. Private key password =
12. Proxy Name =
13. Proxy Port =
14. Proxy User Name =
15. Proxy Password =
16. Proxy Domain =

Enter number of setting to change or 0 to go back to menu.
_
```

5. Type **2** to select **Change a Setting** and enter the machine name of the host of the Diagnostics mediator server.

   Use the System Health view in the Diagnostics UI to determine the machine name for the host of the Diagnostics mediator server. For more information on the System views, see the Diagnostics Server Installation and Administration Guide.

6. Type **3** to select **Save changes and exit**, to complete the updates.

7. Restart the Diagnostics mediator server.

   ./m_daemon_setup -remove (this stops the server and MI Agent)

   ./m_daemon_setup -install (this starts the server and MI Agent)

   On some Linux systems, if you encounter an error saying that the libstdc++.so.5 shared library is missing, you may need to install it. For example, on CentOS, enter the following command to install the library:

   yum install compat-libstdc++-33

Configuring LoadRunner and Performance Center to Work with Diagnostics Firewalls

After the MI Listener has been installed and your Mediator machines are configured, you must update the Diagnostics Configuration in LoadRunner / Performance Center so that the application knows to use the MI Listener when it is transferring the offline data from a Mediator that is outside of a firewall.

For Performance Center:

Make sure that you specified the IP address of the MI Listener machine that is configured to collect application diagnostics data from over the firewall. For details, see the section on MI Listeners in the *Performance Center Administrator's Guide*. Also make sure that Diagnostics is enabled in Performance Center. For details, see the section on Diagnostics in the *Performance Center User Guide*.

For LoadRunner:

Make sure that Diagnostics is enabled in LoadRunner. When you configure your load test scenario to work with Diagnostics, make sure that you select the option to work over a firewall and specify the name of the

relevant MI Listener server. For details, see the section on Diagnostics in the *LoadRunner Controller User Guide*.

Chapter 5: Supported Protocols When Integrating with LoadRunner or Performance Center

The synthetic transactions defined in the VU scripts recorded via the following protocols can be captured by Diagnostics when LoadRunner or Performance Center is integrated with Diagnostics.

Protocols for HTTP - HyperText Transfer Protocol

- AJAX (Click and Script)
- Ajax TruClient on Firefox – Supported in LR/PC 11.51 and above
- Ajax TruClient on Internet Explorer (IE) - Supported in LR/PC 12 and above
- Java Over HTTP
- Oracle Web Applications 11i
- Peoplesoft Enterprise
- SAP (Click and Script)
- SAP - Web
- Siebel - Web
- SOAP
- Web (Click and Script)
- Web (HTTP/HTML)
- Web Services

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on LoadRunner and Performance Center - Diagnostics Integration Guide (Diagnostics 9.51)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to docs.feedback@microfocus.com.

We appreciate your feedback!