

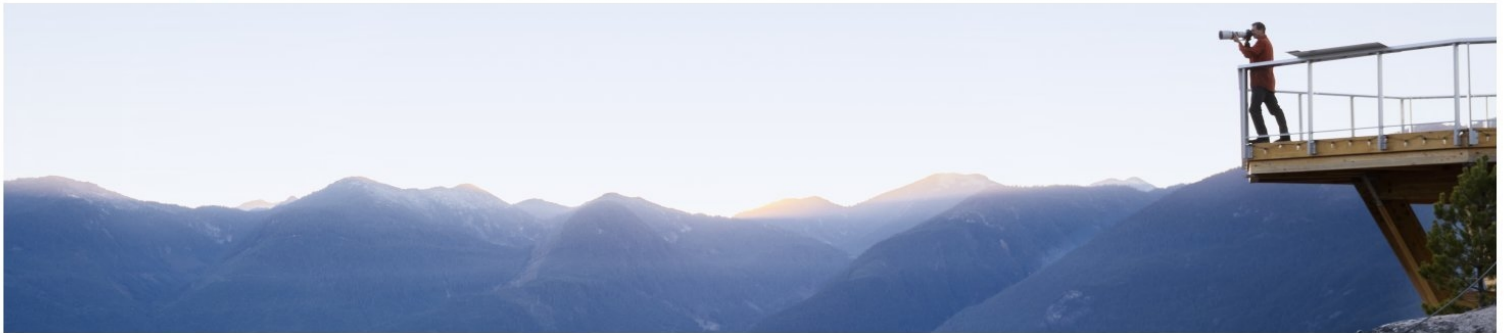
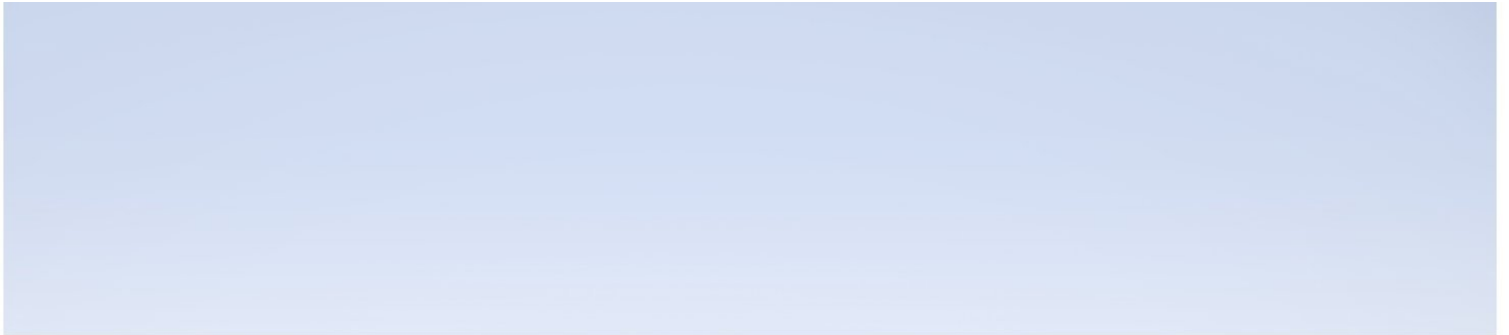


Diagnostics

Version 9.51, Released November 2018

Server Installation and Administration 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

© Copyright 2005 - 2018 Micro Focus or one of its affiliates

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	7
How This Guide Is Organized	7
Diagnostics Documentation	7
Introduction	9
Product Overview	11
Diagnostics Components and Data Flow	11
What does Diagnostics Monitor and Collect?	12
Probes: Agent and Collector Instances	16
Licenses	17
About Diagnostics Licensing	18
Types of Licenses	18
Licensing the Diagnostics Server in Commander Mode	19
View License Information	21
Installation	24
Installation Overview	25
Pre-installation Checklist	27
Download the Installer	28
Steps to Install on Windows	29
Steps to Install on Linux	33
Verify the Installation	37
Verify the Installation	37
Start and Stop the Diagnostics Server	37
Determining the Version of the Diagnostics Server	39
Silent Installation	40
Upgrade	42
Upgrade Overview	43
Upgrade Checklist	44
Upgrade on Windows	45
Upgrade on Linux	48
Configuration	51
Diagnostics Server Configuration	52
Configure for HTTP Proxy and Firewalls	52
Configure Diagnostics Servers for Proxy Communication	53
Configure Agents and Collectors for Proxy Communication	54
Configure for a Firewall Environment	57
Configure Secured Communication	59
Synchronize Time Between Diagnostics Components	60

Configure the Diagnostics Mediator Server for a Large Deployment	62
Optimize the Diagnostics Server in Production to Handle More Probes	66
Override the Default Diagnostics Server Host Name	66
Change the Default Diagnostics Server Port	67
Migrate Diagnostics Server from One Host to Another	67
Configure the Diagnostics Server for Multi-Homed Environments	68
Reduce the Diagnostics Server Memory Usage	70
Configure URI Trimming on the Server	71
Configure Server Request Name Based Trimming	71
Automate the Composite Application Discovery in Diagnostics	71
Prepare a High Availability Diagnostics Server	76
Probe Registration Auto-Assignment for Large Deployments	77
Configure Diagnostics for ServiceGuard (HA solution)	84
Diagnostics Server Assignments (LoadRunner/Performance Center Runs)	85
Configure the Diagnostics Server for LoadRunner Offline Analysis File Size	86
Configure Diagnostics Using the Diagnostics Server Configuration Pages	87
Configure a Custom Context Root	88
Configure Diagnostics for PCF	88
Overview	88
Obtain the Installation files	88
Start the Diagnostics Server Using PCF (Commander Mode)	89
Start the Diagnostics Server Using PCF (Mediator Mode)	89
How to start Diagnostics agent in PCF	89
Known Issues	90
Administration	91
Diagnostics Administration UI	92
Accessing the Diagnostics Administration UI	92
Using the Diagnostics Administration UI	93
Diagnostics Local Client	98
Support Matrix	98
Launch Diagnostics UI using the Diagnostics Local Client	99
Access secure connections (HTTPS) of Diagnostics using the Diagnostics local client ...	99
Access Diagnostics UI from APM local client	99
Configure Proxy	99
Limitations	100
User Authentication and Authorization	101
About User Authentication and Authorization	101
Understanding User Privileges	102
Understanding Roles	102
Accessing Diagnostics Using Default User Names	103
Understanding the Diagnostics Server Permissions Page	103

Creating, Editing and Deleting Users	108
Assigning Privileges Across the Diagnostics Deployment	109
Assigning Privileges for Probe Groups	110
Tracking User Administration Activity	111
List of Active Users	112
Configuring Diagnostics to use JAAS	112
Enable HTTPS Between Components	124
About Configuring HTTPS Communications	124
Filter Encryption Cipher Suites	124
HTTPS Checklist per Diagnostics Component	125
Enable Incoming HTTPS Communication for Diagnostics Components	126
Generate Client Certificate	126
Enable Outgoing HTTPS Communication from Diagnostics Components	132
Enabling HTTPS Communication for APM Integrations	136
Configure Diagnostics Commander to Connect to a BSM/APM Server That Requires a Client Certificate	137
How to use System Views for Administrators	139
System Views for Diagnostics' Administrators	139
System Health View Description	140
System Capacity View Description	141
Diagnostics Data Management	142
About Diagnostics Data	142
Custom View Data	142
Performance History Data	143
Data Retention	146
Disk Space Issues on the Server	149
Back Up Diagnostics Data	149
To handle Diagnostics Data when Upgrading Diagnostics	152
General Reference Information	153
UNIX Commands	153
Regular Expressions	153
Multi-Lingual User Interface Support	158
Data Exporting	159
Task 1: Prepare the target database	159
Task 2: Determine which metrics you want to export	160
Task 3: Determine the frequency and the recovery period	162
Task 4: Modify the data export configuration file for summary data	163
Task 5: Modify the data export configuration file for trend data (1 minute data granularity)	166
Task 6: Monitor the data export operation	169
Task 7: Verify the results	170
Task 8: Select the data from the target database	170

Sample Queries	171
Uninstallation	173
Uninstall on Windows	174
Uninstall on Linux	175
Troubleshoot	176
Diagnostics Installers Do Not Work on Linux	176
Java Agent Fails to Operate Properly	176
Error During WAS Startup with Diagnostics Profiler for Java	176
Missing Server-Side Transactions	176
Event Capture Buffer Full Warning	177
WebSphere Application Server Startup Issue	177
Java Agent Support Collector	178
File Permissions on Linux	178
Appendix A: Library Packages Required on Linux	179
Appendix B: Manual Installation of OM Agent and IAPA Components	182
Send Documentation Feedback	183

Welcome To This Guide

Welcome to the Micro Focus Diagnostics Server Installation and Administration guide. This guide describes how to install the Diagnostics Servers and how to plan for and maintain the Diagnostics deployment environment.

How This Guide Is Organized

This guide contains the following parts:

- Part 1: "[Introduction](#)" on page 9
Provides information of Product Overview to plan for the installation and configuration of the Diagnostics components.
- Part 2: "[Licenses](#)" on page 17
Provides detailed information about the Diagnostics licenses.
- Part 3: "[Installation](#)" on page 24
Describes how to install the Diagnostics Servers.
- Part 4: "[Upgrade](#)" on page 42
Describes how to upgrade the Diagnostics components.
- Part 5: "[Configuration](#)" on page 51
Describes advanced configuration of the Diagnostics servers.
- Part 6: "[Administration](#)" on page 91
Describes administrative tasks for the Diagnostics Administrator including using the Admin UI to configure and manage Diagnostics settings, managing users, permissions, authorization and authentication, enabling HTTPS secure communications between components, using System Health UI, managing data as well as doing backup and recovery, upgrading Diagnostics and installing patch updates, and using the Data Export feature.
- Part 7: "[Uninstallation](#)" on page 173
Describes how to uninstall the Diagnostics servers.
- Part 8: "[Troubleshoot](#)" on page 176
Describes how to troubleshoot the Diagnostics components.

Diagnostics Documentation

Diagnostics includes the following documentation. Unless specified otherwise, the guides are in PDF format only and are available from the [Software Support web site](https://softwaresupport.softwaregrp.com/) (https://softwaresupport.softwaregrp.com/).

- **Diagnostics User Guide and Online Help:** Explains how to choose and interpret the Diagnostics views in the Diagnostics Enterprise UI to analyze your monitored applications. To access the online help for Diagnostics, choose **Help > Help** in the Diagnostics Enterprise UI. If Diagnostics is integrated with another Micro Focus Software product the online help is also available through that product's Help menu. The User Guide is a PDF version of the online help and their content is identical. The User Guide is available from the Diagnostics online help Home page, from the Windows Start menu (open **User Guide**), or from the Diagnostics Server installation directory.

- **Diagnostics Server Installation and Administration Guide:** Explains how to plan a Diagnostics deployment, and how to install and maintain a Diagnostics Server.

The following Agent guides contain content that supports agent installation, setup and configuration.

- **Diagnostics Java Agent Guide:** Describes how to install, configure, and use the Diagnostics Java Agent and the Diagnostics Profiler for Java.
- **Diagnostics .NET Agent Guide:** Describes how to install, configure, and use the Diagnostics .NET Agent and Diagnostics Profiler for .NET.
- **Diagnostics Collector Guide:** Explains how to install and configure a Diagnostics Collector.
- **Diagnostics System Requirements and Support Matrixes Guide:** Describes the system requirements for the various Diagnostics components.
- **Release Notes:** Provides last-minute new information and known issues about each version of Diagnostics. The PDF file is also located in the Diagnostics installation disk root directory.
- **Diagnostics Data Model and Query API:** Describes the Diagnostics data model and the query API you can use to access the data. The guide is also available from the Diagnostics online help Home page.
- **Diagnostics Frequently Asked Questions (FAQ):** Gives answers to frequently asked questions. The FAQ is also available from the Diagnostics online help Home page.

Introduction

Product Overview

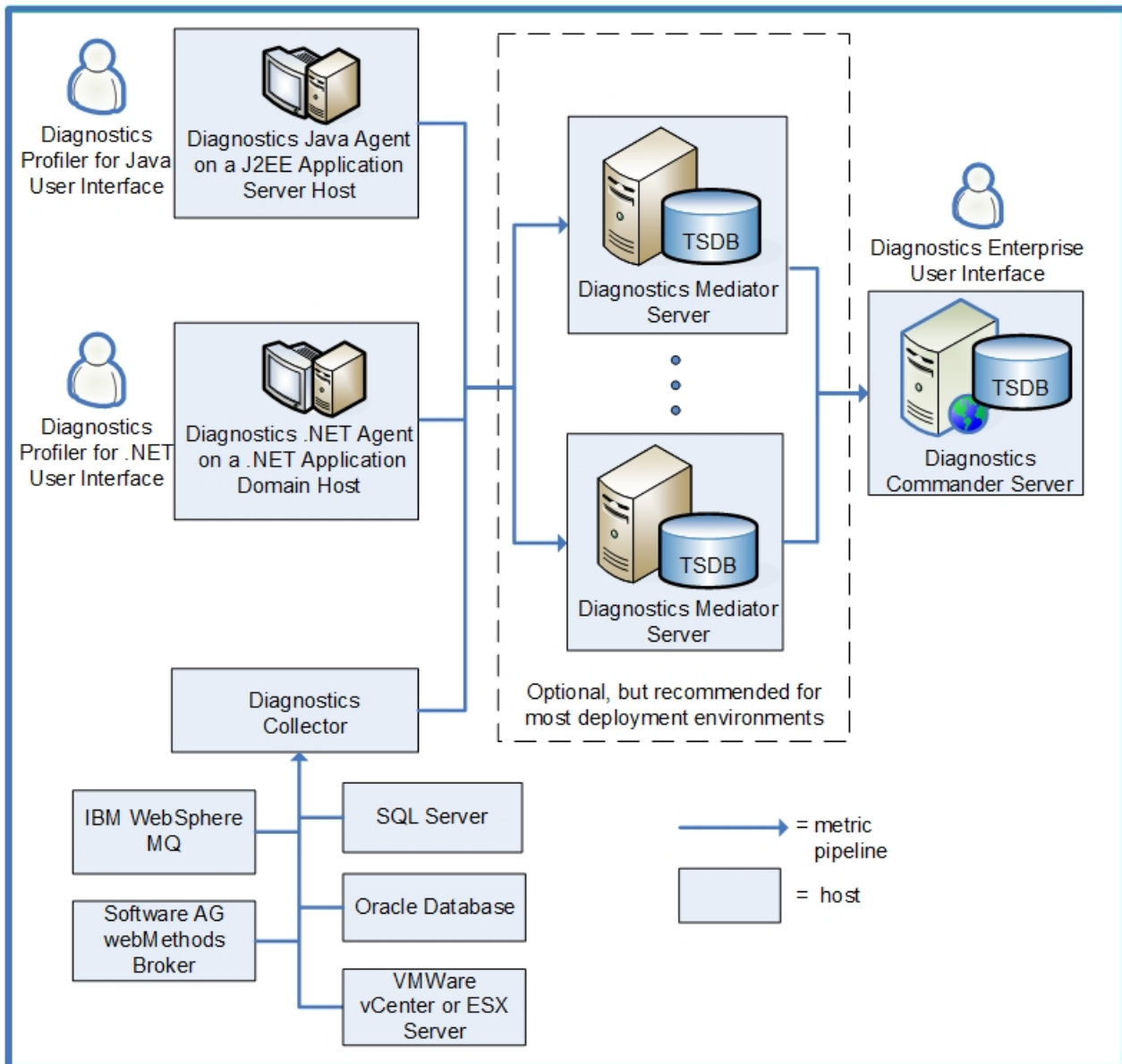
Read the information in this chapter for an overview of Diagnostics.

This chapter includes:

- ["Diagnostics Components and Data Flow" below](#)
- ["What does Diagnostics Monitor and Collect?" on the next page](#)
- ["Probes: Agent and Collector Instances" on page 16](#)
- [Integrations with Other Software Products](#)

Diagnostics Components and Data Flow

The following diagram shows the Diagnostics components and the data flow direction between them:



Diagnostics consists of the following components:

- **Diagnostics Java Agent:** Captures events such as method invocations, server requests, and system usage from a Java application. The agent populates a large set of metrics based on these events, aggregates them, and sends them on to a designated Diagnostics mediator server.

The Diagnostics Java Agent software is installed on the host that contains the Java EE application server that runs the Java application to be monitored. An agent can monitor multiple Java applications on the same host.
- **Diagnostics .NET Agent:** Captures events such as method invocations, server requests, and system usage from a .NET application. The agent populates a large set of metrics based on these events, aggregates them, and sends them on to a designated Diagnostics mediator server.

The Diagnostics .NET Agent software is installed on the host that contains the application domain that runs the .NET application to be monitored. An agent can monitor multiple .NET applications on the same host.
- **Diagnostics Collectors:** Collects metrics related to the operation of the systems shown in the diagram.

The Diagnostics Collector software is installed on any host that can access the system to be monitored. It does not need to be installed on the same host as the system to be monitored.
- **Diagnostics mediator server:** Receives and further aggregates the incoming metrics from the agents and collectors, and manages the storage and aging of the metrics in its own local Times Series Database (TSDB).

In small or test environments you can omit the Diagnostics mediator server. In this case, you must configure the Diagnostics commander server to perform both the commander and mediator roles. This configuration is done during installation of the Diagnostics commander server.

In a typical deployment environment however, there is a single Diagnostics commander server and one or more Diagnostics mediator servers. Each Diagnostics mediator server receives data from specific agents or collectors in the deployment environment.
- **Diagnostics commander server:** Retrieves the data as needed from the Diagnostics mediator servers so that it can be displayed in the views of the Diagnostics Enterprise UI when requested.

The Diagnostics commander server also manages the communication between all Diagnostics components in the deployment and knows their location and status.

In small or test environments with no Diagnostics mediator servers, the Diagnostics commander server has its own TSDB.
- **Diagnostics Enterprise User Interface:** Provides various views of the collected metrics that have been analyzed. The views display performance data at high levels in customizable graphs and tables, with the ability to drill down to individual metric values. The interface is designed to allow you to easily monitor performance of the monitored application, isolate performance problems, and drill down to root causes. This UI is accessed through a supported web browser.
- **Diagnostics Profilers:** Provides access to raw metric data on the agent host directly, before it has been processed by the Diagnostics mediator server or Diagnostics commander server. This UI is accessed through a supported web browser.

What does Diagnostics Monitor and Collect?

Diagnostics supports the monitoring of the following.

- ["Java Applications" on the next page](#)
- [".NET Applications" on page 14](#)
- ["Database Applications" on page 14](#)

- ["Queue Managers in Messaging Middleware" on the next page](#)
- ["Virtualization Environments" on page 15](#)
- ["SAP" on page 15](#)

Java Applications

Diagnostics' Java agents monitor Java applications that run on Java application servers, such as Apache Tomcat, IBM WebSphere, Oracle WebLogic and many others. The Java Platform, Enterprise Edition, or *Java EE*, defines the API and features of these Java application servers. Previous versions of this platform are called J2EE.

Each agent monitors a set of pre-defined methods in the API including those provided by the Servlet, JDBC, and JMS interfaces. Only "interesting" methods are tracked in order to keep overhead low but still provide the ability to find the root cause of performance problems.

For these methods, the agent tracks the following:

- Execution time of the method—the time it took for the method to execute.
- Count invocations—how many times the same method was called.
- Exceptions—any exceptions during the execution of the method.
- Call profiles—the stack of all methods that a server request calls.

Diagnostics Java Agents (as well as .NET agents) also collect system metrics from the host on which the application server is running. The following system metrics are collected on most operating systems:

- Host—the host machine name.
- Memory usage—percentage of primary memory (RAM) in use.
- Virtual memory usage—percentage of virtual memory in use.
- Context switches—average number of context switches among processes per second.
- Disk bytes per second—average number of bytes read and written to disk per second.
- Disk I/O per second—average number of disk I/O operations per second.
- Network bytes per second—average number of bytes sent and received by the host per second, as reported by the operating system and may include the loopback network depending on OS and OS version.
- Network I/O per second—average number of network I/O operations per second.
- Page ins per second—average number of virtual memory pages swapped in to primary memory per second.
- Page outs per second—average number of primary memory pages swapped out to virtual memory per second.

You can configure Java Agents to collect more data than its out-of-box configuration specifies. You can enable metrics to be collected from the JMX technology of an application server. You can enable Client Monitoring which collects metrics related to the web page performance. You can specify custom methods that you want the Java Agent to monitor.

.NET Applications

Diagnostics' .NET agents monitor .NET applications that run on the Microsoft .NET Framework. .NET Agents automatically discover ASP.NET applications and monitor a set of pre-defined methods in the ASP.NET, WCF, WebAPI, and ADO.NET interfaces. Only “interesting” methods are tracked in order to keep overhead low but still provide the ability to find the root cause of the problem.

The agent tracks the following for these methods:

- Execution time of the method—the time it took for the method to execute.
- Count invocations—how many times the same method was called.
- Exceptions—any exceptions during the execution of the method.
- Call profiles—the stack of methods in which the method was called.

.NET Agents collect the system metrics listed previously in the Java Applications section.

You can configure .NET Agents to collect more data than its out-of-box configuration specifies. You can configure .NET Agents to monitor additional types of applications such as NT Service, console, or UI client. You can enable metrics to be collected from the following interfaces:

- IIS5, IIS6, and IIS7
- Lightweight Memory Diagnostics.
- Microsoft Message Queuing
- .NET Remoting
- ASP.NET Web Services

You can specify custom methods that you want the .NET Agent to monitor.

Database Applications

Diagnostics' Collectors monitor Microsoft SQL Server, Oracle 10g, or Oracle 11g databases. Collectors retrieve data from the system tables/views and performance counters of the database themselves.

For example, instance level metrics are retrieved from the **sys.dm_os_performance_counters** view (Microsoft SQL Server) or Instance level metrics are retrieved from the **V\$SYSMETRIC** view (Oracle).

Queue Managers in Messaging Middleware

Diagnostics' Collectors monitor queue managers in messaging middleware. Collectors retrieve the following :

- WebSphere MQ—Queue depth and message transfer rates, and channel message transfer rates are collected.
- TIBCO EMS—Pending message, consumer, producer counts are collected.
- Web Methods Broker—Queue depth, number of clients, and number of connections are collected.

Virtualization Environments

Diagnostics' Collectors monitor the VMware vCenter Server virtualization environment. VMware host and guest metrics are collected from VMware vCenter and VMware ESX servers

Collectors use VMware's vSphere Web Services API to obtain information about how the enclosing VMware infrastructure might affect the performance of guest operating systems hosted on Virtual Machines.

SAP

Diagnostics' Collectors monitor the Remote Function Call (RFC) interface in SAP NetWeaver ABAP systems.

You can also use the Java Agent to monitor the SAP Web Application Server (WAS) Java stack.

Note: Install and configure the Diagnostics Agents and Collectors.

- For Java application monitoring, see the Diagnostics Java Agent Guide.
- For .NET application monitoring, see the Diagnostics .NET Agent Guide.
- For monitoring of an Oracle Database, SAP NetWeaver-ABAP, SQL Server Database, VMware vCenter or VMware ESX servers, WebSphere MQ, TIBCO EMS and Software AG webMethods Broker environments, see Diagnostics Collector Guide.

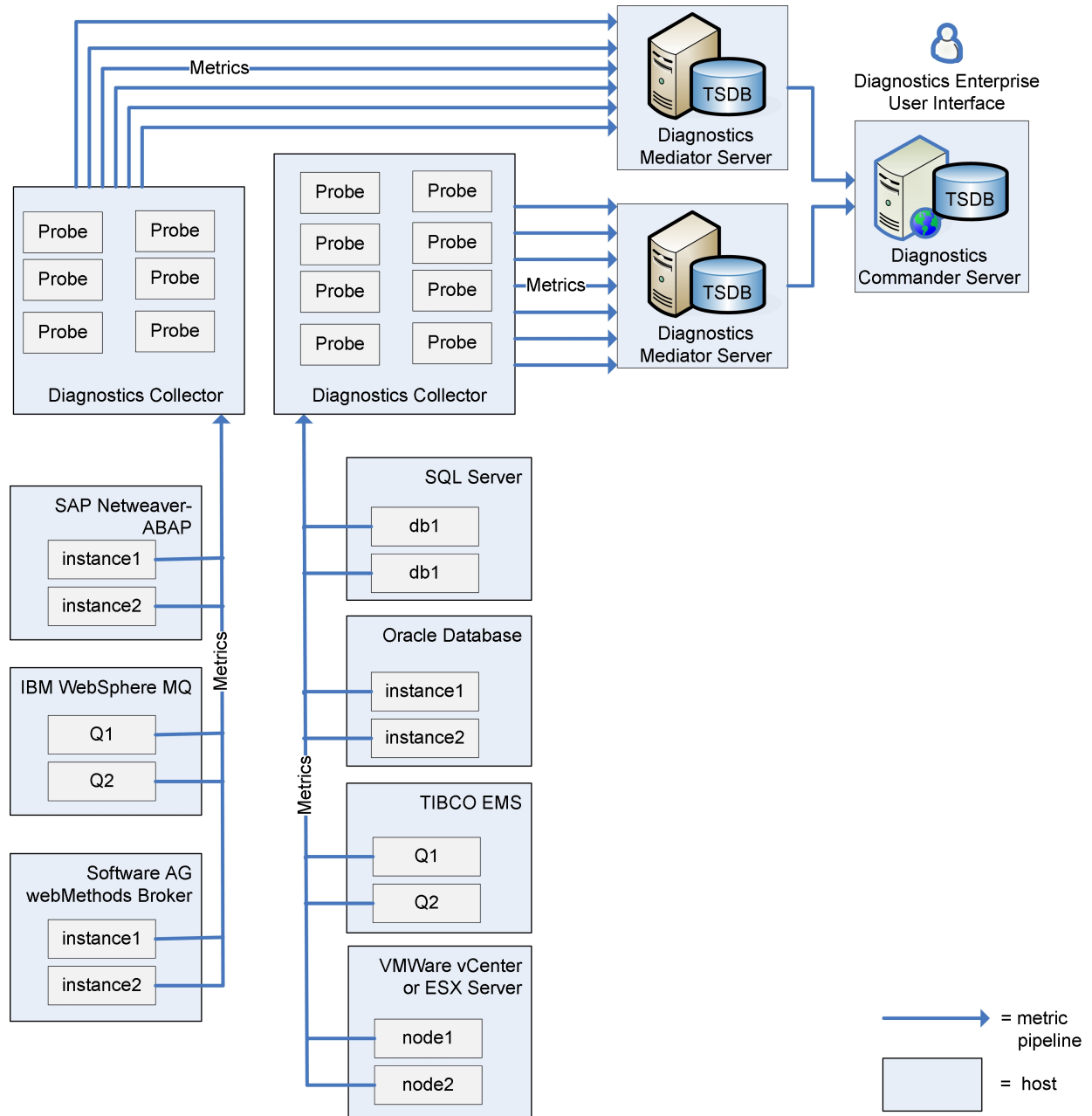
Probes: Agent and Collector Instances

Diagnostics agent and collector instances are referred to as *probes*. To understand the system requirements and licensing of the Diagnostics components, you need to understand the number of probes that your Diagnostics deployment will be using.

A Diagnostics agent can monitor more than one application on a host. Each monitored application server instance (for Java agents) or application domain (for .NET agents) results in an agent instance and is represented by a probe entity. Log files, error messages, and the Diagnostics Enterprise UI always use the term probe.

Each probe sends its metrics to the Diagnostics mediator server that was designated for the Agent at installation time. Probes cannot send to other Diagnostics mediator server in the Diagnostics deployment. Each probe has its own pipeline for the metrics it collects and can therefore be configured independently of other probes on the same host. For example, additional metrics can be enabled for collection, or metric collection can be trimmed to send fewer metrics.

Similar to the way that probes are used for agent instance metrics, probes are used for Collector instances.



Licenses

About Diagnostics Licensing

Micro Focus Diagnostics requires you to upload valid licenses onto the Diagnostics commander server.

This chapter includes:

- [About Diagnostics Licensing](#)
- ["Types of Licenses" below](#)
- ["Licensing the Diagnostics Server in Commander Mode" on the next page](#)
- ["View License Information" on page 21](#)

Diagnostics is licensed using a file that you upload to the Diagnostics commander server. You request this license file from your Micro Focus Software Customer Support representative.

When the Diagnostics agents and Diagnostics mediator server first connect with the Diagnostics commander server they are licensed based on the license installed on the Diagnostics commander server. The Diagnostics servers running as mediators, the Diagnostics Profiler, and the Diagnostics agents do not have independent licenses.

If you want the Diagnostics administrator to receive license checking alerts, when installing the Commander Server specify a comma-separated list of Admin Alert Email Addresses in the SMTP Settings installation dialog. Alternatively, the Admin address can be setup after installation using the Commander Server's Alert Properties page.

Types of Licenses

At installation you are given an **Instant-On license** which is packaged with the product. With the Instant-On license you can install Diagnostics components, begin to monitor applications, and process the performance metrics. The Instant-On license is valid for a fixed period of time from the time of installation or first use of the product.

Within this time period you must obtain a **Permanent license** or request an Evaluation license to extend the evaluation period. Evaluation licenses are available for Diagnostics to provide license keys that are meant to extend a customer's evaluation of the product. The Evaluation license is valid for a fixed period of time.

If the Instant-On license (or the extended Evaluation license) expires before you obtain a permanent license, the Diagnostics Server will issue reminder messages.

Your permanent license will typically be for a specific capacity (see ["License Information Based on Currently Connected Probes" on page 21](#)). After you install the license key, Diagnostics will count usage against this capacity.

For Diagnostics there are two types of LTUs (License to use):

- AM License - For use when using the product in an application management/enterprise mode, typically in a production environment. AM licensed agents can also be used with LoadRunner/Performance Center.
- AD License - For use when using the product in Diagnostics mode for LoadRunner/Performance Center runs in a pre-production load testing environment.

The Instant-On licenses you receive with Diagnostics have the following time and capacity limits: AM - 60 days and capacity of 50, AD - 14 days and capacity of 50.

You will see reminder messages when limits are exceeded. See ["License Information Based on Currently Connected Probes" on page 21](#) for details on AD and AM licenses.

Licensing the Diagnostics Server in Commander Mode

Obtain your Diagnostics license from your Micro Focus Software Customer Support representative. The License Management page described below contains useful information for determining the number of licenses required without having to manually retrieve the information from each system. This information is only available for Diagnostics 8.00 or later probes.

You will receive a license certificate from Micro Focus verifying the terms of the license purchase. License Keys/Passwords are issued after you enter the Sales Order Number associated with their software product purchase, which is unique for every order. This number appears on the license redemption form, as well as on all paperwork associated with the shipment and packaging of the order.

Store the license file in a directory that can be accessed from the License Management page for the Diagnostics commander server. Then upload it to the Diagnostics Commander Server as described in the steps below.

Note: For customers with licenses for versions prior to Diagnostics 9.10 your old licenses will still work with 9.10 or later versions. However the following section describes how to use the new licensing process for new purchases of Diagnostics 9.10 or later.

To license your Diagnostics deployment:

1. Access the License Management page for the Diagnostics commander server by accessing the Diagnostics Enterprise UI (http://<Diagnostics_Server>:2006).
2. Enter the login and password. Either use the default or whatever has been created and assigned to you. Default login is **admin** and default password is **admin**.
3. Select **Configure Diagnostics**.
4. Select the **license** link. The License Management page opens providing the following:
 - Information about current licenses.
 - A utility to upload a license received from Micro Focus Software Support.
 - Information on operating system instance totals as well as application server/probe instances in your monitored environment. You can also find information on usage against Diagnostics AD and AM license capacity.

Diagnostics

License Management

AM License Information	
Attribute	Value
License:	Diagnostics AM Implicit InstantOn License
Type:	Instant-on
Start Date:	Friday, November 16, 2018
Expiration:	Tuesday, January 15, 2019
Duration (days):	60
Days Remaining:	56
Capacity:	50
AD License Information	
Attribute	Value
License:	Diagnostics AD Implicit InstantOn License
Type:	Instant-on
Start Date:	Friday, November 16, 2018
Expiration:	Friday, November 30, 2018
Duration (days):	14
Days Remaining:	10
Capacity:	50
Autopass License Upload	
Note:	The uploaded file will be added to "DiagnosticsLicFile.txt".
License File:	<input type="button" value="Choose File"/> No file chosen
	<input type="button" value="Upload"/>
Server License Upload (Obsolete)	
Note:	You may only upload files ending in ".lic". The uploaded file will be renamed to "DiagnosticsServer.lic".
License File:	<input type="button" value="Choose File"/> No file chosen
	<input type="button" value="Upload"/>
License information based on currently connected probes	
Customer Name: Default Client	
Attribute	Value
Total Operating System Instances:	1
Application Management/Enterprise Mode (AM License) OS instances:	1
Load Runner/Performance Center (AD License) OS instances:	0
Total Application Server Instances:	1
Application Management/Enterprise Mode (AM License) probe instances:	1
Load Runner/Performance Center (AD License) probe instances:	0
.NET processes:	0
Python processes:	0
Java probes:	1
Old .NET probes:	0
Unknown probes:	0
Collector instances:	0
	<input type="button" value="Refresh"/>
	Details

- When you receive the license file for your Diagnostics deployment, upload the file using the **AutoPass License Upload** section of the License Management page.

The **Server License Upload (Obsolete)** section is obsolete and will only appear when the type of license key Diagnostics previously used (.lic file) is installed or only the Instant-On license is installed on

the server. This upload is provided for existing customers who already have a license from a Diagnostics version prior to 9.10 allowing you to upload your old license.

Note:

- Do not attempt to copy the license file directly to the Diagnostics Server installation directory. Always upload the file using the AutoPass License Upload section of the License Management page.
- Due to security restrictions, you cannot upload files from a remote location and can only upload them from the localhost.

Type the path to the location where you stored the license file or click **Browse** to navigate to the license file location. Click **Upload** to apply the license file to the Diagnostics Server.

If successful (the keys in the license file are valid and are not expired), the licenses are added to **DiagnosticsLicFile.txt** by the upload process and stored in the **<diag_server_install_dir>/etc** directory of the Diagnostics Commander Server. With AutoPass licensing you can upload incremental licenses which are added to the license file (you can't do this when mixed with the old licenses).

View License Information

Information on your current licenses is reported in the License Management page. You can see the type of license, expiration date, if any, and the license capacity.

License Information Based on Currently Connected Probes

In the License information section you will see counts based on currently connected probes. Counts are shown for operating system instances (see example below). This is useful in determining the number of licenses required without having to manually retrieve the information from each system.

License information based on currently connected probes	
Customer Name: Default Client	
Attribute	Value
Total Operating System Instances:	39
Application Management/Enterprise Mode (AM License) OS instances:	39
Load Runner/Performance Center (AD License) OS instances:	0
Total Application Server Instances:	69
Application Management/Enterprise Mode (AM License) probe instances:	69
Load Runner/Performance Center (AD License) probe instances:	0
.NET processes:	11
Java probes:	58
Old .NET probes:	0
Unknown probes:	0
Collector instances:	5
<input type="button" value="Refresh"/>	
Details	

The following counts are based on the number of operating system instances running an agent:

- **Total Operating System Instances.** Total number of operating system instances running an agent (not a collector). This is the sum of your AM and AD Operating System Instances. Your license capacity must cover this total
- **Application Management/Enterprise Mode (AM License) OS instances:** The number of OS instances that host Enterprise/AM mode agent instances in your production environment. These are counted against your Micro Focus Diagnostics AM license capacity.

When you install an agent, you are prompted to specify if the agent will be configured in Application Management/Enterprise mode (AM License) to work with a Diagnostics Server in a production environment. If you select this mode then the following values are set in Diagnostics:

- For a Java agent - the value of the **active.products** property in the **etc/probe.properties** file is set to **Enterprise** mode at the time you install the Java Agent. You can change the mode value after installation by modifying this property.
- For a .NET agent - the value of the **probe_config.xml <modes>** element is set to **enterprise** mode at the time you install the .NET Agent. You can change the mode value after installation by modifying this element.

For agents with Enterprise mode set, the agent hosts will be counted against your Micro Focus Diagnostics AM license capacity.

- **LoadRunner/Performance Center (AD License) OS instances:** The number of OS instances that host active LoadRunner or Performance Center AD mode application instances (does not include Enterprise/AM mode agent instances). Only active AD mode agents are counted against your Micro Focus Diagnostics AD license capacity. Those not in a run are not counted.

When you install an agent, you are prompted to specify if the agent will be configured in AD mode for LoadRunner and Performance Center runs. If you select the AD license option then the following values are set in Diagnostics:

- For a Java agent - the value of the **active.products** property in the **etc/probe.properties** file is set to **AD** mode at the time you install the Java Agent. You can change the mode value after installation by modifying this property.
- For a .NET agent - the value of the **probe_config.xml <modes>** element is set to **ad** mode at the time you install the .NET Agent. You can change the mode value after installation by modifying this element.

The advantage of running a probe in AD mode is that you only need license capacity for the number of hosts that are currently in a LoadRunner or Performance Center test run. So for example if you have agents installed on 100 test systems but you will only have probes running on 10 hosts at any one time then you would only need an AD license capacity of 10 hosts.

The following is for information only (these counts are not used as license counts) and relates to probe instances rather than OS instances (you can have more than one probe running on an OS instance).

- **Total Application Server Instances:** An application server instance is a Java Agent instance (a probe) or a .NET Agent instance (.NET worker process). This value is the total of Application Management/Enterprise Mode (AM License) probe instances and Load Runner/Performance Center (AD License) probe instances.
- **.NET processes:** Any processes (application domains) instrumented for monitoring by one or more .NET probes. For example, IIS worker process or .NET console application/service/WCF. In the license report you may see the number of Old .NET probes which are probes versioned prior to 8.00.
- **Java probes:** Monitored java or javaw processes or any other processes embedding the JVM. This is equivalent to a Java probe.

- **Collector instances:** Collector instances include the following:
 - Oracle - An instance in the (executed) Oracle software (Oracle processes) and the memory they use (SGA). A SID identifies an instance. Instances configured for monitoring with a <oracleInstance> entry in **oracle-config.xml** are included.
 - SQL Server - Instances apply primarily to the database engine and its supporting components. Instances configured for monitoring with a <sqlserverInstance> entry in **sqlserver-config.xml** are included.
 - WebSphere MQ - Instances configured for monitoring with a <mqInstance> entry in **mq-config.xml** are included.
 - TIBCO EMS - Instances configured for monitoring with a <emsInstance> entry in **tibco-ems-config.xml** are included.
 - WebMethods Broker - Instances configured for monitoring an <WmBrokerInstance> entry in **wm-broker-config.xml** are included.
 - SAP/ABAP - Each discovered Dialog instance (SAP ABAP probes) is included.
 - VMware - The number of vSphere servers as specified in the **vmware-config.xml** file are included.
- Any probes prior to 8.0x will be listed under Old probes.

License Details

Selecting the Details link at the bottom of the License page displays detailed information for each host with Diagnostics probes and collectors. Details include HostName, Probe Name, port or PID, Run ID (for probes in a LoadRunner/Performance Center load testing run), probe version and product mode.

Following is an example showing part of the License Management Details page:

License Management

Details for Default Client

.NET Probes					
Host Name	PID	Probe Name	Mode	Run ID	Version
OVRNTT209.ovrtest.adapps.com	:12084	L81_1ROOTCallChain2_0_DefaultWebSite.NET_OVRNTT209_W2k3	Enterprise,PRO	1	9.20.116.47012
OVRNTT209.ovrtest.adapps.com	:12084	L81_1ROOTJavaTrader2.WebClient_DefaultWebSite.NET_OVRNTT209_W2k3	Enterprise,PRO	1	9.20.116.47012
OVRNTT209.ovrtest.adapps.com	:12084	L81_1ROOTTestService2.WebClient_DefaultWebSite.NET_OVRNTT209_W2k3	Enterprise,PRO	1	9.20.116.47012
OVRNTT209.ovrtest.adapps.com	:12084	L81_1ROOTTestService2.WebService_DefaultWebSite.NET_OVRNTT209_W2k3	Enterprise,PRO	1	9.20.116.47012

Installation

Installation Overview

This section lists the installation choices and installation tasks for a new installation in the recommended order of installation.

Installation Types

Installation choices available for installing Diagnostics are

Install Type	Reference
Wizard-based Installation	See the "Recommended Order of Installation" below for an overview.
Silent Installation	See the chapter ""Silent Installation" on page 40" for details.

Recommended Order of Installation

Task	Description	Reference
1.	Obtain the Diagnostics license file.	"About Diagnostics Licensing" on page 18
2.	Review the pre-installation checklist.	"Pre-installation Checklist " on page 27
3.	Install the Diagnostics commander server.	"Steps to Install on Windows" on page 29 "Steps to Install on Linux" on page 33
4.	Install the Diagnostics mediator servers.	
5.	Install the license for the commander server.	"Licensing the Diagnostics Server in Commander Mode" on page 19.
7.	Install and configure the Diagnostics Agents and Collectors.	<ul style="list-style-type: none">• For Java application monitoring, see the Diagnostics Java Agent Guide.• For .NET application monitoring, see the Diagnostics .NET Agent Guide.• For Python application monitoring, see the Diagnostics Python Agent Guide.• For monitoring of an Oracle Database, SAP NetWeaver-ABAP, SQL Server Database, VMware vCenter or VMware ESX servers, WebSphere MQ, TIBCO EMS and Software AG webMethods Broker environments, see Diagnostics Collector Guide.
8.	Install OA and IAPA manually, if not already installed from the installation wizard.	"Manual Installation of OM Agent and IAPA Components" on page 182

Task	Description	Reference
9.	Configure the Diagnostics Server as needed for your deployment.	"Diagnostics Server Configuration" on page 52

Pre-installation Checklist

Review the following information before the installation:

- Determine whether you want to upgrade an existing Diagnostics Server or do a new install. An upgrade saves the licensing and other configuration of a Diagnostics Server. If you want to do a new install, uninstall the previous Diagnostics Server on the host if any.
- The Diagnostics deployment can consist of one or many Diagnostics Servers. If there is only one Diagnostics Server in your deployment, it is installed in Commander mode and can perform both commander and mediator roles. When there is more than one Diagnostics Server in a deployment, one is configured in Commander mode and all the rest in Mediator mode reporting to the Commander Server.
- Determine whether you are installing a Diagnostics commander server or a Diagnostics mediator server. Both are installed from the same installer. If you are installing a Diagnostics mediator server you need to obtain the details of the Diagnostics commander server to which it reports.
- Make sure the target host meets the system requirements for the mode of Diagnostics Server you are installing. For details, see "Requirements for the Diagnostics Server Host" in the relevant version of the **Diagnostics System Requirements and Support Matrices Guide** on the [Software Support site](https://softwaresupport.softwaregrp.com/group/softwaresupport/) (<https://softwaresupport.softwaregrp.com/group/softwaresupport/>).

For Diagnostics mediator servers, you will need to estimate the amount of data that this server will receive and size it accordingly.

- Make sure the web browser (client) host from which you will access the Diagnostics Enterprise UI meets the system requirements. For details, see "Requirements for the Diagnostics Enterprise UI" in the relevant version of the **Diagnostics System Requirements and Support Matrices Guide** on the [Software Support site](https://softwaresupport.softwaregrp.com/group/softwaresupport/) (<https://softwaresupport.softwaregrp.com/group/softwaresupport/>).
- The license file for your Diagnostics deployment is copied to the target host. You do not need this to perform the actual installation but you need it to begin setting up Diagnostics.
- If you are installing a Diagnostics commander server that will be integrated with BSM/APM, you must perform the installation the Administrator user. Obtain these credentials. The integration with BSM/APM is specified by the "This Server is to be used with BSM/APM" option during installation.
- For details on integrations with other Micro Focus Software products, see "Compatibility Matrix" in the relevant version of the **Diagnostics System Requirements and Support Matrices Guide** on the [Software Support site](https://softwaresupport.softwaregrp.com/group/softwaresupport/) (<https://softwaresupport.softwaregrp.com/group/softwaresupport/>).

For Linux

- If you are installing a Diagnostics commander server that will be integrated with BSM/APM, you must perform the installation as the root user. Obtain these credentials. The integration with BSM/APM is specified by the "This Server is to be used with BSM/APM" option during installation.
- If you plan to configure the Diagnostics Server to be started automatically after a system boot on Linux, you must perform the installation as the root user. Obtain these credentials. See "[Instructions for Linux Machines](#)" on page 1.
- You must be a root user to install the Diagnostics Server.
- Diagnostics Servers and Collectors on Linux machines require certain library packages to run. Installing a Server or Collector by running the installation program in graphical mode on Linux requires additional library packages. For details of special library packages required for Linux, see "[Library Packages Required on Linux](#)" on page 179.

Download the Installer

Perform the following steps to download the installer from the Micro Focus Software Download Center and launch the installer on Windows or UNIX operating systems.

To download the installer from the Software Download Center

1. Access the Micro Focus Software Download Center from the [Software Support web site](https://softwaresupport.softwaregrp.com/) (<https://softwaresupport.softwaregrp.com/>). This web site requires a Passport login.
2. Locate the relevant **Diagnostics** information and select the appropriate link for downloading the Diagnostics Server software.
3. Extract the contents of the downloaded .zip file and proceed with the installation. See "[Steps to Install on Windows](#)" on page 29 or "[Steps to Install on Linux](#)" on page 33 for detailed installation steps.

Steps to Install on Windows

Perform the following steps to install Diagnostics Server on Windows:

1. Run the installer from the downloaded location. Set the display options as needed using the command **DiagServer_<release number>_<platform>.exe** at the command prompt.
2. Read and accept the End User License Agreement. Click **Next** to continue.
3. Accept the default installation directory (C:\MercuryDiagnostics\Server) or click **Browse** to choose a different directory. Click **Next** to continue.
4. Select the Diagnostics Server mode for the Diagnostics Server that you are installing.

Select	If...
Commander Mode	<ul style="list-style-type: none"> • This is the only server in your deployment. • If there is more than one Diagnostics Server in your deployment, but the one you are currently installing is to be configured as Commander.
Mediator Mode	If there is more than one Diagnostics Server in your deployment, but the one you are currently installing is to be configured as Mediator.

Ignore the **This Server is to be used in an Software-as-a-Service (SaaS) environment** checkbox as this is to be used by an Micro Focus SaaS administrator installing a Diagnostics Server (either Commander or Mediator) on Micro Focus premises.

At this stage, the installation differs based on the choice of mode.

- To install the Diagnostics commander server, continue with ["If you are installing the Diagnostics Server in Commander Mode, continue as follows:"](#) below.
- To install a Diagnostics mediator server, continue with ["If you are installing the Diagnostics Server in mediator mode, continue as follows:"](#) on page 31.

If you are installing the Diagnostics Server in Commander Mode, continue as follows:

5. Select one of the following time synchronization methods. For diagnostics data to be correlated properly, all the components in the Diagnostics deployment must be time-synchronized.

Synchronize with an NTP server	This option applies only if the Diagnostics Server can access an NTP Server outside the firewall. This is the default method.
Synchronize with the registered BSM/APM server	If the Diagnostics Server is to work in a BSM/APM environment, select this option to synchronize with the BSM/APM server.
Synchronize with system time	Select this option if the Diagnostics Server is to work in an environment other than BSM/APM and there is no access to an NTP server.

Click **Next** to continue.

6. Select optional configurations for the Diagnostics Server.

This Server is to be used with Business Service Manager (BSM) / Application Performance Manager (APM): Check this box if the Diagnostics commander server will be integrated with BSM/APM.

Checking this option means additional OM agent and IAPA components are installed for use in sending Health Indicators to BSM/APM. IAPA is the Integration Adapter Policy Activation component of the OMi agent that Diagnostics uses to communicate with BSM/APM.

If the option **Integrate with BSM/APM** is selected, an option to Install OVO Agent/IAPA will appear, select this option if required.

See APM-Diagnostics Integration Guide for additional post install configuration required to integrate with BSM/APM.

Select the option that applies to this Diagnostics Server, and then select **Next** to continue.

Note: The Diagnostic Agent and Collector installer is available in APM if it has been placed in the required directory for APM to access. After installing the Diagnostic Server, you can manually copy the Diagnostics Agent and Collector installers from the installation disk to the **<diag_server_install_dir>/html/opal/downloads** folder of the Diagnostics Server installation directory.

7. Select the OM Agent and IAPA component installations check box to install them now or you can leave the check box unselected to install these components later manually. See ["Manual Installation of OM Agent and IAPA Components" on page 182](#) for details. (Applicable if the Diagnostics Commander Server integrates with BSM/APM).

Errors are reported in the **<BSM_server_install_dir>/server/log.txt** file. See APM-Diagnostics Integration Guide

Note: If the OM agent is already installed on the system then these installers will update the OM agent components if they are an older version.

8. Indicate the SMTP setting for email alerts (optional). You can skip this dialog if you choose to configure these settings later using the Diagnostics Server's Alert Properties page (see the Diagnostics User Guide section on Alerts for more information).

SMTP Server	Host name or IP address of the SMTP server.
SMTP Port	Port number for the SMTP server.
From Email Address	The email address to send the email messages from.
Admin Alert Email Addresses	If you want the Diagnostics administrator to receive email alerts when there are problems with this Diagnostics server then specify a comma-separated list of email addresses for the administrator. Alerts can be issued for problems such as probes generating large number of server requests to the server, disk space issues on the server or from the Commander Server - license checking alerts.

The thresholds that determine these types of alerts for the Diagnostics administrator are factory configured in the server's **server.properties** file. For more details see the comments in this file for the various **watchdog** properties. Also see ["Disk Space Issues on the Server" on page 149](#).

9. Review the pre-installation summary information. To change your settings in the previous installation steps, click **Back**. To start the installation of the Diagnostics Server, select **Next**.

Note: The estimated total size of the Diagnostics Server in Commander mode installation does not include the size of the Agent installers, if they were made available for BSM/APM.

The server installation starts. When the installation is complete you will see the post-installation summary information or, if you selected integration with BSM/APM, an additional dialog is displayed. Check the post-installation summary and select **Finish** to exit the installation or continue on to the next step if integrating with APM.

The Diagnostics Server starts automatically.

If you are installing the Diagnostics Server in mediator mode, continue as follows:

5. Provide the location of the Diagnostics Server in Commander mode. Enter the host name or IP address and port for the Diagnostics commander server.
 - The default port for the Diagnostics commander server is **2006** except as noted above for SaaS. If you changed the port since the Diagnostics Server was installed, specify that port number here instead of the default. For information on changing the Diagnostics Server port, see ["Change the Default Diagnostics Server Port"](#) on page 67.
 - To allow the installer to check the connectivity to the host and port that you specified, select **Check the Connectivity to the Diagnostics Server**.

If you are using Software-as-a-Service (SaaS) then the commander server is installed by Micro Focus on an Micro Focus SaaS system and Micro Focus will provide you with information on the host name and port to use as well as any other configuration required on the mediator server you are installing.

Ignore the **This mediator is to be used in an Software-as-a-Service (SaaS) environment** checkbox as this is to be used by an Micro Focus SaaS administrator installing a Diagnostics mediator server on Micro Focus premises.

Click **Next** to continue.

If you instructed the installer to perform the test for connectivity, it tests the connectivity at this point. If there are negative results, it reports these before proceeding with the next installation step.

6. Indicate the SMTP setting for email alerts (optional). You can skip this dialog if you choose to configure these settings later using the Diagnostics Server's Alert Properties page (see the Diagnostics User Guide section on Alerts for more information).

SMTP Server	Host name or IP address of the SMTP server.
SMTP Port	Port number for the SMTP server.
From Email Address	The email address to send the email messages from.
Admin Alert Email Addresses	If you want the Diagnostics administrator to receive email alerts when there are problems with this Diagnostics server then specify a comma-separated list of email addresses for the administrator. Alerts can be issued for problems such as probes generating large number of server requests to the server, disk space issues on the server or from the Commander Server - license checking alerts.

The thresholds that determine these types of alerts for the Diagnostics administrator are factory configured in the server's **server.properties** file. For more details see the comments in this file for the various **watchdog** properties. Also see ["Disk Space Issues on the Server"](#) on page 149.

7. Indicate the Software-as-a-Service (SaaS) settings.

If you are not installing this mediator to work with an Micro Focus SaaS Diagnostics Commander then skip this dialog, do not check any boxes and leave the default value for Customer Name as Default_Client.

<p>This mediator is to be used in an Micro Focus Software-as-a-Service (SaaS) environment</p>	<p>Select this check box if the Diagnostics Mediator Server will be used in a SaaS environment and enter the customer name.</p>
<p>Customer Name</p>	<p>If you indicated that the Diagnostics Server is to be used in an Software-as-a-Service (SaaS) environment, provide the customer name for the Software-as-a-Service (SaaS) environment. This name would have been given to you by your Micro Focus SaaS administrator. If you are not installing the server for SaaS then leave the value as Default_Client.</p>
<p>This mediator is Micro Focus SaaS-hosted (installed on Micro Focus premises)</p>	<p>Select this check box if you are an Micro Focus SaaS administrator installing this mediator server for the customer on an Micro Focus system. The server will be configured to reduce the frequency of data being sent from the customer site to Micro Focus systems in order to reduce network usage.</p>

Click **Next** to continue.

- Review the pre-installation summary information. To change your settings in the previous installation steps, click **Back**. To start the installation of the Diagnostics Server, select **Next**.

Installation begins. When the installation is complete, review the post-installation summary information to make sure that the installation completed successfully. Select **Finish** to exit the installation.

The Diagnostics Server starts automatically.

Steps to Install on Linux

Perform the following steps to install Diagnostics server on a Linux system:

1. Run the installer from the downloaded location. Choose the mode to run the installer.

Console Mode	Specify the DiagServer_<release number>_<platform>.bin filename with the <code>-i</code> console option, at the command prompt. For example, <code>DiagServer_<release number>_<platform>.bin -i console</code>
Graphical mode	Set the display options as needed and then specify the DiagServer_<release number>_<platform>.bin . For example, <code>DiagServer_<release number>_<platform>.bin</code>

2. Accept the End User License Agreement. Read the agreement and accept the terms of the agreement. You can press Enter as you read to move to the next page of text, or type q to jump to the end of the license agreement. Click **Next** to continue.

In console mode installer, type 1 to select Next, 2 for Previous, 3 to Cancel, or 4 to Re-display the screen.

3. Accept the default installation directory or click **Browse** to navigate to another directory. Click **Next** to continue.

4. Select the Diagnostics Server mode for the Diagnostics Server that you are installing.

Select	If...
Commander Mode	<ul style="list-style-type: none"> • This is the only server in your deployment. • If there is more than one Diagnostics Server in your deployment, but the one you are currently installing is to be configured as Commander.
Mediator Mode	If there is more than one Diagnostics Server in your deployment, but the one you are currently installing is to be configured as Mediator.

Ignore the **This Server is to be used in an Software-as-a-Service (SaaS) environment** checkbox as this is to be used by an Micro Focus SaaS administrator installing a Diagnostics Server (either Commander or Mediator) on Micro Focus premises.

At this stage, the installation differs according to whether you are installing the Diagnostics Server in Commander or Mediator mode.

- To install the Diagnostics commander server, continue with ["If you are installing the Diagnostics Server in Commander Mode, continue as follows:"](#) below.
- To install a Diagnostics mediator server, continue with ["If you are installing the Diagnostics Server in mediator mode, continue as follows:"](#) on page 35.

If you are installing the Diagnostics Server in Commander Mode, continue as follows:

5. Select a time synchronization method.
6. Select one of the following time synchronization methods.

Synchronize with an NTP server	This option applies only if the Diagnostics Server can access an NTP Server outside the firewall. This is the default method.
---------------------------------------	---

Synchronize with the registered BSM/APM server	If the Diagnostics Server is to work in a BSM/APM environment, select this option to synchronize with the BSM/APM server.
Synchronize with system time	Select this option if the Diagnostics Server is to work in an environment other than BSM/APM and there is no access to an NTP server.

This Server is to be used with Business Service Manager (BSM) / Application Performance Manager (APM): Check this box if the Diagnostics commander server will be integrated with BSM/APM.

Checking this option means additional OM agent and IAPA components are installed for use in sending Health Indicators to BSM/APM. IAPA is the Integration Adapter Policy Activation component of the OMI agent that Diagnostics uses to communicate with BSM/APM.

If the option **Integrate with BSM/APM** is selected, an option to Install OVO Agent/IAPA will appear, select this option if required.

See APM-Diagnostics Integration Guide for additional post install configuration required to integrate with BSM/APM.

Select the option that applies to this Diagnostics Server, and then select **Next** to continue.

Note: The Diagnostic Agent and Collector installer is available in APM if it has been placed in the required directory for APM to access. After installing the Diagnostic Server, you can manually copy the Diagnostics Agent and Collector installers from the installation disk to the **<diag_server_install_dir>/html/opal/downloads** folder of the Diagnostics Server installation directory.

8. Select the OM Agent and IAPA component installations check box to install them now or you can leave the check box unselected to install these components later manually. See ["Manual Installation of OM Agent and IAPA Components" on page 182](#) for details.

Note: If the OM agent is already installed on the system then these installers will update the OM agent components if they are an older version.

9. Indicate the SMTP setting for email alerts (optional). You can skip this dialog if you choose to configure these settings later using the Diagnostics Server's Alert Properties page (see the Diagnostics User Guide section on Alerts for more information).

SMTP Server	Host name or IP address of the SMTP server.
SMTP Port	Port number for the SMTP server.
From Email Address	The email address to send the email messages from.
Admin Alert Email Addresses	If you want the Diagnostics administrator to receive email alerts when there are problems with this Diagnostics server then specify a comma-separated list of email addresses for the administrator. Alerts can be issued for problems such as probes generating large number of server requests to the server, disk space issues on the server or from the Commander Server - license checking alerts.

The thresholds that determine these types of alerts for the Diagnostics administrator are factory configured in the server's **server.properties** file. For more details see the comments in this file for the various **watchdog** properties. Also see ["Disk Space Issues on the Server" on page 149](#).

10. Review the pre-installation summary information. To change your settings in the previous installation

steps, click **Back**. To start the installation of the Diagnostics Server, select **Next**.

Note: The estimated total size of the Diagnostics Server in Commander mode installation does not include the size of the Agent installers, if they were made available for BSM/APM.

The server installation starts. When the installation is complete you will see the post-installation summary information or, if you selected integration with BSM/APM, an additional dialog is displayed. Check the post-installation summary and select **Finish** to exit the installation or continue on to the next step if integrating with APM.

Manually start the Diagnostics Server. See [Instructions to Start and Stop Diagnostics Servers for Linux Machines](#).

If you are installing the Diagnostics Server in mediator mode, continue as follows:

5. Provide the location of the Diagnostics Server in Commander mode. Enter the host name or IP address and port for the Diagnostics commander server
 - The default port for the Diagnostics commander server is **2006** except as noted above for SaaS. If you changed the port since the Diagnostics Server was installed, specify that port number here instead of the default. For information on changing the Diagnostics Server port, see "[Change the Default Diagnostics Server Port](#)" on page 67.
 - To allow the installer to check the connectivity to the host and port that you specified, select **Check the Connectivity to the Diagnostics Server**.

If you are using Software-as-a-Service (SaaS) then the commander server is installed by Micro Focus on an Micro Focus SaaS system and Micro Focus will provide you with information on the host name and port to use as well as any other configuration required on the mediator server you are installing.

Ignore the **This mediator is to be used in an Software-as-a-Service (SaaS) environment** checkbox as this is to be used by an Micro Focus SaaS administrator installing a Diagnostics mediator server on Micro Focus premises.

Click **Next** to continue.

If you instructed the installer to perform the test for connectivity, it tests the connectivity at this point. If there are negative results, it reports these before proceeding with the next installation step.

6. Indicate the SMTP setting for email alerts (optional). You can skip this dialog if you choose to configure these settings later using the Diagnostics Server's Alert Properties page (see the Diagnostics User Guide section on Alerts for more information).

SMTP Server	Host name or IP address of the SMTP server.
SMTP Port	Port number for the SMTP server.
From Email Address	The email address to send the email messages from.
Admin Alert Email Addresses	If you want the Diagnostics administrator to receive email alerts when there are problems with this Diagnostics server then specify a comma-separated list of email addresses for the administrator. Alerts can be issued for problems such as probes generating large number of server requests to the server, disk space issues on the server or from the Commander Server - license checking alerts.

- Indicate the Software-as-a-Service (SaaS) settings.
If you are not installing this mediator to work with an Micro Focus SaaS Diagnostics Commander then skip this dialog, do not check any boxes and leave the default value for Customer Name as Default_Client.

This mediator is to be used in an Micro Focus Software-as-a-Service (SaaS) environment	Select this check box if the Diagnostics Mediator Server will be used in a SaaS environment and enter the customer name.
Customer Name	If you indicated that the Diagnostics Server is to be used in an Software-as-a-Service (SaaS) environment, provide the customer name for the Software-as-a-Service (SaaS) environment. This name would have been given to you by your Micro Focus SaaS administrator. If you are not installing the server for SaaS then leave the value as Default_Client.
This mediator is Micro Focus SaaS-hosted (installed on Micro Focus premises)	Select this check box if you are an Micro Focus SaaS administrator installing this mediator server for the customer on an Micro Focus system. The server will be configured to reduce the frequency of data being sent from the customer site to Micro Focus systems in order to reduce network usage.

Click **Next** to continue.

- Review the pre-installation summary information. To change your settings in the previous installation steps, click **Back**. To start the installation of the Diagnostics Server, select **Next**. Installation begins. When the installation is complete, review the post-installation summary information to make sure that the installation completed successfully. Select **Finish** to exit the installation.
Manually start the Diagnostics Server. See [Instructions to Start and Stop Diagnostics Servers for Linux Machines](#).

Verify the Installation

This chapter includes:

- ["Verify the Installation" below](#)
- ["Start and Stop the Diagnostics Server" below](#)
- ["Determining the Version of the Diagnostics Server" on page 39](#)

Verify the Installation

To verify that a Diagnostics Server was installed correctly you can

- Check the `<diag_server_install_dir>/log/server.log` file for errors and warnings.
- Check that the Diagnostics Server service or daemon is running. See ["Start and Stop the Diagnostics Server" below](#).
- You can also launch the Diagnostics Enterprise UI to verify that the server is running, by entering the following URL in your browser: `http://<Diagnostics_commander_server>:2006/`. Use the default user/password of `admin/admin`, or your own login credentials if they have already been set up for you.

Security tip: It is highly recommended that the default credentials are changed as soon as possible after installation. For details on configuring users, roles, permissions and authentication, see ["User Authentication and Authorization" on page 101](#).

Note: Your Diagnostics software comes with an instant-on license so you can start using it right away. But eventually you will need to install your permanent license key; which is done on the Diagnostics Commander Server. For instructions on requesting a license file and uploading it, see ["About Diagnostics Licensing" on page 18](#).

Note that in order to see data from agents in the UI you will also need to install and configure Diagnostics agent and/or collector software to collect and report performance data to the server for display in the UI.

You can also check the System Health view to find information about the Diagnostics servers and the machines that host them.

To access the System Views

1. Open the Diagnostics UI as the System customer from `http://<Diagnostics_Commanding_Server_Name>:2006/query/`.
2. In the query page locate the System customer in the list and select the link to Open Diagnostics.
3. Log in to Diagnostics and on the Applications window select **Entire Enterprise** and select any link to open the Diagnostics Views.
4. In the Views pane you see the System Views view group. Open the view group and select either the **System Health** view or **System Capacity** view.

Start and Stop the Diagnostics Server

On Windows, Diagnostics servers are started automatically when the host on which they are installed is started. On Linux machines, you must start the Diagnostic Server manually or add the command to start it to

the machine's startup script. For troubleshooting purposes, you may need to manually start or stop a Diagnostics server at other times.

The procedures to manually start and stop the Diagnostics Server are described in the sections that follow.

Instructions for Windows

The recommended way to start the Diagnostics Server on Windows is by running it as a service. The service ensures that the server stays running (it automatically restarts the server if it goes down), and also provides communications between LoadRunner/Performance Center and Diagnostics. The service is started automatically when the system starts.

For troubleshooting purposes you may need to start the Diagnostics Server from the command line which runs it in the foreground as a process.

Both procedures are described below.

To start the Diagnostics Server service on Windows

Execute **Start Diagnostics Server** from the computer's Start menu.

You can verify that the Diagnostics Server is running by checking the status of the service named *Micro Focus Diagnostics Server* using Windows Administrative tools.

To stop the Diagnostics Server service on Windows

Execute **Stop Diagnostics Server** from the computer's Start menu.

If secure shutdown is enabled, use the following script to stop the server:

```
<server path>\Server\bin\Diagnostics_secure_shutdown.cmd
```

To start the Diagnostics Server process on Windows

```
<diag_server_install_dir>/bin/server.cmd
```

To stop the Diagnostics Server process on a Windows machine:

Terminate the process by typing **Ctrl + C** on the keyboard or by using the Task Manager.

Instructions for Linux Machines

The recommended way to start the Diagnostics Server on Linux is by using the *nanny*. The nanny is a process that runs as a daemon to ensure that the server stays running (it automatically restarts the server if it goes down), and also provides communications between LoadRunner/Performance Center and Diagnostics. The nanny can be started manually or automatically when the system starts.

For troubleshooting purposes you may need to start the Diagnostics Server from the command line which runs it in the foreground and does not use the nanny.

All procedures are described below.

To automatically start the Diagnostics Server using a nanny on a Linux machine:

Run the following command as the "root" user.

```
<diag_server_install_dir>/bin/nanny.sh -install
```

The nanny is installed as a Linux service under **/etc/rc.d** and will be started each time the machine starts.

To uninstall the Diagnostics Server nanny service on a Linux machine:

Run the following command as the "root" user.

```
<diag_server_install_dir>/bin/nanny.sh -remove
```

To manually start, stop, or restart the Diagnostics Server using a nanny on a Linux machine:

Run the following command ("root" user is not required).

```
<diag_server_install_dir>/bin/nanny.sh {start|stop|restart}
```

To verify that the nanny is started on a Linux machine:

Run the following command ("root" user is not required).

```
<diag_server_install_dir>/bin/nanny.sh status
```

To start the Diagnostics Server without using a nanny on a Linux machine:

Run the following command ("root" user is not required).

```
<diag_server_install_dir>/bin/server.sh
```


To stop the Diagnostics Server without using a nanny on a Linux machine:

Terminate the process using a utility such as **kill**.

Determining the Version of the Diagnostics Server

When you request support, you must know the version of the Diagnostics Server.

You can view the version of the Diagnostics Server in the About dialog box:

Select Help Menu  > **About Diagnostics**.

Silent Installation

A *silent installation* is performed automatically, without the need for user interaction. In place of user input, the silent installation takes values from the file *oviinstallparams.ini*. You can edit this file with a standard text editor.

For example, a system administrator who needs to deploy a component on multiple machines can use the *oviinstallparams.ini* file that contains all the prerequisite configuration information, and then perform a silent installation on multiple machines. This eliminates the need to provide any manual input during the installation procedure.

You can get the *oviinstallparams.ini* file from one of the following locations:

- The file is available as part of the install download along with the binaries. It is located in the **examples\silent_installation** directory.
- This document has an example file that you can copy, see [Example oviinstallparams.ini File](#).

To perform a silent installation

1. Place the **oviinstallparams.ini** file in the same directory as the installer binary file.
2. To run the installer in silent mode, specify the **DiagServer_<release number>_<platform>_setup.bin** or **DiagServer_<release number>_<platform>.exe** with the **-i** silent option at the command prompt.

For example,

```
DiagServer_<release number>_<platform>.bin -i silent
```

Example oviinstallparams.ini File

Following is an example of the *oviinstallparams.ini* file which you can modify as required.

```
#=====
#- Sample oviinstallparams.ini file
#- To install Diagnostics in non-interactive (silent) mode, edit this file according
#- to your needs and place it in the same folder where DiagServer_9.51_setup.exe file and
#- "packages" folder are.
#-
#- For silent installation, run the installer with "-i silent" flag
#- Example: DiagServer_9.51_setup.exe -i silent (Windows)
#- DiagServer_9.51_setup.bin -i silent (Unix)
#-
#- Note: We do not recommend changing the installer properties parameters that follow.
#=====
[installer.properties]
setup=DiagServer
licenseAgreement=true
#=====
#- [Windows only] Installation folder. The path cannot contain spaces and must end with
#- "Server".
#=====
prodInstallDir=C:\MercuryDiagnostics\Server\
#=====
#- User Input Field - installType[Commander or Mediator]
#=====
installType=Commander
#=====
```

```
#=User Input Field - method[NTP or BAC or SERVER]
#=====
timesynch.method=NTP
#=====
#=User Input Field - host
#=====
registrarInfo.host=localhost
#=====
#=User Input Field - port
#=====
registrarInfo.port=2006
#=====
#=User Input Field - customerName
#=====
customername=Default Client
#=====
#=User Input Field - smtp_server
#=====
smtp.smtp_server=
#=====
#=User Input Field - smtp_port
#=====
smtp.smtp_port=25
#=====
#=User Input Field - smtp_from_address
#=====
smtp.smtp_from_address=
#=====
#=User Input Field - smtp_admin_email_addresses
#=====
smtp.smtp_admin_email_addresses=
#=====
#=User Input Field - checkbox
#=====
bsmStatus=false
#=====
#=User Input Field - checkbox
#=====
saas=false
```

Upgrade

Upgrade Overview

This section describes the process for upgrading from earlier releases of Diagnostics. All releases contain a full replacement of the Diagnostics product components. The following table lists the upgrade tasks.

Task	Description	Reference
1.	Upgrade checklist	"Upgrade Checklist" on page 44
2.	Review the upgrade compatibility	
3.	Upgrade Diagnostics	"Upgrade on Windows" on page 45 "Upgrade on Linux" on page 48
4	Upgrade the Collector Upgrade Java Agents Upgrade .NET Agents	Collector Installation Guide Java Agent Guide .NET Agent Guide

Upgrade Checklist

Check the following before you begin the upgrade:

- Make sure that the host meets the system requirements for the latest version of the component, as they may have changed. For details, refer to the relevant version of the **Diagnostics System Requirements and Support Matrices Guide** on the [Software Support site](https://softwaresupport.softwaregrp.com/group/softwaresupport/) (<https://softwaresupport.softwaregrp.com/group/softwaresupport/>).
- All Diagnostics Servers in the deployment must be at the same version, but can use different patches (IPs) of the same version. If you update a Diagnostics Server you must upgrade all of the Diagnostics Servers in your deployment.
- The Diagnostics Server must be at a version higher than or equal to the highest version of connected agents or collectors.
- To obtain the maximum monitoring coverage and functionality from Micro Focus Diagnostics, follow these recommendations:
 - Have all agents, collectors and servers in your deployment use the same version.
 - Use the most current version of each component.
- Plan to upgrade the Diagnostics Server before upgrading agents.
- Contact Micro Focus Software Customer Support if you need to upgrade from an earlier version of APM or Performance Center that is integrated with your Diagnostics deployment. Also refer to the upgrade documentation for these products for important instructions relevant to the Diagnostics integration.
- If you are an Software-as-a-Service (SaaS) customer, contact SaaS Support for server upgrade instructions.
- During the installation the keystore is *overwritten* along with the JRE. As a result your trusted certificates will be unavailable after the upgrade.
- With each new release of Diagnostics you should re-record the Diagnostics Server silent install response files prior to performing silent installation on multiple machines.
- The 9.51 Diagnostics Server continues to support working with the following earlier agent and collector versions, however monitoring coverage and functionality is limited to what the agent/collector at that version provides:
 - Java Agent 9.2x, 9.3x, 9.4x, 9.5x
 - .NET Agent 9.2x, 9.3x, 9.4x, 9.5x
 - Collectors 9.2x, 9.3x, 9.4x, 9.5x

For information about Diagnostic Server compatibility with APM, see the APM System Requirements and Support Matrices. For information about upgrading a Diagnostics Server that is integrated with APM, see the APM-Diagnostics Integration Guide.

Upgrade on Windows

Steps to Upgrade a Diagnostics Server:

1. If the Diagnostics Server is integrated with BSM/APM, uninstall the OM Agent and IAPA components before upgrading the Diagnostics Server. For details, see ["Uninstall on Windows" on page 174](#).
2. Shut down the current Diagnostics Server.
3. Make a **backup copy** of the current Diagnostics Server directory. By default this is:
C:\MercuryDiagnostics\Server

You can specify a different directory when you install the Server.

The upgrade procedure requires you to uninstall the current Diagnostics Server. If there are any problems with the upgrade, you can use the backup copy to restore the Diagnostics Server and contact Customer Support to assist with the upgrade.

4. Uninstall the current Diagnostics Server, but retain the modified files when prompted (click **No to All** at the prompt). For information about how to uninstall the Diagnostics Server, see ["Uninstall on Windows" on page 174](#).
5. To install the new Diagnostics Server, stop the Diagnostics Server. The Diagnostics Server is started automatically when the installer finishes.
6. Compare the **etc** directory with the backup etc directory.

Note: When upgrading from a version prior to 9.20, you may have to modify the following two properties whether or not you previously customized them:

- If a copy of **server.properties** exists in the backup etc directory, copy the **thresholding.evaluation.status.red.for.availability** property and its value from the old file to the new file.
- If a copy of **thresholds.configuration** exists in the backup etc directory, copy the **com.mercury.diagnostics.common.data.graph.node.ProbeData.Availability** property and its value from the old file to the new file; otherwise, set the value in the new file to "-95" instead of ",-95".

These changes preserve the behavior of previously set Availability thresholds. If you missed these changes during the initial upgrade, you may make them later.

Apply any differences that were caused by the customizations that were made (found in the backup etc directory) to the **etc** directory so that they will not be lost. Here are some common changes:

Property File	Configuration Properties to Be Copied to the New Diagnostics Server
alerting.properties	SNMP and SMTP servers, mail addresses.
security.properties	If the system is set up for SSL mode, all parameters should be updated and certificates manually copied to the new /etc folder.
server.properties	Timeout/Trimming settings, Commander's URL. See the Important note above regarding the thresholding.evaluation.status.red.for.availability property.

Property File	Configuration Properties to Be Copied to the New Diagnostics Server
thresholds.configuration	Copy any customizations that were made. See the Important note above regarding the <code>com.mercury.diagnostics.common.data.graph.node.ProbeData.Availability</code> property.
webserver.properties	Default port information.
.htaccess	The <code>.htaccess</code> file is for security and it needs to be copied to the new <code>/etc</code> folder to retain your original settings for user roles.

7. If the system is integrated with LoadRunner or Performance Center, copy **run_id.xml** from the backup etc directory to the new etc directory to ensure that the Run ID is properly incremented for future runs.
8. If the Diagnostics Server is integrated with APM, copy over the **RegistrarPersistence.xml** file from the backup etc folder to the new etc folder. Then check the Diagnostics status in the APM > Admin > Diagnostics page and re-do the registration of Diagnostics server in APM if it is not working properly. See the APM-Diagnostics Integration Guide for more information.
9. If you are upgrading the Diagnostics commander server, copy the **DiagnosticsLicFile.txt** or **DiagnosticsServer.lic** file from the backup etc directory to the new etc directory.
10. If you are upgrading to Diagnostics version 9.51 from versions earlier than Diagnostics 9.23, upgrade the threshold and alert definitions by running the following script: **<diagnostics_server_install_dir>\bin\alert-migration.cmd**
To check that the upgrade of threshold and alert definitions was successful, verify that the following file exists: **<diagnostics_server_install_dir>\storage\thresholding\once\Default Client\script.thresholding**. (This file may disappear once you start the Diagnostics Server and the threshold and alert definitions have been successfully upgraded.)
11. Upgrade Oracle: Berkeley DB (SleepyCat) by running the following script: **<diagnostics_server_install_dir>\bin\sleepycat_upgrade.cmd**
12. Start the Diagnostics Server.
13. Clear your browser's cache and the Java plug-in cache. Restart the browser before you attempt to access the Diagnostics UI.
14. You can verify that the upgraded Diagnostics Server is running by checking the version in the System Health view in the Diagnostics UI. The version should be the latest version if the upgrade was successful and the Diagnostics Server was restarted.
To access the System Health view, open the Diagnostics UI as the System customer from `http://<Diagnostics_Commanding_Server_Name>:2006/query/`. Then in the Views pane select the System Views view group.
15. If you are upgrading to Diagnostics version 9.51 from version 9.23, shared custom views already created in the earlier version are not visible after the upgrade. To make the shared custom views visible, copy them (files with the format **Shared Views - *.xml**) from the **storage\userdata\Default Client\<username>** folder to the **storage\userdata\Default Client\<application id>** folder, for the relevant application. If there is no folder for a specific application in the **storage\userdata\Default Client** folder, create a shared view for that application and a folder is automatically created.
16. Once you are satisfied that the Diagnostic Server has been upgraded successfully, remove the backup copy you created in Step 3.

Note: When you open the custom views that were created in an earlier version of Diagnostics for the

first time in a newer version, Diagnostics will upgrade the view for any changes that are necessary because of changes that were made to the functionality of Diagnostics. When Diagnostics changes your custom views a message is displayed to let you know that your custom view has been modified.

Upgrade on Linux

Steps to upgrade a Diagnostics Server:

1. If the Diagnostics Server is integrated with BSM/APM, uninstall the OM Agent and IAPA components before upgrading the Diagnostics Server. For details, see ["Uninstall on Linux " on page 175](#).
2. Shut down the current Diagnostics Server.
3. Make a **backup copy** of the current Diagnostics Server directory. By default this is: **/opt/MercuryDiagnostics/Server**
You can specify a different directory when you install the Server.
The upgrade procedure requires you to uninstall the current Diagnostics Server. If there are any problems with the upgrade, you can use the backup copy to restore the Diagnostics Server and contact Customer Support to assist with the upgrade.
4. Uninstall the current Diagnostics Server, but retain the modified files when prompted. For information about how to uninstall the Diagnostics Server , see ["Uninstall on Linux " on page 175](#).
5. Install the new Diagnostics Server. The Server is not automatically started so you do not need to stop it.
6. Compare the **etc** directory with the backup etc directory.

Note: When upgrading from a version prior to 9.20, you may have to modify the following two properties whether or not you previously customized them:

- If a copy of **server.properties** exists in the backup etc directory, copy the **thresholding.evaluation.status.red.for.availability** property and its value from the old file to the new file.
- If a copy of **thresholds.configuration** exists in the backup etc directory, copy the **com.mercury.diagnostics.common.data.graph.node.ProbeData.Availability** property and its value from the old file to the new file; otherwise, set the value in the new file to "-95" instead of ",-95".

These changes preserve the behavior of previously set Availability thresholds. If you missed these changes during the initial upgrade, you may make them later.

Apply any differences that were caused by the customizations that were made (found in the backup etc directory) to the **etc** directory so that they will not be lost. Here are some common changes:

Property File	Configuration Properties to Be Copied to the New Diagnostics Server
alerting.properties	SNMP and SMTP servers, mail addresses.
security.properties	If the system is set up for SSL mode, all parameters should be updated and certificates manually copied to the new /etc folder.
server.properties	Timeout/Trimming settings, Commander's URL. See the Important note above regarding the thresholding.evaluation.status.red.for.availability property.

Property File	Configuration Properties to Be Copied to the New Diagnostics Server
thresholds.configuration	Copy any customizations that were made. See the Important note above regarding the <code>com.mercury.diagnostics.common.data.graph.node.ProbeData.Availability</code> property.
webserver.properties	Default port information.
.htaccess	The <code>.htaccess</code> file is for security and it needs to be copied to the new <code>/etc</code> folder to retain your original settings for user roles.

Note: The backup etc directory is not created when upgrading on some Linux systems, even though modified files exist. You need to use the backup server directory from Step 3 instead.

7. If the system is integrated with LoadRunner or Performance Center, copy **run_id.xml** from the backup etc directory to the new etc directory to ensure that the Run ID is properly incremented for future runs.
8. If the Diagnostics Server is integrated with APM, copy over the **RegistrarPersistence.xml** file from the backup etc folder to the new etc folder. Then check the Diagnostics status in the APM > Admin > Diagnostics page and re-do the registration of Diagnostics server in APM if it is not working properly. See the APM-Diagnostics Integration Guide for more information.
9. If you are upgrading the Diagnostics commander server, copy the **DiagnosticsLicFile.txt** or **DiagnosticsServer.lic** file from the backup etc directory to the new etc directory.
10. If you are upgrading to Diagnostics version 9.51 from versions earlier than Diagnostics 9.23, upgrade the threshold and alert definitions by running the following script: **<diagnostics_server_install_dir>\bin\alert-migration.sh**
To check that the upgrade of threshold and alert definitions was successful, verify that the following file exists: **<diagnostics_server_install_dir>\storage\thresholding\once\Default Client\script.thresholding**. (This file may disappear once you start the Diagnostics Server and the threshold and alert definitions have been successfully upgraded.)
11. Upgrade Oracle: Berkeley DB (SleepyCat) by running the following script: **<diagnostics_server_install_dir>/bin/sleepycat_upgrade.sh**
12. If you had previously configured the nanny to use the **umask** command to change the default permissions assigned to new files that are created by the Diagnostics process, you must reconfigure the new **/opt/HP/HPEDiagserver/nanny/linux/dat/mdrv.dat** file with the required umask setting. For further details, see "[File Permissions on Linux](#)" on page 178.
13. Start the Diagnostics Server.
14. Clear your browser's cache and the Java plug-in cache. Restart the browser before you attempt to access the Diagnostics UI.
15. You can verify that the upgraded Diagnostics Server is running by checking the version in the System Health view in the Diagnostics UI. The version should be the latest version if the upgrade was successful and the Diagnostics Server was restarted.
To access the System Health view, open the Diagnostics UI as the System customer from `http://<Diagnostics_Commanding_Server_Name>:2006/query/`. Then in the Views pane select the System Views view group.
16. If you are upgrading to Diagnostics version 9.51 from version 9.23, shared custom views already created in the earlier version are not visible after the upgrade. To make the shared custom views visible, copy

them (files with the format **Shared Views - *.xml**) from the **storage\userdata\Default Client\<username>** folder to the **storage\userdata\Default Client\<application id>** folder, for the relevant application. If there is no folder for a specific application in the **storage\userdata\Default Client** folder, create a shared view for that application and a folder is automatically created.

17. Once you are satisfied that the Diagnostic Server has been upgraded successfully, remove the backup copy you created in Step 3.

Note: When you open the custom views that were created in an earlier version of Diagnostics for the first time in a newer version, Diagnostics will upgrade the view for any changes that are necessary because of changes that were made to the functionality of Diagnostics. When Diagnostics changes your custom views a message is displayed to let you know that your custom view has been modified.

Configuration

Diagnostics Server Configuration

This section describes advanced configuration of the Diagnostics Server. Advanced configuration is intended for experienced users with in-depth knowledge of this product. Use caution when modifying any of the component properties.

This chapter includes:

- ["Configure for HTTP Proxy and Firewalls" below](#)
- ["Synchronize Time Between Diagnostics Components" on page 60](#)
- ["Configure the Diagnostics Mediator Server for a Large Deployment" on page 62](#)
- ["Override the Default Diagnostics Server Host Name" on page 66](#)
- ["Change the Default Diagnostics Server Port" on page 67](#)
- ["Migrate Diagnostics Server from One Host to Another" on page 67](#)
- ["Configure the Diagnostics Server for Multi-Homed Environments" on page 68](#)
- ["Reduce the Diagnostics Server Memory Usage" on page 70](#)
- ["Configure URI Trimming on the Server" on page 71](#)
- ["Configure Server Request Name Based Trimming" on page 71](#)
- ["Automate the Composite Application Discovery in Diagnostics" on page 71](#)
- ["Prepare a High Availability Diagnostics Server" on page 76](#)
- ["Probe Registration Auto-Assignment for Large Deployments" on page 77](#)
- ["Configure Diagnostics for ServiceGuard \(HA solution\)" on page 84](#)
- ["Diagnostics Server Assignments \(LoadRunner/Performance Center Runs\)" on page 85](#)
- ["Configure the Diagnostics Server for LoadRunner Offline Analysis File Size" on page 86](#)
- ["Configure Diagnostics Using the Diagnostics Server Configuration Pages" on page 87](#)
- ["Configure a Custom Context Root" on page 88](#)
- ["Configure Diagnostics for PCF" on page 88](#)

Configure for HTTP Proxy and Firewalls

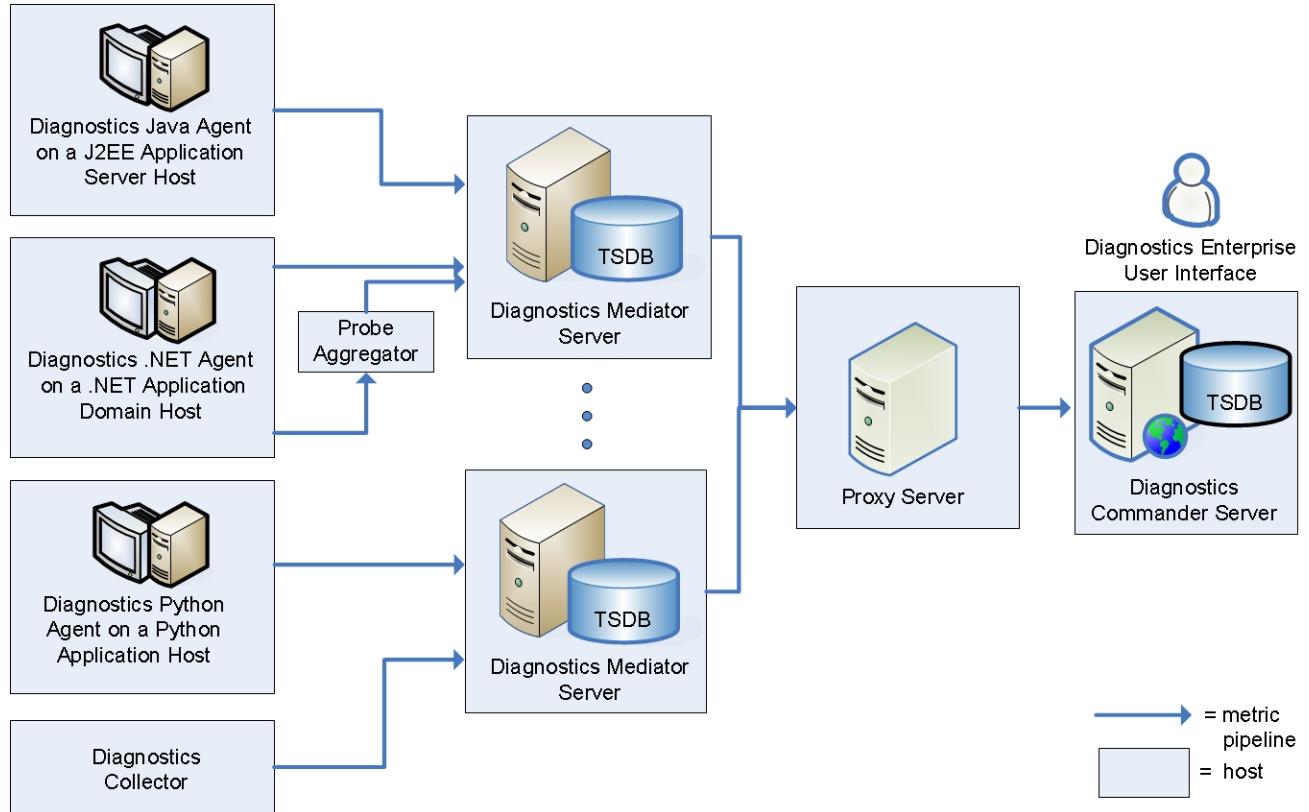
This chapter describes how to enable HTTP proxy communications between the Diagnostics components and how to configure for firewalls.

This chapter includes:

- ["Configure Diagnostics Servers for Proxy Communication" on the next page](#)
- ["Configure Agents and Collectors for Proxy Communication" on page 54](#)
- ["Configure for a Firewall Environment" on page 57](#)

Configure Diagnostics Servers for Proxy Communication

The Diagnostics deployment may require configuration to accommodate an HTTP proxy server between the Diagnostics mediator server and Diagnostics commander server.



The following steps describe how to configure the Diagnostics mediator server and Diagnostics commander server to communicate with each other through an HTTP proxy.

To configure a Diagnostics mediator server for HTTP proxy communication:

1. On the Diagnostics mediator server, edit `<diag_server_install_dir>/etc/server.properties`.
2. Set the following properties
 - Set **proxy.enabled** to true to enable proxy communication for the Diagnostics mediator server.
 - Set **proxy.host** to the host name of the proxy server.
 - Set **proxy.host** to the host name of the proxy server.
 - Set **proxy.port** to the port of the proxy server.
 - Set **proxy.protocol** to the protocol to use for the proxy server (http).
 - Set **proxy.user** to the user used to authenticate the proxy server.

- Set **proxy.password** to the password used to authenticate the proxy server.
- Set **commander.url** to the fully-qualified host name of the Diagnostics commander server.

For example:

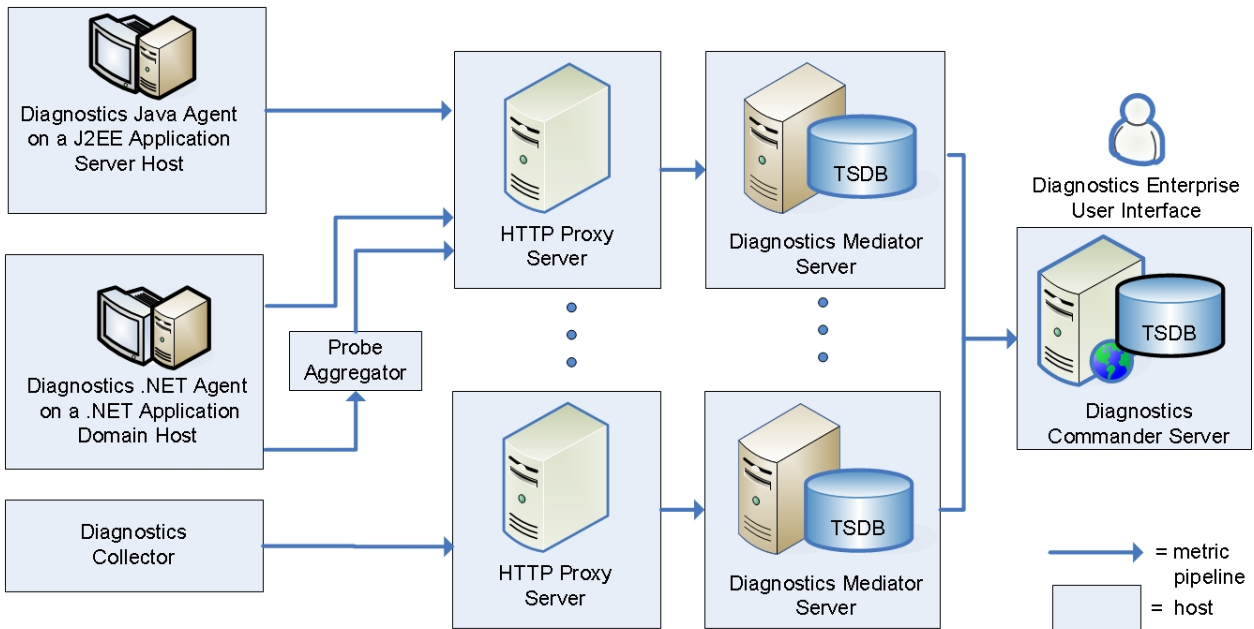
```
commander.url=http://diagcmdr01.mycompany.com:2006

# Proxy to use for communicating with remote Commanding Diagnostics Server
# When proxy.enabled=false, all other settings are ignored.
# When proxy.enabled=true, all other settings are used
# NOTE: Proxy authentication with proxy.user and proxy.password only supports
# "Basic" authentication. "Digest" and "NTLM" are not currently supported.
proxy.enabled=true
proxy.host=<host name of the proxy server>
proxy.port=<port of the proxy server>
proxy.protocol=<protocol to use for the proxy server (http)>
proxy.user=<user used to authenticate the proxy server>
proxy.password=<password used to authenticate the proxy server>
```

3. Restart the Diagnostics mediator server. See ["Start and Stop the Diagnostics Server" on page 37](#).

Configure Agents and Collectors for Proxy Communication

The Diagnostics deployment may require configuration to accommodate an HTTP proxy server between the agents or collectors and the Diagnostics mediator server:



Java Agents

To configure the Java Agent for HTTP proxy communication:

1. During installation of the Java Agent, set the **Use Proxy Server to connect to Diagnostics Server** check box, and specify the proxy-related properties as described below.

Or

After installation, edit the `<agent_install_dir>/etc/dispatcher.properties` file.

2. Set the following properties:
 - Set **proxy.enabled** to true to enable proxy communications for the Java Agent
 - Set **proxy.host** to the host name of the proxy server.
 - Set **proxy.port** to the port of the proxy server.
 - Set **proxy.protocol** to the protocol to use for the proxy server (http).
 - Set **proxy.user** to the user used to authenticate the proxy server.
 - Set **proxy.password** to the password used to authenticate the proxy server.
3. Restart the application server which restarts the probe.

.NET Agents

The way that you configure the .NET Agent for proxy communication depends on whether the .NET Agent is using the Probe Aggregator.

- [".NET Agents without the Probe Aggregator" below](#)
- [".NET Agents with the Probe Aggregator" on the next page](#)

.NET Agents without the Probe Aggregator

To configure a .NET Agent that is not using the Probe Aggregator for HTTP proxy communication:

1. After installation, edit the `<agent_install_dir>/etc/probe_config.xml` and `<agent_install_dir>/etc/metrics.config` files.
2. Add the following section of proxy properties to the `<agent_install_dir>/etc/probe_config.xml` file:

```
<diagnosticsserver url="http://<diagserver_host_name>:2006/registrar/"
proxy="http://proxy:8080" proxyuser=" <username>" proxypassword="<password>"/>
```

Where:

- **url** is the host for the Diagnostics mediator server.
- **proxy** is the proxy url.
- **proxy.user** is the user used to authenticate the proxy server.
- **proxy.password** is the password used to authenticate the proxy server.

3. Add the following section of proxy properties to the `<agent_install_dir>/etc/metrics.config` file:

```
proxy.uri = "http://<proxy_host_FQDN_name>"
proxy.user = "<username>"
proxy.password = "<password>"
```

Where:

- **proxy.uri** is the proxy server url.
 - **proxy.user** is the user used to authenticate the proxy server.
 - **proxy.password** is the password used to authenticate the proxy server.
4. Restart IIS or the Web publishing service to pick up the new agent configuration.

.NET Agents with the Probe Aggregator

To configure a .NET Agent that is using the Probe Aggregator for HTTP proxy communication:

1. After installation, edit the `<agent_install_dir>/ProbeAggregator/etc/probeaggregator.properties` file.
2. Set the following properties:
 - Set **proxy.enabled** to true to enable proxy communications for the .NET agent.
 - Set **proxy.host** to the host name of the proxy server.
 - Set **proxy.port** to the port of the proxy server.
 - Set **proxy.protocol** to the protocol to use for the proxy server (http).
 - Set **proxy.user** to the user used to authenticate the proxy server.
 - Set **proxy.password** to the password used to authenticate the proxy server.
3. Restart the Probe Aggregator Service.

Collectors

To configure a Collector for HTTP proxy communication:

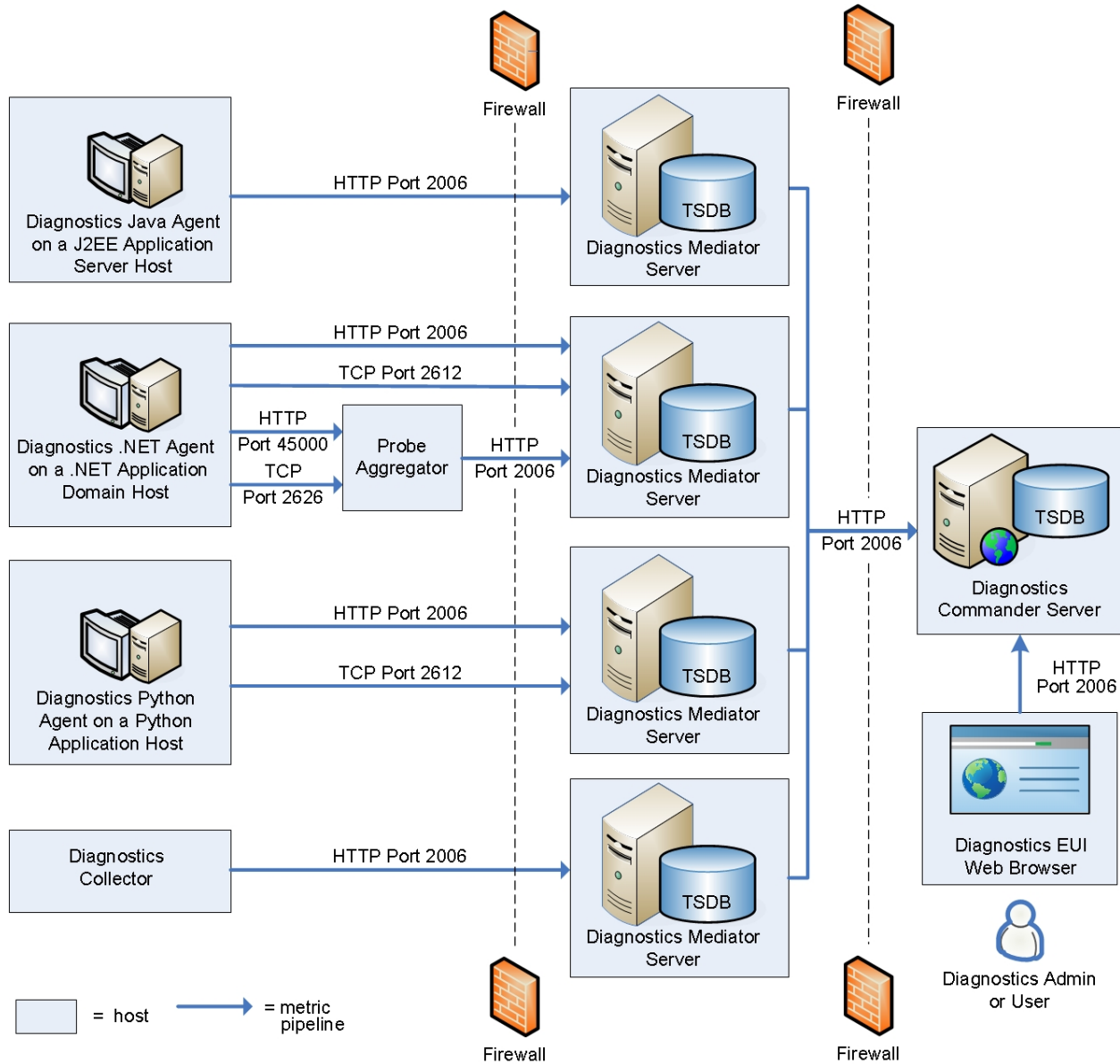
1. Install the Collector.
2. Edit the `<collector_install_dir>/etc/collector.properties` file.
3. Set the following properties:
 - Set **proxy.enabled** to true to enable proxy communications for the Collector.
 - Set **proxy.host** to the host name of the proxy server.
 - Set **proxy.port** to the port of the proxy server.
 - Set **proxy.protocol** to the protocol to use for the proxy server (http).
 - Set **proxy.user** to the user used to authenticate the proxy server.

- Set **proxy.password** to the password used to authenticate the proxy server.

4. Restart the Collector.

Configure for a Firewall Environment

The following diagram shows the default ports in a Diagnostics deployment and two possible firewall locations.



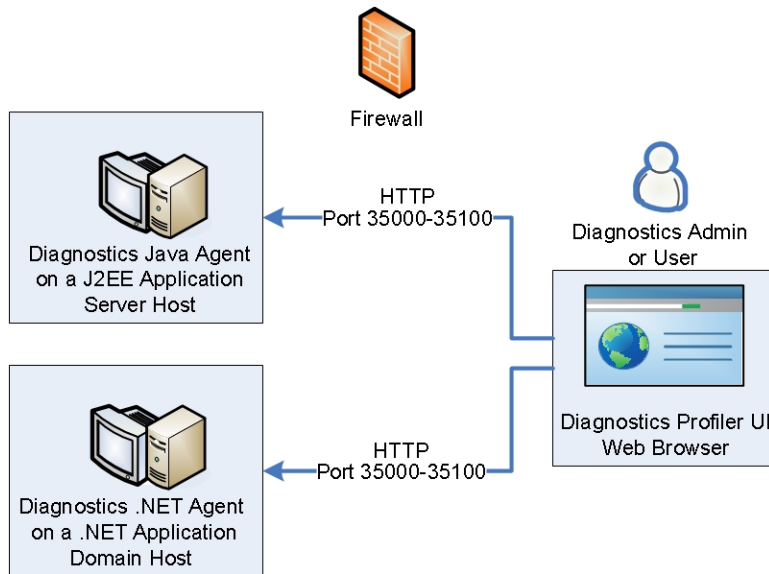
To configure a firewall to enable the communications between Diagnostics components, open the ports that will allow the communication needed for your deployment:

- The Diagnostics mediator server listens for HTTP on port 2006 (HTTPS 8443), and listens for TCP/IP on port 2612.

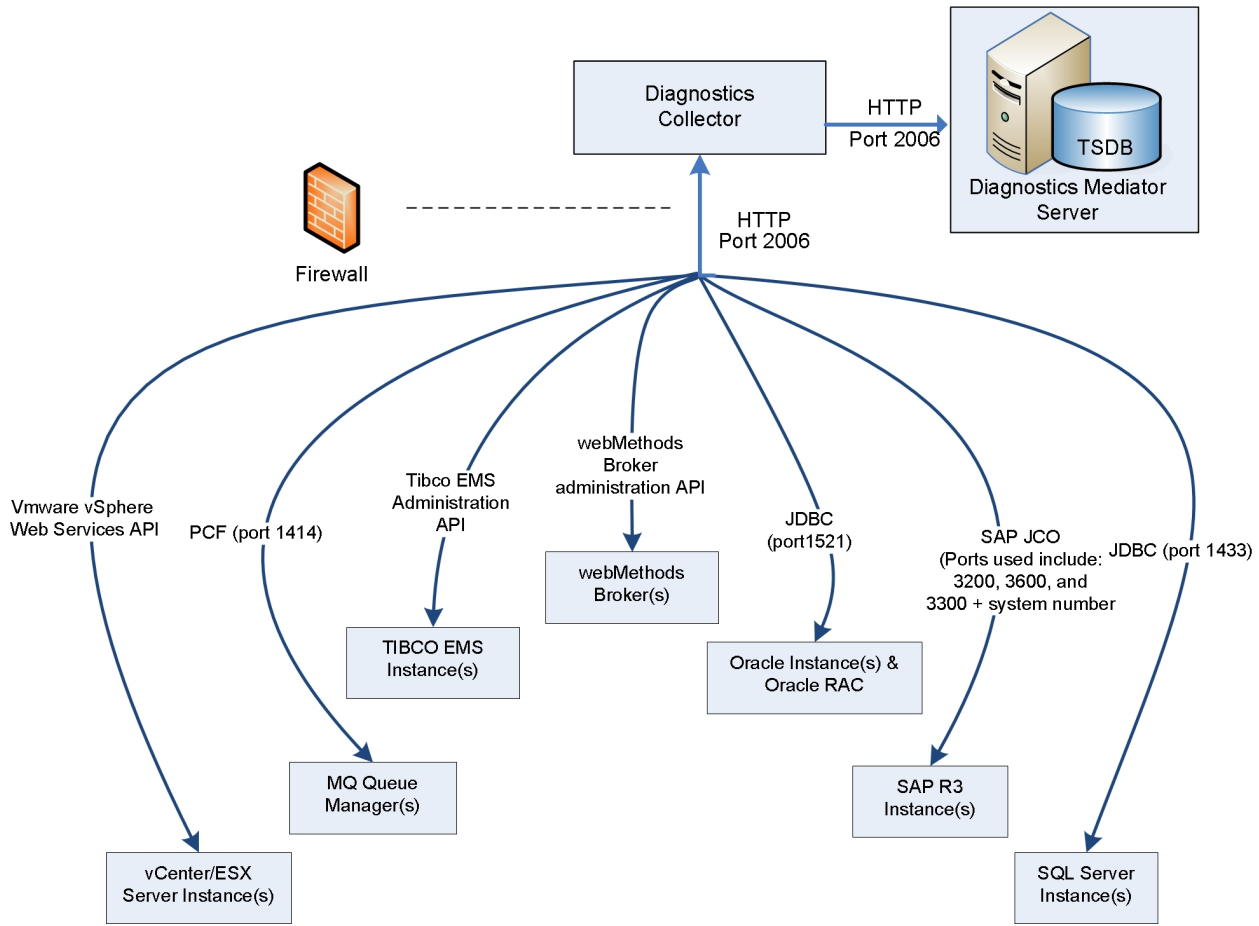
- Java agents use the HTTP port to send collected metrics to the Diagnostics mediator server.
- .NET agents with no Probe Aggregator use both the HTTP and TCP/IP ports to send collected metrics to the Diagnostics mediator server.
- The Probe Aggregator (if enabled) of a .NET agent uses the HTTP port to send collected metrics to the Diagnostics mediator server.
- Collectors use the HTTP port to send collected metrics to the Diagnostics mediator server.
- The Diagnostics commander server listens for HTTP on port 2006 (HTTPS 8443).
 - Diagnostics mediator servers use this port to send metrics to the Diagnostics commander server.
 - The Diagnostics Enterprise UI web browser client uses this port to access the Diagnostics Enterprise UI.
- The Probe Aggregator listens for HTTP on port 45000, and listens for TCP/IP on port 2626 .
 - .NET agents with a Probe Aggregator use both ports to send collected metrics to the Probe Aggregator.

If your deployment allows direct access to the Diagnostics Profilers UI, additional ports must be opened. Accessing the Diagnostics Profiler UI through the Diagnostics Enterprise UI does not require additional ports.

By default, the Java or .NET agent listens on ports 35000-35100 (HTTPS 45000-45100) for requests from the Diagnostics Agent Profiler UI web browser client. The actual ports on which you must allow communications depends on the port numbers that were enabled when the agent is configured for monitoring.



The Collector listens on various ports.



Configure Secured Communication

This security feature enables secured communication between all Diagnostics components without the need for SSL.

During the Diagnostics server installation, if you select the **commander** mode, the setup program will generate a **diagssso.properties** file which contains the security key in obfuscated format. This key will be used to communicate between different Diagnostics components such as mediator, collector, and probes.

To enable this feature, copy the **diagssso.properties** file to each of the components' **etc** folder. When there is communication between components, it checks to see if this file exists. If it exists, it will use enhanced security. If not, it will use the normal/default communication.

Note: The key gets generated during commander mode installation. Do not modify this key. We recommend that you maintain backup copy of this file.

Synchronize Time Between Diagnostics Components

For Diagnostics data to be stored and correlated properly, it is critical that time is synchronized between the Diagnostics components. To facilitate synchronization of data, the Diagnostics data is adjusted and saved to the synchronized GMT time of the Diagnostics Server in Commander mode. Synchronization makes it possible to display the data correctly for any local time in which the UI can be located.

The following sections describe how time synchronization works, and how to configure the components properly so that the time will be synchronized.

- ["Time Synchronization" below.](#)
- ["Configure Time Synchronization on the Diagnostics Server" on the next page.](#)

Probe collections running in VMware hosts have additional time synchronization requirements. See the Diagnostics Java Agent Guide.

Time Synchronization

Time synchronization in Diagnostics begins with the Diagnostics commander server determining the difference between its time and the GMT time provided by a designated **Time Source**. The **Time Source** to be used is set using the `timemanager.time_source` property in `<diag_server_install_dir>/etc/server.properties`.

The valid values for the `timemanager.time_source` property are:

- **NTP.** Indicates that an NTP Server is to be used as the source of GMT time. This is the default value. The NTP servers that are to be used are listed as values of the `timemanager.ntp.servers` property in `<diag_server_install_dir>/etc/server.properties`.

Note: Make sure that one of the NTP servers in the list can be contacted from the Diagnostics Server, or add your local NTP server as the first server in the list.

- **BAC.** Indicates that the registered BSM/APM gateway server is to be used as the source of GMT time.

Note: If BSM/APM is configured to use Database time, you should also configure the Diagnostics commander server to use this setting as the time source.

- **SERVER.** Indicates that the Diagnostics commander server is to be used as the **Time Source**. This should only be used when the Diagnostics Server is being used in Standalone mode.

The Diagnostics Servers that are in Mediator mode synchronize their time by establishing the time difference between the Diagnostics Server in Mediator mode and the Diagnostics Server in Commander mode.

If the Diagnostics Server in Commander mode did not yet synchronize with the **Time Source**, the Diagnostics Servers in Mediator mode are considered to be “unsynched.” The Diagnostics Servers in Mediator mode that are unsynched attempt to synchronize their time every 15 seconds until they succeed.

When a Diagnostics probe connects to a Diagnostics Server in Mediator mode or to a Diagnostics Server in Commander mode, the time difference is established between the Diagnostics Server and the probe.

If the probe attempts to connect to a Diagnostics Server that is still “unsynched,” the probe connection is not allowed and is dropped.

Because the data is stored based on the GMT, differences in time zones or daylight savings times for the various components are not an issue. For example, the data that is displayed in the Diagnostics UI can be adjusted to display correctly for the time zone in which the UI is running.

Note: All data is adjusted and saved to the synchronized GMT time of the Diagnostics Server in Commander mode. If the UI is running on a machine whose time was not synchronized properly with the **Time Source**, the data displayed in the UI appears shifted by the amount of time the UI machine is off from the synchronized GMT time.

Configure Time Synchronization on the Diagnostics Server

You can synchronize the Diagnostics commander server by performing the following procedure.

Note: Time Synchronization settings for Diagnostics Servers in Mediator mode are ignored because their time is automatically synchronized with the Diagnostics Server in Commander mode.

To ensure that time on the Diagnostics Server in Commander mode can be synchronized:

1. The default configuration for the Diagnostics Server is set such that the value of the **timemanager.time_source** property in **<diag_server_install_dir>/etc/server.properties** is **NTP**.

If the Diagnostics Server has an internet connection and the ability to connect to a server in the list of available NTP servers specified in the **timemanager.ntp.servers** property, the default configuration will work and no changes are necessary.

Because BSM/APM also uses NTP for time synchronization by default, this is the recommended setting.

2. If the Diagnostics Server does not have an internet connection or the ability to connect to the list of available NTP servers specified in **timemanager.ntp.servers** property, you *must* do one of the following:
 - Set up a local NTP server that can be contacted by the Diagnostics Server in Commander mode. List this local NTP server as the first entry in the **timemanager.ntp.servers** property in **<diag_server_install_dir>/etc/server.properties**.

Note: Have backup NTP servers in case the primary NTP server is not available.

- If you are using Diagnostics in a BSM/APM or Micro Focus Software as a Service (SaaS) environment, you can set the **timemanager.time_source** property in **<diag_server_install_dir>/etc/server.properties** to **BAC** to indicate BSM/APM. This causes the Diagnostics Server to connect to the registered BSM/APM core server to establish the time.

Note: To set up BSM/APM to use Diagnostics, see the APM-Diagnostics Integration Guide.

- If the Diagnostics Server in Commander mode is to be used in Standalone mode, with no intention of using it with BSM/APM, and there is no internet connection allowing time synchronization with an NTP server, you can set the **timemanager.time_source** property in **<diag_server_install_dir>/etc/server.properties** to **SERVER**. This causes the Diagnostics Server to use its own time as the **Time Source**.

Note: It is recommended that you do not change the value of the **timemanager.time_source** property in **<diag_server_install_dir>/etc/server.properties** once data is captured and persisted using the designated **Time Source**. Changing the **Time Source** after data is captured can result in a significant corruption to the data that was captured and persisted. This is because the data that was persisted might have been captured while the Diagnostics Server was not

synchronized with GMT. If the data that is captured later is captured while the Diagnostics Server is synchronized with GMT, the data could get re-aggregated multiple times or could get recorded into time buckets where it does not belong.

Configure the Diagnostics Mediator Server for a Large Deployment

If you are using a Diagnostics Server in Mediator mode with more than 200 probes, it is recommended that you make modifications to the default configuration of the Diagnostics Server.

Note: These changes to the configuration are not needed for the Diagnostics Server in Commander mode unless it also has probes assigned to it so that it also serves as a Diagnostics Server in Mediator mode.

Adjust the Heap Size

The size of the heap can impact the performance of the Diagnostics Server in Mediator mode. If the heap is too small, the Diagnostics Server in Mediator mode could “hang” for periods of time. If the heap is too large, the Diagnostics Server in Mediator mode could experience long garbage collection delays (especially if there aren’t enough CPU resources available such as multiple CPUs/cores or fast CPUs).

The default value for the heap size is 2560 MB. The heap size is set in the **server.nanny** file located at:

<diag_server_install_dir>\nanny\windows\dat\nanny for Windows

<diag_server_install_dir>/nanny/linux/launch_service/dat/nanny/ for Linux

Use the following VM argument to set the size (where ??? is the size in MB):

-Xmx???m

If you encounter problems with the Diagnostics Server in Mediator mode hanging, you can increase the heap size specified by updating the value specified in the **-Xmx???m** option.

To adjust the heap size of the Diagnostics Server in Mediator mode:

1. Use the following table to determine the amount of heap the Diagnostics Server in Mediator mode will need:

Number of Probes	Concurrent Users	Recommended Heap Size
Up to 200 Java Probes	10	2560 MB
Up to 400 Java Probes	10	8 GB
Up to 200 Java Probes	200	8 GB

Caution: Make sure the heap size does not exceed more than 75% of the physical memory of the machine. For example, if a machine has 8 GB, the heap size must not exceed 6 GB.

For VMware installations follow the best practices as described in VMware's "Enterprise Java Applications on VMware Best Practices Guide". In essence, use multiple vCPUs and fixed memory allocations (no ballooning or swapping to disk) and ensure installation of VMware Tools.

2. Open the **server.nanny** file that is to be edited. This file is located at:
<diag_server_install_dir>\nanny\windows\dat\nanny for Windows

<diag_server_install_dir>/nanny/linux/launch_service/dat/nanny/ for Linux

3. On the start_<platform> line that is appropriate, replace the heap size specified in the **-Xmx???m** option with the optimal heap size that you calculated:

-Xmx???m

Continuing the previous example, the current heap size, represented by ??? is replaced with 8 GB.

-Xmx8g

Before you modify this line in the **server.nanny** file, it will look like this:

```
start=C:\MercuryDiagnostics\Server\jre\bin\javaw.exe^ -server -Xmx2560m -
XX:+PrintGCDetails -XX:+PrintGCDateStamps -
Xloggc:C:\MercuryDiagnostics\Server\log\server_gclog.gc -XX:+UseGCLogFileRotation
-XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=5M -XX:+UseConcMarkSweepGC -
XX:+HeapDumpOnOutOfMemoryError -XX:+UseCompressedOops -DentityExpansionLimit=1 -
Dsun.net.client.defaultReadTimeout=70000 -
Dsun.net.client.defaultConnectTimeout=30000 -
Dorg.owasp.esapi.resources=C:\MercuryDiagnostics\Server\etc "-
javaagent:C:\MercuryDiagnostics\Server\probe\lib\probeagent.jar" "-
Xbootclasspath/p:C:\MercuryDiagnostics\Server\probe\classes\diag_
server\instr.jre" -classpath
"C:\MercuryDiagnostics\Server\lib\mediator.jar;C:\MercuryDiagnostics\Server\lib\c
ommon-
boot.jar;C:\MercuryDiagnostics\Server\lib\common.jar;C:\MercuryDiagnostics\Server
\lib\jsp-2.1.jar;C:\MercuryDiagnostics\Server\lib\jsp-api-
2.1.jar;C:\MercuryDiagnostics\Server\lib\common-
webapps.jar;C:\MercuryDiagnostics\Server\lib\mercury_picocontainer-1.1.jar"
com.mercury.opal.mediator.util.DiagnosticsServer
```

After you modify this line in the **server.nanny** file, it will look like this:

```
start=C:\MercuryDiagnostics\Server\jre\bin\javaw.exe^ -server -Xmx8g -
XX:+PrintGCDetails -XX:+PrintGCDateStamps -
Xloggc:C:\MercuryDiagnostics\Server\log\server_gclog.gc -XX:+UseGCLogFileRotation
-XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=5M -XX:+UseConcMarkSweepGC -
XX:+HeapDumpOnOutOfMemoryError -XX:+UseCompressedOops -DentityExpansionLimit=1 -
Dsun.net.client.defaultReadTimeout=70000 -
Dsun.net.client.defaultConnectTimeout=30000 -
Dorg.owasp.esapi.resources=C:\MercuryDiagnostics\Server\etc "-
javaagent:C:\MercuryDiagnostics\Server\probe\lib\probeagent.jar" "-
Xbootclasspath/p:C:\MercuryDiagnostics\Server\probe\classes\diag_
server\instr.jre" -classpath
"C:\MercuryDiagnostics\Server\lib\mediator.jar;C:\MercuryDiagnostics\Server\lib\c
ommon-
boot.jar;C:\MercuryDiagnostics\Server\lib\common.jar;C:\MercuryDiagnostics\Server
\lib\jsp-2.1.jar;C:\MercuryDiagnostics\Server\lib\jsp-api-
2.1.jar;C:\MercuryDiagnostics\Server\lib\common-
webapps.jar;C:\MercuryDiagnostics\Server\lib\mercury_picocontainer-1.1.jar"
com.mercury.opal.mediator.util.DiagnosticsServer
```

Adjust the Amount of Data Pulled from the Probe

Large call profiles require significant network bandwidth between the probe and server, and significant CPU resources on the server.

If the network becomes a bottleneck—for example, network utilization above 25% on a mediator as observed in Windows task manager, or probes report less than 100% availability although they were up—you should reduce the data that is generated via trimming, to enable compression, if the probe system's CPU is not fully used. You can also reduce the frequency of the data that the server pulls from the probe.

The main trimming parameters on the probe are:

- In the **capture.properties** file:
 - `maximum.stack.depth = 25`
 - `maximum.method.calls = 1000` (for example, can be set to 25 to limit overall number of methods in a Call Profile)
 - `minimum.method.latency = 51ms`
- In the **dispatcher.properties** file:
 - `minimum.fragment.latency = 51ms` (for example, can be increased to 101ms). Note that by default, trimming doesn't affect synthetic transactions (BPM/vuGen/LoadRunner/Performance Center) so all these server requests are reported.

For more information on trimming, see ["Configure Server Request Name Based Trimming" on page 71](#) for the server, "Configuring Latency Trimming and Throttling" and "Configuring Depth Trimming" in the Diagnostics .NET Agent Guide, "Controlling Automatic Method Trimming on the Agent" in the Java Agent Guide.

To enable compression, on the probe set **webserver.properties: rhttp.gzip.replies = true**. This reduces network traffic on the server significantly. However, the probe (and server) require additional CPU for compression.

Another way of decreasing network traffic is to change the frequency that data is pulled from the probe. By default, trends are pulled every 10 seconds and trees (Call Profiles) are pulled every 45 seconds. To lower the frequency for call trees, change **probe.trees.pull.interval** on the mediator in the **server.properties** file—for example, 90 seconds or 240 seconds depending on how many methods a Call Profile contains. First, lower the pull frequency of call trees. If this is not enough, lower the trend pull frequency by changing **probe.trends.pull.interval**—for example, 20 seconds.

Changing any of these parameters requires restarting the probe or server.

Additional Adjustments

- If more than 200 probes are connected, make the following adjustments:
 - Increase the number of threads used for pulling data from the probe. For each mediator, in the **server.properties** file set **probe.pull.max.threads=70** and restart the server.
 - Increase the number of threads available for jetty by setting **jetty.threads.max=1200** in the **webserver.properties** file.
- If call tree and trend files (see also ["Diagnostics Data Management" on page 142](#)) become too large

(greater than 4 GB) in their uncompressed state, offload some of the probes to a new mediator. Otherwise, the aggregation and compression of the files could start to lag due to the large amount of data.

- When many probes are connected to a server, the default purging setting of 30 GB might not be enough. For example, for 400 probes, set **persistence.purging.threshold** to 50 GB. For more information, see ["Data Retention" on page 146](#).
- Increase the maximum load for each mediator by increasing the value in the following parameters in the **server.properties** file:
 - mediator.max.load.count.5s=800000
 - mediator.max.load.count.20s=1000000

The value depends on the number of monitored server requests per probe, server request depth (methods in the call profile), number of trended methods, number of outbound calls, and other entities monitored by Diagnostics. For further details on the mediator.max.load.count parameters, see ["Default Auto-Assignment Algorithm" on page 78](#).

- If more than 20 concurrent users are connected to the UI, make the following adjustments in the **/etc/ui.properties** file on the server:
 - Change the **ui.topn** setting to **10** (up.topn=10).
 - Uncomment the following lines by removing the hash (#) sign:

```
#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.SummarySetDataHandle=2.5

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.SummaryDataHandle=2

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.TrendDataHandle=2

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.EntityContentsDataHandle=2.5

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.GetAlertEventsHandle=2

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.GetMetaDataHandle=2

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.BizTxnTopologyDataHandle=3

#ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.
builder.handles.ServerRequestTopologyDataHandle=3
```

Note: Open (static) pages are counted as concurrent users due to auto-refresh.

Optimize the Diagnostics Server in Production to Handle More Probes

The number of probes that a single diagnostic server process can handle depends largely on the number of unique server requests per 5-minute interval, and the number of methods and layers in each server request. The following optimizations increase the number of probes that can be handled per server process.

- The default setting is for the diagnostic server to pull the trends from each probe every 10 seconds, and the trees from each probe every 45 seconds. If a single diagnostic server process is handling more than 100 probes, this could be optimized such that the trends and trees are pulled less often. A suggested optimal setting in production is a 30-second trend pull interval, and a 120-second tree pull interval. These values can be configured in `<diag_server_install_dir>\Server\etc\server.properties` as follows:

```
# The interval at which to pull trends from probes
probe.trends.pull.interval = 30s

# The interval at which to pull trees from probes
probe.trees.pull.interval = 120s
```

- The maximum heap size of the server process is determined by the `-Xmx` parameter in the server's startup script. The default setting is 2560 MB for maximum heap size. Increase the maximum heap size according to the load from the probes. The suggested values for maximum heap size, based on the number of probes to be handled, is available in "[Product Overview](#)" on page 11.
- A 1 Gbps link is strongly recommended in production for the diagnostic server when the server is handling more than 30 probes.
- If a single server process is handling more than 200 probes, increase the number of jetty threads. The general rule of thumb for sizing the number of threads is twice the number of probes + 40. The default value is 500. The number of jetty threads can be increased by modifying the `jetty.threads.max` property in `<diag_server_install_dir>\Server\etc\webserver.properties`; for example:

```
jetty.threads.max=500
```

Note: When a large number of probes (approximately 1000) is in use, the System Health view in the Diagnostics Server UI can sometimes take several minutes to load. Distributing the probes over more mediators does not solve this problem as this view shows all probes from all mediators.

Override the Default Diagnostics Server Host Name

When a firewall or NAT is in place, or the host for the Diagnostics Server in Mediator mode was configured as a multi-homed device, the Diagnostics Server in Commander mode might not be able to communicate with the Diagnostics Server in Mediator mode using the host name assigned when the Diagnostics Server in Mediator mode was installed. The `registered_hostname` property enables you to override the default host name the Diagnostics Server in Mediator mode uses to register itself with the Diagnostics Server in Commander mode.

To override the default host name for a Diagnostics Server in Mediator mode, set the `registered_hostname` property located in `<diag_server_install_dir>/etc/server.properties` to an alternate machine name or IP address that will allow the Diagnostics Server in Commander mode to communicate with the Diagnostics Server in Mediator mode.

Change the Default Diagnostics Server Port

If the configuration of the Diagnostics Server host does not allow the default Diagnostics port to be used, choose a different port for the Diagnostics Server communications with the probes and other Diagnostics Servers.

Note: Make sure that the new port number is not already used by another application and that the other Diagnostics components can communicate with this port.

If you decide to use an alternative port number after you deploy Diagnostics, you must update the properties in the following table for each of the indicated components in your deployment with the new port number to ensure that the proper communications can take place.

Component Type	Properties
Diagnostics commander server	<p><diag_server_install_dir>/etc</p> <ul style="list-style-type: none"> • webservice.properties – jetty.port • server.properties – commander.url • probe/etc/dispatcher.properties – registrar.url
Diagnostics mediator server	<p><diag_server_install_dir>/etc</p> <ul style="list-style-type: none"> • server.properties – commander.url <p><diag_server_install_dir>/probe/etc</p> <ul style="list-style-type: none"> • dispatcher.properties – registrar.url
Probes	<p><probe_install_dir>/etc</p> <ul style="list-style-type: none"> • dispatcher.properties – registrar.url

Migrate Diagnostics Server from One Host to Another

The following procedure shows how to migrate your Diagnostics Server from one host to another and assumes the new host name is different from the old host name.

To migrate a Diagnostics server from one host to another:

1. Ensure that the existing Diagnostics Server has been shut down by verifying that there are no java/javaw processes in your process list. On Windows systems, you can use the Task Manager to do this and on UNIX systems, you can use ps.
2. Unregister the Diagnostics Commander Server from BSM/APM if an integration exists.
3. Install the new Diagnostics Server on the new host.
4. On Windows, the Diagnostics Server is started automatically when the installer finishes so you must shut down the Diagnostics Server.

On UNIX, the server is not automatically started so you do not need to shut it down.

Ensure that the Diagnostics Server has been shut down by verifying that there are no java/javaw processes in your process list. On Windows systems, you can use the Task Manager to do this and on UNIX systems, you can use ps.

Be sure you know the host name of the old Diagnostics Server (you can find the name in the **/archive** directory).

5. Delete the `<diag_server_install_dir>/archive` directory on the new Diagnostics Server.
6. Copy the `<diag_server_install_dir>/archive` folder and all subfolders from the old server into the new server `<diag_server_install_dir>/`.
7. If the host name for the new Diagnostics Server is different than the host name for the old Diagnostics Server, you must rename `<diag_server_install_dir>/archive/mediator-<host-name>` so that `<host-name>` reflects the new Diagnostics Server host name. For example, if your old host name was `oldhost` and the new host name is `newhost` you would change

```
<diag_server_install_dir>/archive/mediator-<oldhost> to <diag_server_install_dir>/archive/mediator-<newhost>
```
8. Delete the `<diag_server_install_dir>/storage/` directory for the new Diagnostics Server.
9. Copy the `<diag_server_install_dir>/storage/` folder and all subfolders from the old server into the new server `<diag_server_install_dir>/`.
10. On the new server, rename `<diag_server_install_dir>/storage/server-<hostname>` so that `<host-name>` reflects the new Diagnostics Server host name. For example, if your old host name was `oldhost` and the new host name is `newhost` you would change

```
<diag_server_install_dir>/storage/server-<oldhost> to <diag_server_install_dir>/storage/server-<newhost>
```
11. Copy the `<diag_server_install_dir>/etc` folder from the old server into the new server `<diag_server_install_dir>/` and copy the new license to `etc` folder.
12. Start the new Diagnostics server and register it in BSM/APM if an integration is required.
13. If the new server was the Commander then on all the mediators, you need to scan the `etc` folder and change the old server name to the new server name. Double-check the `dispatcher.properties` file to make sure the commander server hostname changed. Then restart all the mediators.

 There is no change required on the probe side unless the probe is directly reporting to the commander server or you are migrating the mediator server the probe is connected to. If that is the case, scan the `etc` folder on the probe system and change the old server name to the new server name (double-check the `dispatcher.properties` file to make sure the mediator server hostname changed).

Configure the Diagnostics Server for Multi-Homed Environments

The machines that host the Diagnostics Server can be configured with more than one Network Interface Card (NIC). The Diagnostics Server process listens on all interfaces on its host. Some customer environments do not allow applications to listen on all network interfaces on a machine. If your environment has this restriction, use the following instructions to configure the Diagnostics Server to listen on specific network interfaces.

Set the Event Host Name

If the Diagnostics Server host has multiple network interfaces, and you want to specify the hostname that the Diagnostics Server will listen on, you must set the `event.hostname` property.

This property can be found in:

`<diag_server_install_dir>/etc/server.properties`

Uncomment the property `event.hostname` and specify the hostname value.

By default, the `event.hostname` property is not set. This means that the Diagnostics Server will listen on all hostnames.

Modify the jetty.xml File

The **jetty.xml** file has a section that defines the interfaces on which the Diagnostics Server is permitted to listen. By default, the **jetty.xml** file included with the Diagnostics Server has no listeners defined. The Diagnostics Server listens on all of the interfaces.

To configure the Diagnostics Server to listen on specific network interfaces on a machine:

1. Open `<diag_server_install_dir>/etc/jetty.xml` and locate the following line:

```
<Configure class="org.mortbay.jetty.Server">
```

2. Add the following block of code after this line, changing the `<Set name="Host">.....</Set>` to contain the NIC's IP address.

```
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SocketListener">
      <Set name="Host">127.0.0.1</Set>
      <Set name="Port"><SystemProperty name="jetty.port" default="2006"/></Set>
      <Set name="MinThreads">1</Set>
      <Set name="MaxThreads">5</Set>
      <Set name="MaxIdleTimeMs">30000</Set>
      <Set name="LowResourcePersistTimeMs">5000</Set>
      <Set name="ConfidentialPort">8443</Set>
      <Set name="IntegralPort">8443</Set>
    </New>
  </Arg>
</Call>
```

3. Repeat the previous step adding a new copy of the block of code and setting the IP address for the NIC for each interface on which the Diagnostics Server is to listen.

Make sure that the **</Configure>** tag follows the listener code for the last NIC.

Note: Make sure that components that access the Diagnostics Server can resolve the hostnames of the Diagnostics Server to the IP address that you specify in the **jetty.xml** file for the host values. Some systems could resolve the host name to a different IP address on the Diagnostics Server host. For more information, see ["Override the Default Diagnostics Server Host Name" on page 66](#).

Sample jetty.xml File

The following example shows the **jetty.xml** file for the Diagnostics Server, where the Diagnostics Server will listen on loopback and one IP address on the system.

```
<!-- Configure the Jetty Server -->
<!-- ===== -->
<Configure class="org.mortbay.jetty.Server">
<!-- ===== -->
<!-- Configure the Request Listeners -->
```

```

<!--===== -->
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SocketListener">
      <Set name="Host">127.0.0.1</Set>
      <Set name="Port"><SystemProperty name="jetty.port" default="2006"/></Set>
      <Set name="MinThreads">1</Set>
      <Set name="MaxThreads">5</Set>
      <Set name="MaxIdleTimeMs">30000</Set>
      <Set name="LowResourcePersistTimeMs">5000</Set>
      <Set name="ConfidentialPort">8443</Set>
      <Set name="IntegralPort">8443</Set>
    </New>
  </Arg>
</Call>

<-Listen on specific IP Address on this machine for incoming Commander calls->
<Call name="addListener">
  <Arg>
    <New class="org.mortbay.http.SocketListener">
      <Set name="Host">10.241.3.109</Set>
      <Set name="Port"><SystemProperty name="jetty.port" default="2006"/></Set>
      <Set name="MinThreads">1</Set>
      <Set name="MaxThreads">5</Set>
      <Set name="MaxIdleTimeMs">30000</Set>
      <Set name="LowResourcePersistTimeMs">5000</Set>
      <Set name="ConfidentialPort">8443</Set>
      <Set name="IntegralPort">8443</Set>
    </New>
  </Arg>
</Call>
</Configure>

```

Reduce the Diagnostics Server Memory Usage

The Transaction Timeout Period is a safety mechanism that prevents the Diagnostics Server from using excessive amounts of memory because it is holding on to old data for too long. The Diagnostics Server holds on to all of the information it receives for a transaction until it receives the End of Transaction Notification (ELT), which tells the Diagnostics Server the transaction is complete. The timeout period for a transaction is reset each time the Diagnostics Server receives data for the transaction.

If the machine on which the Diagnostics Server in Commander mode is running is overloaded (CPU is heavily loaded or there are too many transactions per second for it to handle), or if there are network connectivity issues between the Load Generators or BSM/APM and the Diagnostics commander server, or between Business Process Monitor and BSM/APM, the Diagnostics Server might not receive the ELT that lets it know when a transaction ended. If the ELT is not received by the time the transaction timeout period expires, the Diagnostics Server assumes that the ELT is not coming and proceeds to process the data for the transaction and free the memory the transaction data is using.

The **correlation.txn.timeout** property sets the duration of the transaction timeout period. If you experience out-of-memory conditions in the Diagnostics Server, you could reduce the transaction timeout period so that the Diagnostics Server waits less time for the end of a transaction. Use caution when adjusting the value of this property because multiple probes could be sending data to the Diagnostics Server, and an active transaction could be idle in one Diagnostics Server. Setting the value of this property too low can cause transactions to be reported incorrectly. If you need to reduce the value of this property, set it to 90 seconds more than the longest transaction in your test.

Configure URI Trimming on the Server

To enable URL trimming capability on the server, modify the following properties in **<server_install_dir>/server.properties**.

Modify the properties below to limit the number of different fragments or methods collected. If there are dynamically created class and method names, you can fold down the names using regular expression substitution.

```
fragment.name.pattern.replace=s/detail.*/trimmeddetail/
method.name.pattern.replace=s/dynamic_method_name.*/something_else/
method.class.pattern.replace=s/dynamic_class_name.*/something_else/
```

Configure Server Request Name Based Trimming

Server Request name based trimming lets you configure Diagnostics to filter out server requests that appear to be causing Diagnostics Server performance issues without changing the configuration or the instrumentation used by the probes.

Note: Server request name-based trimming is not intended to be used instead of the latency and depth trimming you configure for the probes.

Using the **trim.fragment** properties in the **<diag_server_install_dir>\etc\trimming.properties** file, you can specify the names of the server request fragments that Diagnostics is to trim. Diagnostics trims the fragments for both Real User and Virtual User server requests.

By default, the properties **trim.fragment.1** and **trim.fragment.2** are commented out in **trimming.properties**. To specify a fragment to be trimmed, uncomment one of the properties and type the fragment name that is to be trimmed as it is listed in the Diagnostics views. If more than two fragments need to be trimmed, create additional **trim.fragment** properties. Make sure to increment the number at the end to ensure that each property name is unique. For example, the next **trim.fragment** property would be named **trim.fragment.3**.

Events and fragments that are trimmed as a result of these property settings are counted in the dropped event and dropped fragment counts.

Automate the Composite Application Discovery in Diagnostics

Composite Application Discovery (CAM) provides a convenient way to group application servers (probes) and to continuously detect new components that are connected to these application servers by following the calls a probe is making to these other components.

In addition to configuring applications in the UI, Diagnostics provides scripting support for CAM. This allows the dynamic creation of new applications based on newly added probes outside the UI.

Scripting Applications

Scripts that are used to create new applications are stored on each mediator in the **etc/appDiscoveryRules.properties** file. The scripts in the file only run locally (on the machine on which they are stored) so if you want a script to run across a distributed environment, you must copy the script to the **appDiscoveryRules.properties** file on each mediator. By default, the scripts in the file are run automatically every 24 hours, but are also run a few minutes after you make changes to the file. Each line in a script is in the format **query=code**.

Queries

A query is any regular Diagnostics XPath query. Note that:

- You must always specify **/groupby**.
The groupby path is used to query the Diagnostics data model and select instances. This query path is used in scripts for application discovery.
The groupby definition periodically queries all probes on a mediator and executes the script (Java code) against the returned probe names.
- You cannot cross groupbys
- You must escape equal signs (=) and space characters.

Code

Code can be any valid Java code. Note that:

- If the code spans multiple lines, you must escape each **CRLF** with a backslash (\).
- Different methods are available on each node type, but `getName()` and `getDisplayname()` will always work.
- To place a matched entity in an application, the code must set the **uid** variable to the globally unique ID for that application. If an application with that UID does not exist, it is created.
- When creating an application, you can set the following variables to initialize the application as required:
 - **name**. The user visible name of the application. Defaults to the UID if not specified.
 - **path**. The groups in which the application should reside.
 - **viewUsers**. The set of users who can view the application. Defaults to `(any_diagnostics_user)` if not specified, which means that everyone can see the application.
 - **changeUsers**. The set of users who can add and remove entities from the application. Defaults to `(any_diagnostics_superuser)`.
 - **editUsers**. The set of users who can change the UI screens visible inside the application. Defaults to `(any_diagnostics_user)`.

The following examples show various scripts and how they create applications based on different data. The first four examples are included in the **appDiscoveryRules.properties** file.

• Example 1 - Create applications based on URLs

If you consistently deploy applications to context URLs, it may be convenient to automatically create applications based on those URLs.

In the following example, from the URLs

```

/sales/login.jsp
/sales/browse.jsp
/support/viewKnowledgeBaseEntry.jsp
/support/index.jsp

```

two applications, **sales** and **support**, are created in the **Newly Detected Applications** group and can be viewed by Diagnostics administrators only. (The assumption is that an administrator will periodically review newly discovered applications and will grant permissions to the relevant teams and move the application to an appropriate group.)

```

/groupby[name\='Default\ Client']/probegroup/index[name\='rollup_fragment']
/fragment=\
String fname = fragment.getUri();\
if( fname != null ) { \
    int idx = fname.indexOf('/'); \
    if( idx == 0 && fname.length() > 1 ) {\
        fname = fname.substring(1);\
        idx = fname.indexOf('/');\
    }\
    if( idx > 0 ) { \
        String firstPart = fname.substring(0,idx);\
        uid = name = firstPart;\
        path = "Newly Detected Applications";\
        viewUsers = "(any_diagnostics_admin)";\
        editUsers = "(any_diagnostics_admin)";\
    } \
}

```

• Example 2 - Place all probes with a particular naming pattern into an application

The following example illustrates an easy way to create new applications based on parts of the probe name. In the example, the code creates an application with the name **Sales** for all probes that start with **cs_**.

```

/groupby[name\='Default\ Client']/probegroup/probe=\
String probeName = probe.getName();\
if( probeName.startsWith("cs_") ) {\
    uid=name="Sales";\
}

```

• Example 3 - Use application names reported to BEA WebLogic

Diagnostics can detect the application names reported to BEA WebLogic. The following example shows how to auto-create applications based on the name of the EAR file deployed into BEA WebLogic.

```

/groupby[name\='Default\ Client']/probegroup/index[name\='rollup_fragment']
/fragment[notequals(applicationName,'')]=\
String appName = fragment.getApplicationName();\
if( appName != null ) { \

```

```

uid = name = appName;\
path = "BEA WebLogic Applications";\
}

```

• Example 4 - Create specific applications based on URL prefixes

This example shows the creation of two applications, **DXP** and **CXP**, in the **Detected Applications** group and with view and edit permissions for Diagnostics administrators only, based on detected URL prefixes.

```

/groupby[name\='Default\ Client']/probegroup/index[name\='rollup_fragment']
/fragment=
String fname = fragment.getUri();\
if( fname != null ) { \
    if( fname.startsWith("/OA_HTML/qot") ) { \
        uid = name = "DXP";\
        path = "Detected Applications";\
        viewUsers = "(any_diagnostics_admin)";\
        editUsers = "(any_diagnostics_admin)";\
    } else if( fname.startsWith("/OA_HTML/ibe") || fname.startsWith("/OA_
HTML/emc_ibe")) { \
        uid = name = "CXP";\
        path = "Detected Applications";\
        viewUsers = "(any_diagnostics_admin)";\
        editUsers = "(any_diagnostics_admin)";\
    }\
}
}

```

For details on probe groups and assigning privileges, see ["Assigning Privileges for Probe Groups" on page 110](#).

• Example 5 - Create the same application under different groups

This example shows the same application being created under more than one group. For example, when different divisions in an organization use the same application. The application name is created based on a URL prefix. Note that while the application name can be the same in each group, the application UID must be unique.

```

/groupby[name\='Default\ Client']/probegroup/index[name\='rollup_fragment']
/fragment=
String fname = fragment.getUri();\
if( fname != null ) { \
    int idx = fname.indexOf('/'); \
    if( idx == 0 && fname.length() > 1 ) {\
        fname = fname.substring(1);\
        idx = fname.indexOf('/');\
    }\
    if( idx > 0 ) { \
        String firstPart = fname.substring(0,idx);\
        if( fname.startsWith("/Cars/sells") ) { \
            name = firstPart;\
        }\
    }\
}
}

```

```

        path = "Cars";\
        uid = path + "_" + firstPart;\
        viewUsers = "(any_diagnostics_admin)";\
        editUsers = "(any_diagnostics_admin)";\
    } else if( fname.startsWith("/TV/sells")) { \
        name = firstPart;\
        path = "TV";\
        uid = path + "_" + firstPart;\
        viewUsers = "(any_diagnostics_admin)";\
        editUsers = "(any_diagnostics_admin)";\
    }\
} \
}

```

Further it is possible to automatically include all related probe entities such as Server Requests and SQL statements. To do this, set the variable **discoveryPolicies** with the value **"applyAppFilterToProbeContents"**:

If **discoveryPolicies** variable is not set, then the default values for the discovery policies (**Add all Connected probes to application** and **Add probe contents to application**) will be set from the value of the `discoveryPoliciesDefaultValue` property in the `<Commander-Server>\etc\ui.properties` file. This applies for applications created from the UI or from the `appDiscoveryRules.properties` file.

```

/groupby[name='Default\ Client']/probegroup/probe=\
  String probeName = probe.getName();\the
  if( probeName.startsWith("cs_") ) {\
    uid=name="Sales";\
    discoveryPolicies="applyAppFilterToProbeContents"; \
  }

```

How to move Composite Applications Between Environments

The scripting approach provides an easy way to move applications between environments such as from QA to production. All application definitions, however, need to be created using the script. One master script can be used on all mediators even if the probes are not reporting to this mediator.

It is important to use a naming scheme that works between production and pre-production. This can be achieved by:

- Putting probes in specific probe groups that are constant between production and pre-production
- Using a probe naming convention that allows the script to create an application name as shown in the example above.

If the probes are in the same probe group and this name is constant between environments (but the probe name changes), use **probegroup.getName()** in the script to access the probe group name:

```

/groupby[name='Default\ Client']/probegroup/probe=\ String probegroupName =
  probegroup.getName(); \
  String probeName = probe.getName();\

```

```

if( probegroupName.startsWith("cs") ) { \
    uid=name="Sales"; \
    discoveryPolicies="applyAppFilterToProbeContents"; \
} \
else if ( probegroupName.startsWith("is") ) { \
    uid=name="Information Systems"; \
    discoveryPolicies="applyAppFilterToProbeContents"; \
}

```

This script is generic and can be exchanged between environments.

Prepare a High Availability Diagnostics Server

If your Diagnostics deployment requires that the Diagnostics Server have high availability, you can create a standby Diagnostics Server for each Diagnostics Server. The standby is then ready to be used during a hardware failure or other problem with the host of the Diagnostics Server.

Create a Standby Diagnostics Server

You can create a standby for each Diagnostics Server by installing the Diagnostics Server onto a standby machine and then periodically replicating the primary Diagnostics Server data into the standby Diagnostics Server.

To configure a standby Diagnostics Server:

1. Install the Diagnostics Server onto the standby machine. Make sure that the version of the Diagnostics Server to be installed on the standby server is the same as the Diagnostics Server on the primary server.
2. Schedule a periodic remote backup of the primary server into the standby server. For details on remote backup, see ["Backing up Data Remotely" on page 150](#).

Failover to the Standby Diagnostics Server

If the host for the primary Diagnostics Server fails, configure the standby Diagnostics Server so that it can begin to function as the primary Diagnostics Server.

To make the standby Diagnostics Server the primary Diagnostics Server:

1. Change the hostname of the standby Diagnostics Server to match the hostname of the failed host of the primary Diagnostics Server. This allows the probes to reconnect to the Diagnostics Server when it is started.
2. Start the standby Diagnostics Server as a Windows Service, or use the **bin/server.sh** or **bin\server.cmd** scripts. The probes reconnect to the Diagnostics server. Whenever a probe loses its connection to its Diagnostics Server it attempts to reconnect approximately every 30 seconds.
3. The standby Diagnostics Server is now the primary Diagnostics Server. Configure a new standby Diagnostics Server as described in ["Create a Standby Diagnostics Server" above](#).

Note: When the failed Diagnostics Server host is recovered, do not make it the primary Diagnostics Server because it loses any data gathered from the probes while the new primary Diagnostics Server is being used.

Probe Registration Auto-Assignment for Large Deployments

This feature enables multiple Java probes in a large deployment to be automatically assigned to any mediator in the deployment.

Learn About

This section includes:

- ["Automated Diagnostics Server Assignments Overview" below](#)
- ["Default Auto-Assignment Algorithm" on the next page](#)
- ["Rule Based Auto-Assignment" on page 79](#)
- ["Failure Conditions & Forced Reassignment" on page 82](#)

Automated Diagnostics Server Assignments Overview

In a large deployment environment, probes can be assigned to any mediator in the deployment. The assignment can be to a specific mediator or to one that is automatically assigned by the commander.

Probe auto-assignment to a mediator is beneficial in deployments where it does not matter which probe is assigned to a specific mediator.

The probe auto-assignment feature uses a number of properties in the following file:

<Agent host machine>/etc/dispatcher.properties

- **commander.registrar.url** – The Commander Registrar URL for Probe to Mediator Auto-Assignment. If the **registrar.url** property is empty, the commander specified by the **commander.registrar.url** property is requested to auto-assign a mediator for the probe. This auto-assignment feature requires the following properties to be configured in the **server.properties** file on each participating Diagnostics Server:
 - **commander.max.load.count.5s**
 - **commander.max.load.count.20s**
 - **mediator.max.load.count.5s**
 - **mediator.max.load.count.20s**
- **registrar.url** – This property should be set to blank initially, and should not be manually modified if you want to use auto-assignment.
- **always.use.commander.registrar.url** – By default, the auto-assigned mediator will be recorded within this file by overwriting the **registrar.url** property. Thus the auto-assignment will take place only once. This may be inconvenient, if this property file is shared by multiple probes. When the **always.use.commander.registrar.url** property is set to **true** and **commander.registrar.url** is set, the **commander.registrar.url** will always be used for auto-assignment. The auto-assignment will be maintained only at runtime, and the **registrar.url** property in this file will be ignored and will not be updated.
- **force.auto.assign** – If this property is set to **True**, if the probe has a previously assigned mediator and the load on the mediator is over the maximum load limit, the commander will not assign that mediator, but instead will return a new, mostly-filled mediator.

If this property is set to **False**, the system does not check whether the mediator is over the load limit. The default value is **False**.

Note: After the probe re-assignment, due to mediator expiration or overload, the probe has a new entity ID. This means that the CAM applications, alerts, and custom views/snapshots will be impacted.

- For alerts: you must manually export all rules and then re-import them.
- For applications: you must manually re-add those probes into the existing application with the same probe name.

The following table shows the relationship of these properties and the behavior when certain values are set for them:

always.use.commander.registrar.url	commander.registrar.url	registrar.url	Behavior
true	Contains a valid registrar url	Any setting	Ignores the setting in registrar.url and uses the commander.registrar.url setting to ask the commander for the registrar url to be used. Nothing is written to registrar.url.
false	Contains a valid registrar url	Blank	Uses the commander.registrar.url to ask the commander for the registrar.url to be used. registrar.url is rewritten with the new registrar.url provided by the commander.
false	Contains a valid registrar url	Any other setting	registrar.url is used for registration in the regular way.

Note: Probe auto-assignment requires the **commander.max.load.count.5s**, **commander.max.load.count.20s**, **mediator.max.load.count.5s**, and **mediator.max.load.count.20s** properties to be configured in the server.properties file on each participating Diagnostics Server. For details, see "[Default Auto-Assignment Algorithm](#)" below.

In addition to the default auto-assignment algorithm, you can also configure Diagnostics to assign probes to mediators based on rules that you create. For details, see "[Rule Based Auto-Assignment](#)" on the next page.

Default Auto-Assignment Algorithm

Auto-assignment attempts to "fill" a mediator before assigning the next one. This follows a "Most Filled" algorithm meaning the Commander will choose the mediator (or the Commander itself, if configured to accept probe assignments) that appears to be most full. This is to help keep probes assigned together first to the same mediator, since only those probes belonging to the same mediator can participate in X-VM correlation of call profiles.

The auto-assignment algorithm is affected by the following properties in the **server.properties** file of the server:

- **commander.max.load.count.5s** – The maximum number of Active Entities (a metric measured by the internal server probe that reflects the amount of data currently being processed in the server's Online Cache) that the Commander can hold when trend data is sampled for 5 second intervals.
- **commander.max.load.count.20s** – The maximum number of Active Entities that the Commander can

hold when trend data is sampled for 20 second intervals.

- **mediator.max.load.count.5s** – The maximum number of Active Entities that a mediator can hold when trend data is sampled for 5 second intervals.
- **mediator.max.load.count.20s** – The maximum number of Active Entities that a mediator can hold when trend data is sampled for 20 second intervals.

Note:

- If the above properties are commented out or set to blank or zero, then that server will not participate in auto-assignment.
- A server uses the settings appropriate to its role. A Commander uses the `commander.max.load.count` settings and a mediator uses the `mediator.max.load.count` settings.

- **max.load.threshold** – The threshold (default 0.9) for the relative mediator load (as a ratio of the current load count compared to the configured maximum load count) after which a server is considered to be fully utilized. For example, if the current load is 50,000 and `max.load.count` is set to 100,000, the current relative mediator load is 0.5 and the server is not considered fully utilized. Once the load increases to 90,000 the server is fully utilized and is no longer considered for auto-assignment.
- **mediator.load.query.time.frame** – The commander uses this property to specify the granularity of the load query that it sends to the mediator for load calculations. The default value is 1 day.
- **mediator.expiration.time** – This property is considered only when `always.use.commander.registrar.url` is set to `True`. This property specifies the time period any probe waits in auto assignment mode before asking the commander for another mediator when the current mediator is down. The default value is 24 hours.

Note: A server may be over allocated in the following circumstances:

- All servers are already at full capacity.
- Previously assigned probes are temporarily inactive.
- The commander or mediator `max.load.count` properties (described above) previously had a higher value.
- A newly assigned probe increases the load to more than configured by the maximum load threshold.

These properties define the maximum number of active entities that a Commander or mediator can handle and determine the total number of probes that can be assigned. The load is recorded on each Diagnostics Server, so they can have their own independent maximum load settings (some with higher capacity than others).

The Commander uses these capacity values to determine the utilization percentage of each server to execute the "Most Filled" algorithm. Typically, the Commander capacity would be set to 0 as we usually recommend only mediators handle the probes to let the Commander be dedicated to its Commander and UI query tasks.

Auto-assignment will not block a probe from being assigned if all servers reach their defined capacity. When this state is reached, the algorithm changes over to a "Least Filled" algorithm. It will select the server which is the least "over capacity" based on utilization percentage.

Rule Based Auto-Assignment

Rule based auto-assignment uses metadata (name/value pairs) sent by the probe to select a mediator based on rules. The metadata sent by the probe is used as input to match against the rules and if a match is found, a set of mediators (organized in Mediator Groups) is returned as potential candidates for assignment. A mediator is then chosen from the set of candidates using the default auto-assignment algorithm.

To use rule based auto-assignment, you must configure the following:

On the probe machine:

1. **Configure metadata.** In the **<Agent host machine>/etc/dispatcher.properties** file, configure the metadata for the probe. The metadata consists of a list of name/value pairs. The value metadata can be a regular string, or can contain system properties or environment variable references. For example:

```
metadata=Application=App1;Env=Dev;key1=val1_${HOSTNAME};key2=val2_${OS}
```

Tip: You can enter a large amount of text over multiple lines by ending each line with a backslash (\) character.

You can also specify metadata on the command line as a system property. For example:

```
-Dprobe.metadata=Application=App1;Env=Dev
```

Note: If no metadata is sent by a probe, the default auto-assignment algorithm is used to assign a mediator.

2. **Enable auto-assignment for the probe.** In the **<Agent host machine>/etc/dispatcher.properties** file, set the **commander.registrar.url** to the relevant Commander URL and optionally, set **always.use.commander.registrar.url** to **true**. For details of these settings and how they affect auto-assignment, see ["Automated Diagnostics Server Assignments Overview" on page 77](#).

On the Commander machine:

1. **Define mediator groups.** In the **/etc/MediatorGroups.xml** file, configure the mediator groups. Each mediator group must have a unique name and can have an optional description. The identifier used for specifying a mediator is the mediator's server id, which is usually **server-<hostname>** and which can be found in the **<id>** element of the mediator entries in the Commander's **RegistrarPersistence.xml** file. You can configure a mediator to be included in multiple mediator groups.

```
<MediatorGroups xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MediatorGroups.xsd">

  <MediatorGroup name="DevPool" description="DevelomentGroup">
    <Mediator>server-VM1</Mediator>
    <Mediator>server-VM2</Mediator>
  </MediatorGroup>

  <MediatorGroup name="ProdPool1">
    <Mediator>server-VM1</Mediator>
    <Mediator>server-VM3</Mediator>
    <Mediator>server-VM4</Mediator>
  </MediatorGroup>

</MediatorGroups>
```

The MediatorGroups.xml file is checked for updates every 60 seconds and updates are made dynamically.

2. **Configure rules.** You configure the rules on the Commander, in the

/etc/MediatorAssignmentRules.xml file. Each rule comprises:

- A unique name.
- An optional priority level. Rules are matched according to their priority. If no priority is configured, rules are matched according to the order they appear in the MediatorAssignmentRules.xml file. You cannot mix rules with and without priority; either all rules must have a priority or none of them must have a priority.
- Zero or more conditions. For a rule to be matched, all the conditions must be matched. Conditions use operators for matching the metadata received from the probe. The following table describes the operators you can use and the resulting match, assuming a rule condition of **name=RULENAME operator=OP value=RULEVAL**:

Operator (OP)	Matching Result
EQ	Matches if there is an input data pair with name RULENAME and value RULEVAL.
NOT_EQ	Matches if there is an input data pair with name RULENAME and its value is not equal to RULEVAL, or if there is no input pair with name RULENAME.
LT	Matches if there is an input data pair with name RULENAME, its value is numeric and smaller than RULEVAL.
GT	Matches if there is an input data pair with name RULENAME, its value is numeric and greater than RULEVAL.
REGEXP	Matches if there is an input data pair with name RULENAME, and its value matches the regular expression RULEVAL.

- One or more mediator groups (the result of the rule being matched). When a rule is matched, one of the mediators in the resulting mediator groups is selected for assignment, based on the default auto-assignment algorithm. The mediator groups must match a mediator group defined in the MediatorGroups.xml file.

Rule matching and mediator assignment is made according to the following logic:

- Rules are matched according to their configured priority, or if no priority is configured, according to their order in the file.
- At the first successful match, matching is stopped and one of the mediators from the resulting mediator groups is assigned based on the default auto-assignment algorithm.
- The MediatorAssignmentRules.xml file contains a setting at the beginning called **enableDefaultRule**, which by default is disabled (set to **false**). This means that if no match is made, the probe is rejected and no mediator is assigned. You can enable this rule (set to **true**) so that if no match is made, the probe is not rejected and a mediator is assigned based on the default auto-assignment algorithm.
- If no rules are configured in the MediatorAssignmentRules.xml file, or if no metadata is sent by the probe, rule based auto-assignment is not used and a mediator is assigned based on the default auto-assignment algorithm.

Tip: If you want to assign mediators from a specific mediator group as a default option if no match is made, create a rule at the end of the MediatorAssignmentRules file (or with the lowest priority) that

has no conditions, but includes the required mediator group in the Results section.

The following is a sample MediatorAssignmentRules.xml file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AssignmentRules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="AssignmentRules.xsd" enableDefaultRule="false">

  <Rule name="DevRule">
    <Condition>
      <PropertyCondition name="Application" operator="EQ" value="App1"/>
      <PropertyCondition name="Env" operator="EQ" value="Dev"/>
    </Condition>
    <Results>
      <Result>DevPool</Result>
    </Results>
  </Rule>

  <Rule name="ProdRule">
    <Condition>
      <PropertyCondition name="Application" operator="REGEXP" value="Banking.*"/>
      <PropertyCondition name="Env" operator="EQ" value="Prod"/>
    </Condition>
    <Results>
      <Result>ProdPool1</Result>
      <Result>ProdPool2</Result>
    </Results>
  </Rule>

</AssignmentRules>
```

The MediatorAssignmentRules.xml file is checked for updates every 60 seconds and updates are made dynamically.

Failure Conditions & Forced Reassignment

The Commander keeps track of which probes have been previously assigned to which mediator. This means that when a probe is restarted and requests a mediator assignment, it always receives the same mediator to which it was previously assigned.

There may be instances in which a mediator may have been taken off-line, indefinitely, and it is preferable for the probe to be re-assigned to a different mediator. This is automatically handled by the following **registrar.auto_assignment_expiration** property in the **webserver.properties** file on the Commander. For example:

```
registrar.auto_assignment_expiration = 24h
```

If a probe is unable to communicate with a mediator (whether the probe or mediator is down) for the configured amount of time (in the above example, for 24 hours), the Commander forgets that probe to mediator assignment, and if the probe is restarted, the Commander reassigns it to a mediator.

Previously assigned probes on each mediator are tracked in the Registrar. They are also persisted in the RegistrarPersistence.xml file on the Commander so that they are not lost if the Commander is restarted. The following is an example of the relevant entry in the RegistrarPersistence.xml file:

```
<entry type="mediator">
  <id>server2</id>
  <lanid>Default</lanid>
  <name>server2</name>
  <host>TestHost600.mycompany.net</host>
  <port>2613</port>
  <protocol>http</protocol>
  <runs>
  </runs>
  <active>true</active>
  <probes>
  </probes>
  <assigned>
    <probe oid="TestProbe-0:35000" />
    <probe oid="TestProbe-1:35001" />
  </assigned>
</entry>
```

If necessary, you can force an auto-assignment in one of the following ways:

- In the RegistrarPersistence.xml file, under the <entry type="mediator"> section, manually remove the relevant probe entry from the <assigned> section. Restart the Commander and probe.
- Stop the probe and wait for registrar.auto_assignment_expiration. This removes the entry from the RegistrarPersistence.xml file at run-time and enables reassignment when the probe is restarted.

Tasks

This section includes:

- ["How to Force Pre-Assignment" below](#)
- ["How to Move a Probe Assignment to a Different Mediator" on the next page](#)

How to Force Pre-Assignment

As previously assigned probes are recorded in the RegistrarPersistence.xml file, you can edit the file to pre-assign probes to a given mediator even before they are initially registered in Diagnostics.

To force pre-assignment of a probe:

1. In the RegistrarPersistence.xml file, manually enter **probe oid** (probe_id:port) under the <assigned> section of a mediator.

```
<entry type="mediator">
  .
  .
  .
  <assigned>
```

```

    <probe oid="TestProbe-0:35000" />
    <probe oid="TestProbe2-0:35000" />
  </assigned>
</entry>

```

2. Restart the Commander. The Commander considers the probes you added to have been previously assigned to the relevant mediator, and when they ask for assignment, they are allocated that mediator to register with.

How to Move a Probe Assignment to a Different Mediator

1. In the RegistrarPersistence.xml file, manually move the **<assigned>** **<probe>** entry to a different mediator.
2. Restart the Commander.
3. Restart the probe so that it reregisters with the new mediator assignment.

Tips/Troubleshooting

The following can assist you in troubleshooting:

- For debug logging of assignment rules, set: `log4j.logger.full.registrar=${DEBUG}` in **logging.properties**.
- System Health and System Capacity views. For details, see "System Views" in the Diagnostics User Guide.

Note: The System Capacity view now shows the current capacity settings for each server.

- On the Commander, view the **etc/RegistrarPersistence.xml** file.

Configure Diagnostics for ServiceGuard (HA solution)

You can configure Diagnostics for Micro Focus ServiceGuard as a HA (High Availability) solution. This section outlines the necessary steps for configuring Micro Focus Diagnostics (Version 7.50 and higher) to run under ServiceGuard (Linux).

Note: It is assumed that you are familiar with both, Diagnostics and Micro Focus ServiceGuard.

The configuration steps described in this section can be used for other HA solutions as well (for example Microsoft Cluster Service).

The Diagnostics server should be installed on the shared disk with enough room for the Diagnostics time series database (TSDB) and other configuration items (for example user rights, custom dashboard screens, etc).

Both Diagnostics servers (active and standby) need to be time synchronized via NTP or BSM/APM. It is not recommended to use SYSTEM as the time synchronization mechanism since the "clock" used by the Diagnostics server needs to be the same on both servers.

The Diagnostics server uses the hostname as a prefix for sub-directories in the archive and storage directory. This needs to be overwritten on the Java command line that starts the server by specifying -**Dmediator.id=cluster -Dserver.id=<cluster>** (<cluster> can be replaced by any other unique name)

Example: `<installdir>/bin/server.sh`

```
$JAVA1_5_HOME/bin/java -Dserver.id=cluster -Dmediator.id=cluster -server
-Xmx512m
$SERVER_BCP $JAVA_OPTS -Dsun.net.client.defaultReadTimeout=70000
-Dsun.net.client.defaultConnectTimeout=30000
-classpath $SERVER_HOME/lib/mediator.jar$PATHSEP$SERVER_HOME/lib/
loading.jar$PATHSEP$SERVER_HOME/lib/common.jar$PATHSEP$SERVER_HOME/
lib/mercury_picocontainer-1.1.jar com.mercury.opal.mediator.util.DiagnosticsServer
```

Note: All command line components need to be on the same line.

The ServiceGuard package requires start and stop commands for the application. The start command for Diagnostics is the `<diag_server_install_dir>/bin/server.sh` script.

The stop command requires a new script that should reside in `<diag_server_install_dir>/bin` as well, with the following content:

`<installdir>/bin/stop.sh`

```
#!/bin/sh
PID=`ps -ef | grep -v grep | grep DiagnosticsServer | awk '{ print $2 }' `
kill $PID
sleep 10
```

Note, make sure that the script has execute permissions (`chmod u+x stop.sh`).

In the ServiceGuard package script, add the following lines:

```
stop_command:
<installdir>/bin/stop.sh

start_command:
<installdir>/bin/server.sh &
```

Note, replace `<installdir>` with the Diagnostics' server install directory and make sure that there is an ampersand (&) at the end of `server.sh`.

Diagnostics Server Assignments (LoadRunner/Performance Center Runs)

By default, a probe that is selected for a LoadRunner or Performance Center run uses the Diagnostics Server specified in its `<agent_install_dir>/etc/dynamic.properties`.

It is possible to override the configuration when the probe is started for a run. To do so, modify a mapping file on the Diagnostics Commander Server. This enables you to override the Diagnostics Server assignment for a probe.

This can be useful when you are running Diagnostics in a combined LoadRunner / Performance Center and BSM/APM environment. You could have the probes use different Diagnostic Servers when they are in a LoadRunner / Performance Center run than when they are reporting data to BSM/APM.

It might be more convenient to use this mechanism than to edit the probe configuration file.

Note: When the probe is not in a run, it uses the Diagnostics Server specified in its `<agent_install_dir>etc/dynamic.properties` file.

To override the Diagnostics Server assignment for a probe, modify the `server_assignment.properties` file in the `<diag_server_install_dir>\etc` directory on the Diagnostics Server in Commander mode host machine.

The format of the `server_assignment.properties` file is:

```
<ProbeID> = <Server.id>
```

- Replace **<ProbeID>** with the ID of the probe.
- Replace **<Server.id>** with the ID of the Diagnostics Server.

The `server_assignment.properties` file is dynamically read at the start of each LoadRunner / Performance Center run. Changes made to this file become effective without restarting the Diagnostics Server in Commander mode.

Configure the Diagnostics Server for LoadRunner Offline Analysis File Size

For each LoadRunner scenario or Performance Center test that is run, the Diagnostics Server in Mediator mode produces a file that is needed for LoadRunner offline analysis containing the Java data captured during the scenario. The size of this file can grow quite large. Make sure you have enough disk space to hold the LoadRunner Offline file on both the Diagnostics Server in Mediator mode host machine where the file is stored temporarily while the scenario is running and the Load Runner Controller host machine where the file is stored when the scenario ends.

Estimate the Size of the LoadRunner Offline File

Estimating the size of the offline file is highly dependent upon the data and rate at which the data is captured.

To estimate the size of the LoadRunner offline file:

1. Run a load test for five minutes and monitor the size of the offline file created by the Diagnostics Server in Mediator mode when the Load Runner scenario is started.
Locate the offline file on the Diagnostics Server in Mediator mode host machine in `<diag_server_install_dir>/data/<newest directory>`. The offline file has an extension of `.inuse`.
2. After five minutes, note the size of the offline file.
3. Extrapolate the size of the offline file after an hour by multiplying the size of the offline file from the previous step by 12.
4. Determine the anticipated size of the offline file at the end of the load test by multiplying the 1 hour file size calculated in the previous step by the number of hours you expect your actual load test to run.
5. Determine if the Diagnostics Server in Mediator mode host machine and the Controller host machine have enough disk space to accommodate the anticipated offline file size.

Reduce the Size of the LoadRunner Offline File

If you are concerned about the size of the offline file, you can reduce the file size by increasing the offline aggregation periods for the Diagnostics Server in Mediator mode. This will reduce the level of granularity in the offline data and the size of the offline files.

The default settings for these properties are **5s** (5 seconds), which causes the Diagnostics Server in Mediator mode to aggregate all data into 5-second time slices. Increasing the value of these properties makes the

offline file smaller because fewer data points need to be stored when the aggregation period is longer. For example, increasing the offline aggregation period properties to 45s reduces the file size by 50-75%.

Note: The impact on the size of the offline file size that will be achieved by adjusting the offline aggregation period is highly dependent upon the behavior of the application and the specifics of your load test.

Use the following steps to modify the Diagnostics Server in Mediator mode offline aggregation period properties **bucket.lr.offline.duration** and **bucket.lr.offline.sr.duration** in `<diag_server_install_dir>/etc/server.properties`.

To reduce the size of the offline files by increasing the Diagnostics Server in Mediator mode offline aggregation periods:

1. Make sure that the Diagnostics Server in Mediator mode is not participating in any active LoadRunner / Performance Center runs. This is necessary because the Diagnostics Server in Mediator mode must be restarted before the property changes described in the following steps can take effect.
2. Access the Mediator Configuration Page by navigating to the following URL:

```
http://<diagnostics_server_hostname>:8081/configuration/Aggregation?level=60
```
3. Increase the Offline VU Aggregation Period by increasing the setting for the **Load Runner / Performance Center Offline VU Aggregation Period** property. The value of this property must be a multiple of 5; for example, 45s.
4. Increase the Offline Server Request Aggregation Period by increasing the value of the **Load Runner / Performance Center Offline Server Request Aggregation Period** property. The value of this property must be a multiple of 5; for example, 45s.
5. Update the Diagnostics Server in Mediator mode with the revised property values by clicking **Submit** at the bottom of the page.

A message appears at the top of the page to indicate that the changes were saved along with a reminder to restart the Diagnostics Server in Mediator mode. The **Restart Mediator** button is also displayed.

For more information on updating property values from the Configuration Page and a screen image showing the command buttons, see ["Making Server Configuration Changes" on page 95](#).

To cause the configuration changes to take effect, restart the Diagnostics Server in Mediator mode by clicking **Restart Mediator**.

Configure Diagnostics Using the Diagnostics Server Configuration Pages

The Diagnostics Server Configuration pages enable you to set the property values that control how the Diagnostics Server communicates with the other Diagnostics components, and how it processes the data it receives from the probes.

Note: To ensure that you are entering valid property values, use these pages to update the Diagnostics Server properties rather than editing the property files directly.

For information about viewing and modifying Diagnostics using the Diagnostics Server Configuration pages, see ["Diagnostics Administration UI" on page 92](#).

Configure a Custom Context Root

To configure a custom context root on Diagnostics commander server set the following in the **etc/webserver.properties** file:

```
# Reverse proxy prefix for Diag URLs (e.g. /diag/customername in ES environment) #  
reverse_proxy.prefix=
```

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.

Configure Diagnostics for PCF

This section describes how to configure the diagnostics agent to monitor applications running in PCF (Pivotal Cloud Foundry).

Overview

Cloud Foundry is a PaaS (Platform as a Service) open-source technology that can be implemented over an IaaS (Infrastructure as a Service) cloud, such as AWS, Microsoft Azure, and so forth) or on a laptop with a virtual machine.

It is also called multi-cloud since the interaction with Cloud Foundry is independent from the backend platform. This interaction can be done using a command line calling REST services from the backend Cloud Foundry implementation.

Pivotal Cloud Foundry (PCF) is an implementation of Cloud Foundry. It allows you to launch or push the server onto the cloud, both private and public.

For more information, see cloudfoundry.org.

Obtain the Installation files

To prepare for your installation:

1. Download and save the Diagnostics PCF zip files from the [Software Support site](#).
2. [Download](#) and install the CF Command Line Interface (CLI). The command line tool is **cf**.
3. If you have an HTTP proxy server on your network between the host running the **cf** CLI and your Cloud Foundry API endpoint, you must set the https proxy with the hostname or IP address of the proxy server. For details, see <https://docs.cloudfoundry.org/cf-cli/http-proxy.html>.

Start the Diagnostics Server Using PCF (Commander Mode)

1. Extract the contents of the PCF zip. Make sure the contents include the **diagnosticsserver-dist-zip** and the **manifest.yml** files.
2. Verify that the **cf** CLI is available in the folder in which you extracted the zip file.
3. Run **cf apps**. You should see a list of all running apps.
4. Run **cf push <commander server name>**. You can enter any meaningful name that you want to associate with the Diagnostics server.

Start the Diagnostics Server Using PCF (Mediator Mode)

1. Extract the contents of the PCF zip. Make sure the contents include the **diagnosticsserver-dist-zip** and the **manifest.yml** files.
2. Edit the manifest.yml file and append the following lines to the end of the file:


```
env:
  DIAG_COMM_HOST: <commander server name>
  DIAG_COMM_PORT: 80
```
3. Verify that the **cf** CLI is available in the folder in which you extracted the zip file.
4. Run **cf apps**. You should see a list of all running apps.
5. Run **cf push <mediator server name>**. You can enter any meaningful name that you want to associate with the Diagnostics server.

How to start Diagnostics agent in PCF

To start a Diagnostics agent in PCF, you will need to push the application to be monitored using the buildpack created for Diagnostics agent. Follow the steps below in the app directory of the application to be monitored.

1. Run **cf push <app name> -b https://github.com/nayaknee/java-buildpack.git**
2. Get the url of the app from the console.

```
name:                springapp-950
requested state:     started
instances:          1/1
usage:              768M x 1 instances
routes:             springapp-950-uncrusted-planimeter.cfapps.io
last uploaded:     Tue 24 Apr 15:07:22 IDT 2018
stack:              cflinuxfs2
buildpack:          https://github.com/nayaknee/java-buildpack.git
```

3. To bind the app to the Diagnostics agent, you create a service and bind it to the app.
 - a. Run **cf create-user-provided-service diagnostics -p '{ "mediator-url" : "http://<<diagserverhostname>>:80" }'**
 Replace <<diagserverhostname>> with the name of the Diagnostics server. **Note:** The server host name begins with "diagnostics".

```
requested state: started
instances: 1/1
usage: 2G x 1 instances
urls: diag950commander.cfapps.io
last uploaded: Tue Apr 24 10:19:48 UTC 2018
stack: cflinuxfs2
buildpack: https://github.com/cloudfoundry/java-buildpack
```

In the above example, the Diagnostic mediator-url is <http://diag950commander.cfapps.io:80>.

- b. Run **cf bind-service app diagnostics**

Known Issues

The following known issues apply to the PCF integration:

- The Diagnostics Next Gen user interface is not supported.
- Diagnostics Agent Profiling is not supported.

Administration

Diagnostics Administration UI

Information is provided on how to access and use the Diagnostics Server Administration UI, where you can configure Diagnostics properties and manage your Diagnostics software.

This chapter includes:

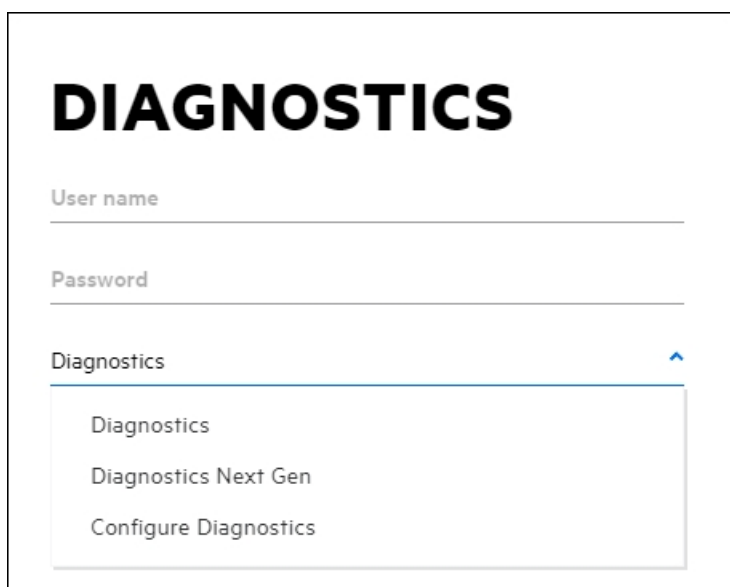
- ["Accessing the Diagnostics Administration UI" below](#)
- ["Using the Diagnostics Administration UI" on the next page](#)

Accessing the Diagnostics Administration UI

You can view information about the Diagnostics configuration, set the user privileges, configure Diagnostics settings and manage your Diagnostics software directly from the main UI of Diagnostics.

To access the Diagnostics Administration UI:

1. Open the main Diagnostics UI by navigating to `http://<diagnostics_server_host>:2006` in your browser, or by opening **Administration** from the computer's Start menu. The port number in the URL, **2006**, is the default port for the Diagnostics Server. If you configured the Diagnostics Server to use an alternative port, use that port number in the URL.
2. In the Diagnostics UI opening screen, you are prompted for a user name and password. This must be a valid user name, and must have both **View** and **Change** privileges. For information about valid user names and privileges, see ["User Authentication and Authorization" on page 101](#).



The Diagnostics opening screen contains the following options for accessing the Diagnostics UI, where you can view the performance metrics collected by the agents that are reporting to the Diagnostics Server. For more information about the Diagnostics views, see the online help or the Diagnostics User Guide.

The **Diagnostics** dropdown menu contains three options:

- **Diagnostics:** Opens the UI in the browser (if Java Applet is supported).
 - **Diagnostics Next Gen:** Opens a new user interface that is designed to provide a modern and intuitive user experience. The initial phase covers only a few important flows and metrics. It does not cover all the existing functionality of Diagnostics.
 - **Configure Diagnostics:** Opens the Components administration page.
3. Select **Configure Diagnostics** from the **Diagnostics** drop-down menu, and click **Log in**. See "[Using the Diagnostics Administration UI](#)" below for details.

Note:
 Diagnostics continues to prompt for a user name and password until valid credentials are entered.
 If you click **Cancel**, the following error message is displayed in your browser: **Access denied. You must specify a valid user name and password.**
 If you entered a valid user name and password, but do not have the proper privileges, the following error message is displayed in your browser: **Access denied. You do not have the required permission to view this screen.**

To log on as a different user to the one you are currently logged on as, you must close your browser and reopen it.

Using the Diagnostics Administration UI

In the Diagnostics Administration UI, you view information about your Diagnostics configuration, set the property values that control how the Diagnostics Server communicates with the other Diagnostics components, and how it processes the data that it receives from the probes.

To ensure that you are entering valid property values, it is recommended that you use the configuration pages to modify the Diagnostics Server properties, rather than editing the property files directly.

From the main Diagnostics UI select **Configure Diagnostics** and the **Components** page is displayed.

Diagnostics	
Components	
Component Name	Component Description
registrar	Central list of all Diagnostics component deployments
query	Query API - allows you to download diagnostics data in HTML, XML or as Java objects
security	Built-In User Management
logging	Configure log files and logging details.
configuration	Configuration
files	Installation directory browser - upload and download property files, log files, etc
license	License Management
synchronize	Synchronize uCMBD Models in BSM
thresholding	Script thresholds and alert rules
(Show Advanced Options)	
Diagnostics Server "server-myd-vm08972", version 9.30.6.238	

You can also access this Components page by selecting the **Maintenance** link in any Diagnostics view.

You can select from the following links to go to different administration pages for Diagnostics. Some of the links take you to information pages and other links allow you to make configuration changes.

- **Registrar:** Central list of all Diagnostics component deployments.
- **Query:** Query API which enables you to download Diagnostics data in HTML, XML or as Java objects. An example is provided in the /contrib directory of the use of the Diagnostics Query API to create a custom dashboard. Also see the Micro Focus Diagnostics Data Model and Query API Guide (pdf) available from the online help Home page and in the Documentation directory.

If you select the **Active Users** link at the bottom of the initial query page you can get a list of active users seen by the Diagnostics server in the last 60 seconds. And you can see the Queries/sec indicating how much load the user generates with summary or trend queries.

- **Security:** Built-In User Management. See "[Understanding the Diagnostics Server Permissions Page](#)" on [page 103](#).
- **Logging:** Configure log files and logging details. See, "[Configuring and Using Diagnostic Server Logs](#)" on [page 96](#).
- **Configuration:** Configure the Diagnostics Server. See "[Making Server Configuration Changes](#)" on the [next page](#) for more information on additional configuration pages.
- **Files:** Installation directory browser for use in uploading and downloading property files, log files and other files.

Note: By default, the upload button on this page is disabled. To enable it, in the `<INSTALL_DIR>/etc/common.properties` file, change the value of the `enable.file.uploadFromUI` parameter to `true` (`enable.file.uploadFromUI=true`).

Caution: Enabling this feature may lead to security issues.

- **License:** License management. See "[About Diagnostics Licensing](#)" on [page 18](#) for details.
- **Synchronize:** Synchronize CIs with BSM/APM. You can force a hard sync (perform full synchronization with BSM/APM) or soft sync (synchronize only new CIs with BSM/APM).
- **Thresholding:** Script statements for setting thresholds and alerts.

The components displayed on the Components page are the commonly used components. The more advanced components are hidden by default.

Note: Do not manipulate the advanced options without the guidance of your Micro Focus Software Customer Support representative.

To display the advanced options:

At the bottom of the page, click **Show Advanced Options**.

The list of options on the page is updated to include the advanced configuration options, and the link changes to **Hide Advanced Options**.

Additional advanced configuration options are displayed.

To hide the advanced options:

At the bottom of the page, click **Hide Advanced Options**.

The list of options on the page is updated so that the advanced configuration options are no longer visible, and the link changes to **Show Advanced Options**.

Making Server Configuration Changes

From the main Diagnostics UI, select **Configure Diagnostics** and then select the **configuration** link to access the Configuration page shown below.

Diagnostics	
Configuration	
Name	Description
Alert Properties	Properties that are used to configure the alert settings for the Server. All changes take effect dynamically, without server restart.
Component Communications	Properties that are used to configure how this Diagnostics Component communicates with the other diagnostic components.
Memory Diagnostics	Properties that configure the way that the Server will handle memory diagnostics
Online Cache	Properties that are used to configure the Server's Online Cache.
logging	Configure log files and logging details.
Show Advanced Options	
Diagnostics Server "server-myd-vm08972", version 9.30.6.238	

1. Click the link to the page whose properties you want to update. For the Diagnostics Server you can configure:
 - Customer information
 - Alert properties
 - Component Communications
 - Memory Diagnostics
 - Logging
 - Online cache

The configuration options displayed on this page are the commonly configured options. The more advanced configuration options are hidden by default. Select **Show Advanced Options** to see more configuration options.

Caution: Do not manipulate the advanced options without the guidance of your Micro Focus Software Customer Support representative.

2. For example if you select **Memory Diagnostics** the page below is displayed. Review the properties that are displayed and make updates.

Diagnostics			
Memory Diagnostics			
Name	Value	Description	Default Value
Worst Collection Limit	<input type="text" value="10"/>	The number of "worst" collections that should be tracked.	10
<input type="button" value="Submit"/> <input type="button" value="Reset All"/>			
Show Advanced Options			
<small>Diagnostics Server "server-myd-vm08972", version 9.30.6.238</small>			

3. When you are satisfied with your changes, click **Submit** to save them. Click **Reset All** to reset ALL values back to the default settings or close the dialog if you do not want to submit any changes.

A message appears at the top of the page to indicate that your changes were saved.

- For most properties that you update, a message is displayed reminding you to restart the Diagnostics Server. The property changes will not take effect until you restart the Diagnostics Server.

If you want make other changes to the Diagnostics Server properties, you should finish making all of your changes before restarting the Diagnostics Server.

Restarting the server will result in a small loss of data (up to 6 minutes). You should therefore schedule restarts at a time that is convenient.

- Modifying the logging level details does not require restarting the Diagnostics Server; however, it could take up to a minute for your changes to be applied.

Configuring and Using Diagnostic Server Logs

To configure and view the Diagnostic Server logs, from the main Diagnostics UI, select **logging**. This opens a page which comprises the following log options:

• View Log Files

Selecting this option displays a page with a list of all the server-side logs that are located in the **<installdir>/log** directory. Click a specific log file to display its content. The way the log content is displayed depends on the configuration options available at the top of the page. The following table describes the various log configuration options:

Configuration Option	Description	Values	Default Value	Default Parameter Name (in the <installdir>/etc/webserver.properties file)
View File Options	<p>Tail - shows the last configured number of lines of the file and the page is automatically refreshed when the log file content changes.</p> <p>Whole File - displays all of the file content and there is no automatic page refresh.</p>	Tail Whole File	Tail	tail.view.option
Refresh	The frequency the page is refreshed (in seconds) when the View File Option is set to Tail and the log file content changes.	1, 5, 10, 30	10	tail.view.refresh
Number of lines to display	The number of lines from the log file to display (and refresh) when the View File Option is set to Tail . When set to 0, the entire log file content is displayed and new content is added on each page refresh.	Any whole number	50	tail.lines
Output Filter	Sets a filter which selects which log lines are displayed. You can set a filter to display errors only, errors and warnings, or those lines that match a configured regular expression.	None Error Only Error and Warning Regular Expression	None	tail.filter (If using a filter with a regular expression, then also configure the tail.filter.regex parameter.)
Output Coloring	You can configure a regular expression that causes a line that matches the expression to be displayed in green.			tail.coloring.regex
Reverse Output	By default, log lines are displayed from old to new. Select this check box so that the log lines are displayed in reverse order, from new to old.		false	tail.output.reverse

Configuration Option	Description	Values	Default Value	Default Parameter Name (in the <installdir>/etc/webserver.properties file)
Pause Tail	Select this check box to temporarily pause the log output and automatic refresh. When you deselect the check box, the view is refreshed with the most recent output and the display continues according to the configured options.			

You can also download the entire log file by clicking the **Download File** link.

• View Log Levels

Selecting this option enables you to set the logging level for each log file. For each log file listed on the page, you can set the logging level for the entire file (that is, for the **logging** component of the file), as well as for individual components of the file. Setting a log level for an individual component causes data for that specific component to be included in the full log file (full.log) as well. The following are the log levels you can set:

- **DEBUG:** This is for troubleshooting purposes only.

Caution: Do not keep this level set for a long period of time in a production system as some of the components can generate a very large amount of logging data.

- **INFO:** This logs informational data only and can be used as required, even permanently.
- **WARNING:** This logs warnings of potential errors. While they may not indicate a real problem, they can point to potential issues and provide additional information should actual issues occur.
- **ERROR:** This logs errors that need attention.
- **IO:** Not used.

Note: You can also change the logging level by configuring the <installdir>/etc/logging.properties file directly. Changes to either the UI page or the logging.properties file can take up to a minute before they become effective.

Diagnostics Local Client

The Diagnostics Local Client enables you to launch the User Interface of a target Diagnostics server as a desktop application without the need for a web browser.

Support Matrix

- **Operating Systems:** Windows 2012R2, 2016, 7 and 10.
- **Diagnostics versions:** Diagnostics 9.40 and later.

Launch Diagnostics UI using the Diagnostics Local Client

1. Download the **diag_local_client-9.51.zip** from [Software Support](#).
2. Extract the contents from **diag_local_client-9.51.zip** and save in a folder.
3. Double-click **local_client-start.bat** to open Diagnostics Local Client.
4. Enter the Diagnostics URL in the format **http://<host name>:2006/**
5. Click on **LOAD** button to open Diagnostics login screen.
6. Enter the user credentials to open Diagnostics UI.
7. To launch Diagnostics Profiler UI, enter login credentials in the pop up wizard and click Login.

Access secure connections (HTTPS) of Diagnostics using the Diagnostics local client

1. Copy the security certificate **diag_server_commander.cer** from the Diagnostics server to a temp location.
2. Import the certificate into cacert keystore available with local client and run the following command:

```
<Local client path>\Diagnostics-Client-9.51\java\bin\keytool -import -alias
downloadedCertAlias -keystore <Local client path>\Diagnostics-Client-
9.51\java\lib\security\cacerts -file <Security certificate path>\diag_server_
commander.cer
```

The default password is **changeit**. Type **Yes** when prompted to trust the certificate.

Open local client launcher and enter Diagnostic URL **https://<hostname>:8443/**, the Diagnostics login screen opens.

Note: Next Gen UI , Maintenance and other HTML links will open in default browser

Access Diagnostics UI from APM local client

If you have Diagnostics integrated with APM, then perform the following steps to launch Diagnostics UI from APM:

1. Download APM local client from [Software Support](#)
2. Extract the Local client zip file and launch APM local client.
3. Provide APM URL and click **Load**.
4. To register Diagnostics with APM, go to **Admin > Diagnostics** and provide the Diagnostics server host name.
5. To launch the Diagnostics UI, go to **Applications > Diagnostics**.

Configure Proxy

The Diagnostics Local Client works with the default proxy settings present within **Tools > Internet options > Connections > LAN settings**.

Limitations

1. Once the Diagnostics probe is launched, the Enter option is disabled.
2. Diagnostics server running on PCF (Pivotal Cloud Foundry) cannot be loaded.
3. Localization is not enabled.

User Authentication and Authorization

Information is provided on the Diagnostics authentication and authorization process and describes how to create and maintain user security permissions.

This chapter includes:

- ["About User Authentication and Authorization" below](#)
- ["Understanding User Privileges" on the next page](#)
- ["Understanding Roles" on the next page](#)
- ["Accessing Diagnostics Using Default User Names" on page 103](#)
- ["Understanding the Diagnostics Server Permissions Page" on page 103](#)
- ["Creating, Editing and Deleting Users" on page 108](#)
- ["Assigning Privileges Across the Diagnostics Deployment" on page 109](#)
- ["Assigning Privileges for Probe Groups" on page 110](#)
- ["Tracking User Administration Activity" on page 111](#)
- ["List of Active Users" on page 112](#)
- ["Configuring Diagnostics to use JAAS " on page 112](#)

About User Authentication and Authorization

User authentication and authorization settings for all the Diagnostics components are configured in the Diagnostics commander server.

Authentication is the process of verifying a person's identity. Authorization is the process of verifying that a known person has the authority (permission or privilege) to perform a certain action. Roles are bundles of permissions assigned to a user.

You manage authentication and authorization by creating and editing user names and granting the users privileges so that users are able to perform the functions within the application for which they are responsible.

User permissions and privileges for the Profilers (.NET Diagnostics Profiler or Java Diagnostics Profiler) of the probes connected to a particular Diagnostics Server are also defined in the Diagnostics commander server. You can assign users one set of permissions for accessing Profilers in a particular probe group and a different set of permissions for accessing the Diagnostics Server.

Important:

- When you install the agent as a profiler only (not connected to any Diagnostics Server), you manage the authentication and authorization of users of the Profiler in the agent itself.
- For information about managing authentication and authorization for the Java Agent installed as a profiler only, see the Diagnostics Java Agent Guide.
- For information about managing authentication and authorization for the .NET Agent installed as a profiler only, see the Diagnostics .NET Agent Guide.

Before you can view any Diagnostics data, or make any changes to the Diagnostics configuration or user privileges, you must log on to the Diagnostics commander server using a user name that has valid security access with the appropriate privileges.

After logging on to the Diagnostics Server in a particular browser session, the user name remains in effect until the browser session ends. When you are finished using Diagnostics, close your browser to prevent others from accessing Diagnostics using your privileges.

Understanding User Privileges

The following privilege levels can be assigned to Diagnostics users:

Privilege	Description
View	The user can view Diagnostics data from the UI.
Execute	The user can make changes to the settings on the UI, such as changing thresholds or adding comments. On the Profiler, this privilege gives permission to perform garbage collection and clear the performance data held by the Profiler.
Change	The user can access the Configure Diagnostics menu to alter component configuration, and maintain user information. On the profiler, this gives permission to run potentially risky operations, such as taking a heap-dump or changing instrumentation.

Note: The privilege levels, **rhttpout** and **system** are for internal purposes only. **rhttpout** is used to grant the user access to the rhttp/out URL for doing remote management of distributed servers.

system is an internal permission generally granted only to the **mercury** special user. It is the permission that allows Diagnostics components to talk to one another (for example, the permission required for a probe to register with the Diagnostics Server). **System** permission is required to view system health.

Each privilege level stands alone. There is no inheritance of privileges from one level to the next. You must grant a user all of the privilege levels that are necessary to perform the functions that they need to perform.

For example, a user must be granted both **View** and **Execute** privileges to be able to make changes to thresholds. A user name that has been granted only **Execute** privileges would not be useful, as it would not allow the user to see the UI on which they have permission to make changes.

For information about assigning privileges to users, see ["Assigning Privileges Across the Diagnostics Deployment" on page 109](#).

When a user opens Diagnostics through an integration with APM, their permissions are determined by APM. See the APM-Diagnostics Integration Guide for details.

Understanding Roles

In addition to the user/privilege assignment, it is also possible to assign privileges to roles and assign these roles to users. This makes the management of multiple users easier: when a new user is added to Diagnostics only the user/role assignment has to be performed. This is especially helpful when a user is set up to have different privileges for accessing the Diagnostics Server and the Profiler of a particular probe group.

Consider the following example:

Two development teams (Dev1 and Dev2) that require all permissions (view, execute, change) to the Profiler on the agent system that they own and view permission on the agent system that they do not own. Both teams should have view and execute permissions for the UI.

The following roles must be created:

Role	Privileges
Enterprise (access to the UI)	[DevUI] = view,execute
Dev1 Probe Group	[Dev1All] = view,execute,change [Dev2View] = view
Dev2 Probe Group	[Dev2All] = view,execute,change [Dev1View] = view

Note that roles need to be enclosed in brackets to distinguish them from users. For example, if a new user to the Dev1 team is added to Diagnostics, it would need to be part of the following roles: [DevUI],[Dev1All],[Dev1View].

Accessing Diagnostics Using Default User Names

The following default user names are defined for Diagnostics:

Default User Names	Privileges	Description
user	View	Can only view the data from the UI.
superuser	View, Execute	Can view data, change thresholds, and create alerts and comments from the UI.
admin	View, Change, Execute, System	Can view data, change thresholds, and create alerts and comments from the UI. Can configure components and maintain user information.

You can use these default user names to access Diagnostics functionality.

The passwords for the default user names are the same as the user names. For example, for the user name `admin`, the password is `admin`.

You can modify the password or privileges for the default user names to suit your needs. You can also define new user names to control user access to Diagnostics.

Security tip: It is highly recommended that the default credentials are changed as soon as possible after installation. For details on configuring users, roles, permissions and authentication, see "[User Authentication and Authorization](#)" on page 101.

Note: There are two default users, **mercury** and **bac**, that are used for internal purposes and should never be modified. These users are for internal communication between components.

Understanding the Diagnostics Server Permissions Page

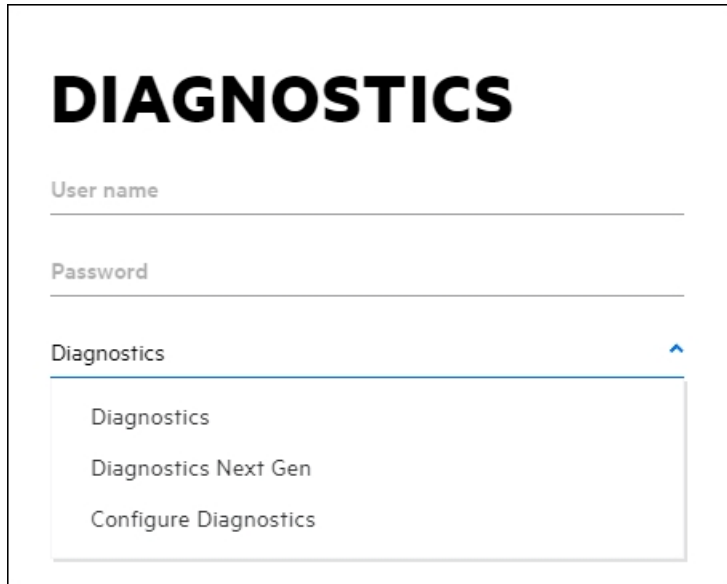
You manage users and assign user privileges in the Permissions page.

This section includes:

- ["Accessing the Permissions Page" below](#)
- ["The Permissions Page at a Glance" below](#)
- ["Enterprise and Application Permissions" on the next page](#)

Accessing the Permissions Page

You can access the Permissions page from the main Diagnostics UI by selecting **Configure Diagnostics**.



The screenshot shows the Diagnostics UI. At the top, the word "DIAGNOSTICS" is displayed in large, bold, black letters. Below this, there are two input fields: "User name" and "Password", each with a horizontal line underneath. Below the password field, there is a "Diagnostics" label with a small blue upward-pointing arrow to its right. A dropdown menu is open below this label, showing three options: "Diagnostics", "Diagnostics Next Gen", and "Configure Diagnostics". The "Configure Diagnostics" option is highlighted with a blue border.

You can also access the Permissions page by selecting the Maintenance link in any Diagnostics view and then selecting the security link.

Before you can view any Diagnostics data, or make any changes to the Diagnostics configuration or user privileges, you must log on to the Diagnostics commander server using a user name that has valid security access with the appropriate privileges.

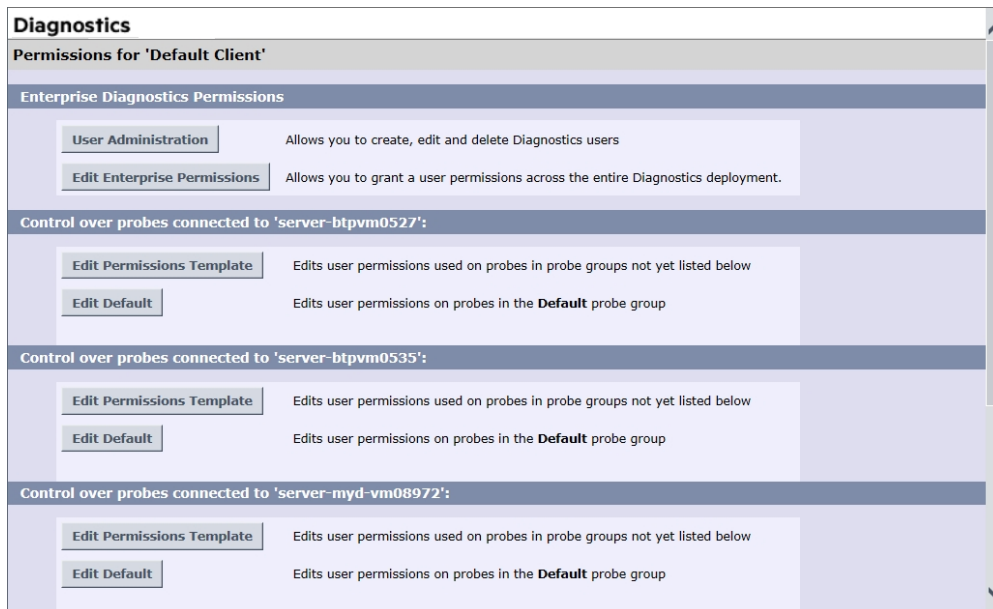
When the Permissions page opens, if you are not already signed into the Diagnostics Server, you might be prompted for a user name and password. You must have at least **View** privileges to view your privileges and modify your password. To add or delete users, or update user privileges, you must have both **View** and **Change** privileges.

Note:

- Diagnostics continues to prompt for a user name and password until valid details are entered.
- If you click **Cancel**, the following error message is displayed in your browser: **Access denied. You must specify a valid username and password.**
- If you entered a valid user name and password, but do not have the proper privileges, the following error message is displayed in your browser: **Access denied. You do not have the required permission to view this screen.**

The Permissions Page at a Glance

The following screen is an example of the Diagnostics Server Permissions page:



The Permissions page is divided into the following three sections:

- **Enterprise Diagnostics Permissions.** In this section you manage Diagnostics users and you assign privileges across the whole Diagnostics deployment, including the Diagnostics Servers and agents. By default, if users are authorized to access a particular Diagnostics Server, they also have the same authorization (and privileges) to access all probes connected to that server.

Note: Diagnostics has a centralized permissions system whereby permissions can be set for a user and they will apply to all distributed servers and probes connected to the Diagnostics system. However, permissions are only pushed out to the distributed components once every 5 minutes, so permission changes do not take effect immediately.

- **Control over probes connected to <Diagnostics_commander_server>.** In this section you assign privileges for users accessing the probe Profilers. You can assign users one set of permissions for accessing Profilers in a particular probe group and a different set of permissions for accessing the Diagnostics Commander Server.
- **Encrypt Internal Diagnostics Passwords.** You can access the EncryptPassword utility to encrypt a password.

Enterprise and Application Permissions

In addition to the enterprise and probe level permissions you set on the Permissions page, you can also set application level permissions. Application permissions are set in the Diagnostics UI in the initial Applications window. See the Diagnostics User Guide for details on setting application permissions.

The three groups of permissions are as follows:

- Enterprise
 - **View:** The user can look at performance data in the Diagnostics UI.
 - **Execute:** The user can change thresholds and add comments and create applications.
 - **Change:** The user has full administration access to the system (for example, can create users).

- Per Probe Group (applied in the Profiler)
 - **View:** The user can view performance data collected by the Profiler.
 - **Execute:** The user can run Garbage Collections and clear the performance data held by the Profiler.
 - **Change:** The user can run operations such as taking a heap-dump or changing instrumentation.
- Application
 - **View:** The user can view applications and edit entity properties (this requires Enterprise permissions set to Change).
 - **Modify:** The user can delete, rename, modify applications, and can add or remove an entity from an application.
 - **Edit Screens:** The user can edit using the Application Overview screen.

Note: Permissions are NOT inclusive (Execute does not include View).

Area and Action	Enterprise Permissions			Application Permissions		
	view	execute	change	view	modify	edit screens
Diagnostics UI						
View Diagnostics data in UI	X					
Change custom attributes		X				
Set thresholds in UI		X				
Create/Modify/Delete comments in UI		X				
Create alert rules in UI		X				
Configure Diagnostics page			X			
Configure Business Transactions		X				
View system health Note: To view system health, you also require system enterprise permission.			X			
Manage authorization and authentication for other users			X			
Access maintenance page			X			
Working with incidents	X					
Working with custom views	X					

Area and Action	Enterprise Permissions			Application Permissions		
	view	execute	change	view	modify	edit screens
Profiler UI						
Perform garbage collection in Profiler		X				
Clear performance data in Profiler		X				
View Diagnostics data in Profiler	X					
Perform heap-dump (Memory & Allocation Analysis)			X			
Change configuration			X			
User defined applications						
Create application		X				
Delete application		X			X	
Rename application		X			X	
Change application path		X			X	
Modify application permissions		X			X	
Edit custom application screen	X					X
Add entity to application	X				X	
Remove entity from application	X				X	
Edit entity properties (thresholds, comments, etc)		X		X		
View application	X			X		
Auto discovered applications, transaction applications or Entire Enterprise						
Create, Delete and Change of application path is not allowed						
Modify application permissions		X			X	
Edit application screen	X					X
Add entity to application	X				X	
Remove entity from application	X				X	
Edit entity properties		X				
View application	X					

Creating, Editing and Deleting Users

Users with both **View** and **Change** privileges can create new users, edit the password for an existing user, or delete users. Users with only **View** privileges can maintain their own password.

To create a new user:

1. Access the Diagnostics Server Permissions page as described in "[Accessing the Permissions Page](#)" on page 104.
2. On the Permissions page, click **User Administration** to open the User Administration page.
3. On the User Administration page, click **Create User**.
4. In the **New User Name** box, type a user name for the new user and click **OK**. The new user appears in the list of user names.

The name and role can contain any alphanumeric character, hyphens (-), underscores (_), periods (.), and the at symbol (@).

Note: User name and password must contain English characters only due to Browser restrictions in handling basic authentication.

5. Under **Change Password**, in the **Password** box, type a password for the new user, and confirm it by retyping it in the **Confirm Password** box.

Note: The password must be 8 characters and contain at least one upper case character (A-Z), one digit (0-9), and one special character (. - ! @ # \$ ^ & *).

6. In the **Password for <current user>** box, type the password of the user currently logged on.
7. Click **Save Changes**.


By default the new user, has **view** privileges. For information about changing the privileges assigned to the user, see "[Assigning Privileges Across the Diagnostics Deployment](#)" on the next page.

To assign roles:

1. Access the Diagnostics Server Permissions page as described in "[Accessing the Permissions Page](#)" on page 104.
2. On the User Administration page, assign the roles for a user. Make sure that the roles are enclosed in brackets ([aRole]). Roles can be separated by comma ([Role1],[Role2]).

Note: Permissions must be set up for roles under the Enterprise and/or Per Probe Group dialogs (see "[Assigning Privileges Across the Diagnostics Deployment](#)" on the next page and "[Assigning Privileges for Probe Groups](#)" on page 110).

To delete a user:

1. Access the Diagnostics Server Permissions page as described in "[Accessing the Permissions Page](#)" on page 104.
2. On the Permissions page, click **User Administration** to open the User Administration page.
3. On the User Administration page, in the **Password for <current user>** box, type the password of the user currently logged on.
4. Click the **Delete user** button  corresponding to the user you want to delete.
5. A message box opens asking if you want to delete the selected user.
Click **OK** to delete the user.

To change a user's password if you have View and Change privileges:

1. Access the Diagnostics Server Permissions page as described in ["Accessing the Permissions Page" on page 104](#).
2. On the Permissions page, click **User Administration** to open the User Administration page.
3. On the User Administration page, in the row representing the relevant user, type the new password in the **Password** and **Confirm Password** boxes.
4. In the **Password for <current user>** box, type the password of the user currently logged on.
5. Click **Save Changes** to save all the changes you made to the different user names.

To change your own password if you only have View privileges:

1. Access the Diagnostics Server Permissions page as described in ["Accessing the Permissions Page" on page 104](#).
2. On the Permissions page, click **User Administration** to open the User Administration page.
3. On the User Administration page, type the new password in the **Password** and **Confirm Password** boxes.
4. In the **Old Password** box, type your old password.
5. Click **Save Changes**.

Assigning Privileges Across the Diagnostics Deployment

Users with both **View** and **Change** privileges can grant users privileges across the entire Diagnostics deployment.

Note: For a description of the user privileges that you can assign to Diagnostics users, see ["Understanding User Privileges" on page 102](#).

To assign user privileges across the entire Enterprise:

1. Access the Diagnostics Server Permissions page as described in ["Accessing the Permissions Page" on page 104](#).
2. On the Permissions page, click **Edit Enterprise Permissions** to open the Editing Enterprise Permissions page.

The Editing Enterprise Permissions page is an editable page which enables you to modify user privileges.

3. Locate the name of the user, whose privileges you want to modify.
You add users on the User Administration page, as described in ["Creating, Editing and Deleting Users" on the previous page](#).
4. Add the privileges to the username as comma separated values.

For example, if you defined a user by the name of **newuser** and you want to assign this user with view and execute privileges you must locate **newuser** and edit the line so that it appears as follows:

```
newuser = view,execute
```

The Editing Enterprise Permissions page also includes a set of default users. These users are described in ["Accessing Diagnostics Using Default User Names" on page 103](#). You can modify the privileges of these default users.

Assigning Privileges for Probe Groups

Users with both **View** and **Change** privileges can grant users privileges for accessing the probe Profilers belonging to particular probe groups.

By default, if users are authorized to access a particular Diagnostics Server, they also have the same authorization (and privileges) to access all probe Profilers connected to that server.

However, you can assign users a different set of permissions for different probe groups than what they have for the Diagnostics Servers themselves.

Note: For a description of the user privileges that you can assign to Diagnostics users, see ["Understanding User Privileges" on page 102](#).

You can modify user privileges for each probe group individually and you can also modify a Permissions template that defines the user privilege settings for all future probe groups added to your system.

Note: User and permission settings could take up to 1 minute after the changes are saved to take effect.

For each probe group, there are three default user groups of users with certain privileges. You can choose to comment out these groups or to modify their privileges. The following groups of users are defined by default in all the probe groups:

User Group	Permissions
any_diagnostics_admin	This group refers to any user with administration (change) privileges on the Diagnostics Server. By default, any user who falls into this category and does not have any other predefined permission settings has administration permissions for all probes connected to that server.
any_diagnostics_superuser	This group refers to any user with superuser (execute) privileges on the Diagnostics Server. By default, any user who falls into this category and does not have any other predefined permission settings has execute permissions for all probes connected to that server.
any_diagnostics_user	This group refers to any user with user (view) privileges on the Diagnostics Server. By default, any user who falls into this category and does not have any other predefined permission settings has view permissions for all probes connected to that server.

To assign user privileges for accessing a particular probe group:

1. Access the Diagnostics Server Permissions page as described in ["Accessing the Permissions Page" on page 104](#).
2. In the **Control over probes connected to <Diagnostics_commander_server>** section of the permissions page, click **Edit <name of probe group>**.
The Editing Permissions page opens. This is an editable page which enables you to modify user privileges.
3. Enter the username to which you want to assign unique privileges and add the privileges to the username as comma separated values.

For example, if you defined a user by the name of **newuser** and you want to assign this user with view and execute privileges on this particular probe group, enter the following line:

```
newuser = view,execute
```

To assign user privileges using the Permissions template:

1. Access the Diagnostics Server Permissions page as described in ["Accessing the Permissions Page" on page 104](#).
2. In the **Control over probes connected to <Diagnostics_commander_server>** section of the permissions page, click **Edit Permissions Template**.

The Editing Template Permissions page opens. This is an editable page which enables you to modify user privileges.

3. Enter the username to which you want to assign unique privileges and add the privileges to the username as comma separated values.

For example, if you defined a user by the name of **newuser** and you want to assign this user with view and execute privileges on this particular probe group, enter the following line:

```
newuser = view,execute
```

You could also modify or comment out one of the user group settings defined in the template.

All future probe groups that are connected to your Diagnostics Server will inherit the user privilege settings from this Permissions template.

Tracking User Administration Activity

Each time a user enters the Diagnostics Server User Administration page, all activity that takes place is logged in the following log file: **<diag_server_install_dir>\log\useradmin.log**.

The data logged in the file includes the date and time of each action performed, a description of the action, and the name of the user performing the action.

To view the log file:

1. Open the Diagnostics Server administration page in one of the following ways:
 - By executing **Administration** on the computer's Start menu.
 - By navigating to `http://<diagnostics_server_host>:2006` in your browser. The port number in the URL, **2006**, is the default port for the Diagnostics Server. If you configured the Diagnostics Server to use an alternative port, use that port number in the URL.

The Diagnostics UI main page opens.

2. Click **Configure Diagnostics**.
3. If you are not already signed into the Diagnostics Server, you are prompted for a user name and password. This must be a valid user name, and must have both **View** and **Change** privileges. For information about valid user names and privileges, see ["Understanding User Privileges" on page 102](#).

Note:

- Diagnostics continues to prompt for a user name and password until valid credentials are entered.

- If you click **Cancel**, the following error message is displayed in your browser: **Access denied. You must specify a valid user name and password.**
- If you entered a valid user name and password, but do not have the proper privileges, the following error message is displayed in your browser: **Access denied. You do not have the required permission to view this screen.**

The Diagnostics Server Components page opens.

4. Click **logging**. The logging page opens.
5. Click **View Log Files**. A list of log files appears.
6. Click the `<diag_server_install_dir>\log\useradmin.log` link.

The log file is displayed at the bottom the page. For additional information, see ["Configuring and Using Diagnostic Server Logs" on page 96](#).

List of Active Users

You can get a list of active users seen by the Diagnostics server in the last 60 seconds. And you can see the Queries/sec indicating how much load the user generates with summary or trend queries.

From the main Diagnostics UI select **Configure Diagnostics** and the Components page is displayed. (You can also access this Components page by selecting the Maintenance link in any Diagnostics view).

Select the **query** link and then select the **Active Users** link at the bottom of that page to display a list of active users.

Diagnostics				
Active users in the last 60 seconds				
IP-Address	User Name	Last	First	Queries/sec
127.0.0.1	mercury	Thu Dec 01 17:12:49 IST 2016	Wed Nov 30 09:15:00 IST 2016	0.03
16.59.19.58	admin	Thu Dec 01 17:12:42 IST 2016	Thu Dec 01 17:12:42 IST 2016	?
Diagnostics Server "server-myd-vm08972", version 9.30.6.238				

You can configure limits on the queries the UI is executing if you find there are very high query loads. Use the **ui.properties** file on the server to set properties to throttle update frequency of UI queries to the server.

Configuring Diagnostics to use JAAS

Diagnostics can be configured to use JAAS (Java Authentication and Authorization Service) for authentication of users. If JAAS is enabled, the user name and password entered in the login dialog when the UI is accessed is authenticated by a configured JAAS pluggable authentication module (LoginModule).

Note: JAAS support is only available on the Diagnostics commander server.

JAAS must be enabled in the `<INSTALL_DIR>/etc/server.properties` file by un-commenting the following two lines:

```
authentication.jaas.config.file=jaas.configuration
authentication.jaas.realm=Diagnostics
```

The **authentication.jaas.config.file** property specifies the configuration file (relative to the etc directory) that defines the LoginModules and **authentication.jaas.realm** specifies the entry that should be used in the configuration file.

Example jaas.configuration:

```
Diagnostics
{
com.mercury.diagnostics.server.jaas.spi.SiteMinderLoginModule sufficient
ip="1.2.3.4";

com.mercury.diagnostics.server.jaas.spi.LDAPLoginModule sufficient
  useSSL="true"
  serverCertificate="etc/ldap.keystore"
  providerURL="ldap://ldap.yourdomain.com:636"
  baseDN="ou=People,o=yourdomain.com";
};
```

For more information on the JAAS configuration file, see Oracle's documentation on JAAS and Oracle's javadoc on `javax.security.auth.login.Configuration`.

Note: The users that were created by Diagnostics through the Manage Authorization and Authentication web page are used *first* when authenticating a username and password. Only if that authentication fails will the JAAS authentication be performed.

Diagnostics provides the following LoginModules:

- **LDAP:** (`com.mercury.diagnostics.server.jaas.spi.LDAPLoginModule`) which allows authentication against an LDAP server.
- **SiteMinder:** (`com.mercury.diagnostics.server.jaas.spi.SiteMinderLoginModule`) which allows authentication against a SiteMinder environment.

Note:

- After making any changes to `server.properties` and/or the `jaas.configuration` file, you must restart the Diagnostics commander server.
- When using a JAAS authentication provider that is also used in other applications, such as LDAP, it is recommended to turn on HTTPS for accessing the Diagnostics UI.
- When using a JAAS authentication provider, user accounts are maintained by the authenticating source. Ask your administrator for details on user name syntax.
- Subsequent authorization of authenticated users privileges is maintained using the permissions page. Roles can also be used if the appropriate LoginModule is configured to use them. In this case, existing roles can be used or new roles can be created.

Configuring LDAP Authentication

To configure LDAP authentication in Diagnostics you must:

1. Configure Diagnostics to use JAAS (see "[Configuring Diagnostics to use JAAS](#)" on page 112).
2. Configure the **LDAPLoginModule** on the Diagnostics commander server (see "[Configuring the LDAPLoginModule](#)" below).
3. Define the group and attribute privileges (see "[LDAP Permission Handling](#)" on page 119).

Configuring the LDAPLoginModule

Edit the Diagnostics JAAS realm (application) block in `<INSTALL_DIR>/etc/ jaas.configuration` with option values specific to your LDAP server.

The LDAPLoginModule may be used in a simplified or an advanced mode.

- In both modes:
 - SSL and a server certificate may be configured.
 - Roles may be configured.
 - Debug information may be requested.
- In simplified mode:
 - Anonymous directory searches may be done.
 - A predefined search filter is used.
 - Only a single base DN (distinguished name) may be configured.
 - Referrals are not available.
- In advanced mode:
 - Credentials must be provided for directory searches.
 - RFC 2254 compliant search filters may be used.
 - Multiple base DN's may be configured.
 - Referrals are available.

You can launch the **ldp.exe** utility on your Active Directory system to test settings.

The following table lists **LDAPLoginModule common attributes** (for all modes):

Attribute	Description and Examples	Values
authType	Specifies the security level to use when authenticating the user.	"simple" (default) "none" "strong"
debug	Specifies whether to write debug information to server.log.	"false" (default) "true"

Attribute	Description and Examples	Values
defaultRoles	Comma-delimited list of roles to assign each authenticated user. Example: "SuperUsers"	
ldapReadTimeOut	Specifies the timeout value in seconds for LDAP read operations. Example: ldapReadTimeOut="15"	
ldapConnectTimeOut	Specifies the timeout value in seconds for connect to LDAP. Example: ldapConnectTimeOut="15"	
roleAttributes	Comma-delimited list of the user's DN attributes whose values will be used as the user's roles. roles (Note: the value of a designated DN role attribute can itself contain one or more Diagnostics role or user names separated by commas.) If defaultRoles is also set, the resulting roles will be the union of the defaultRoles and roleAttributes . Example: "employeeType,JobFunction"	"roles" (default)
serverCertificate	Path to the trust store file containing the LDAP server's certificate. Path can be absolute or relative to the server's installation directory. See "Enable HTTPS Between Components" on page 124 for information on generating a keystore. Example: "etc/jssecacerts"	
useSSL	If set to true , use SSL to connect to the LDAP server.	"false" (default) "true"

The following table lists **LDAPLoginModule simple mode attributes**:

Attribute	Description and Examples	Values
allowAnonymous	If set to true , then allow anonymous searches of the LDAP server to retrieve the user's principal DN. To be effective the searchFirst attribute must also be set to true .	"false" (default) "true"
baseDN	Used to construct the principal's DN. If anonymous searches are allowed, then it is also used to specify which base DN to search for the user in. (required) Example: "OU=Users,DC=your,DC=ldap,DC=domain,DC=com"	
providerURL	URL to the LDAP server. Used for authentication. If anonymous searches are allowed then it is also used to search for the user. (required) Example: "ldap://your.ldap.domain.com:389" SSL example: "ldaps://yourldap.domain.com:636"	

Attribute	Description and Examples	Values
searchFirst	If set to true and allowAnonymous is also true then do an anonymous search for the users; otherwise, construct the user's principal DN from the uidAttribute , the user's login name and the baseDN attribute.	"false" (default) "true"
uidAttribute	Used in the construction of the user's principal DN. If anonymous searches are allowed, it is also used to construct the search filter.	"uid" (default) common values: "uid", "CN"

Example of constructing the user's principal DN:

If uidAttribute="UID", and user login name is jsmith, and baseDN="OU=Users,DC=your,DC=ldap,DC=domain,DC=com", then the user's principal DN will be:

```
"UID=jsmith,OU=Users,DC=your,DC=ldap,DC=domain,DC=com"
```

The following table lists **LDAPLoginModule advanced mode attributes**:

Attribute	Description and Examples	Values
providerURL	URL to the LDAP server used for authentication. Used to authenticate the user. Example: "ldap://yourldap.domain.com:389" SSL example: "ldaps://your.ldap.domain.com:636"	Default is the value of the searchProviderURL attribute
searchBaseDNs	Semicolon-separated list of base DN's to which to apply the search filter. (required) Example: "DN=America,DN=ns,DN=root,DN=com; DN=asia,DN=ns,DN=root,DN=com; DN=europe,DN=ns,DN=root,DN=com" Referral Example: "DN=ns,DN=root,DN=com"	
searchDN	The principal's DN used to search for the user principal to authenticate. (required) Assumes that searchFirst is set to true even if you don't specify this. Example: "CN=SearchAdmin,OU=Administrators,DC=americas,DC=ns,DC=root,DC=com"	

Attribute	Description and Examples	Values
searchFilter	An RFC 2254 compliant search filter (see http://www.ietf.org/rfc/rfc2254.txt). The "{USERNAME}" string in the filter will be replaced with the user's login name before the directory is searched. When connecting to Active Directory, it is useful to test search filters using ldp.exe before putting them in the jaas.configuration file. (required) Example1: "(uid={USERNAME})" Example2: "(&(CN={USERNAME})(objectClass=user))" Example 3: "(sAMAccountName={USERNAME})"	
searchFirst	Is set to true .	"true"
searchPassword	The password of the searchDN attribute. It may be plain text or obfuscated. (required) See "Enable HTTPS Between Components" on page 124 for information on password obfuscation. Example: "Secret123" Obfuscated example: "OBF:1fof1j1u1igh1ym51t331ym91ldp1iz01fmn"	
searchProviderURL	URL to the LDAP server used to search for the user's principal DN. This is used to find the user. Example: "ldap://america.ns.root.com:389" SSL example: "ldaps://america.ns.root.com:636" Referral example: "ldaps://ns-root.com:636"	Default is the value of the providerURL attribute.
searchReferral	If set to follow , then the LDAP sever will refer search requests to other LDAP servers. if it cannot resolve the search or authentication request. If set to follow then only the forest's principal DN needs to be listed in the searchBaseDNs . See http://download.oracle.com/javase/1.5.0/docs/guide/jndi/jndi-ldap-gl.html#referral .	"ignore" (default) "follow" "throw"

Note: After making any changes to **server.properties** or the **jaas.configuration** file, you must restart the Diagnostics commander server.

The following example is a configuration where all users have the same base DN that starts with "CN", so their principal DBs can be determined without searching for them.

```

Diagnostics {
  com.mercury.diagnostics.server.jaas.spi.LDAPLoginModule sufficient
  baseDN="OU=Users,DC=simple,DC=domain,DC=com"
  providerURL="ldap://simple.domain.com:389"
  uidAttribute="CN"
  ;
};

```

If "larry" logs in, then his principal DN will be "CN=larry,OU=Users,DC=simple,DC=domain,DC=com".

The following example is a configuration where all users have the same base DN that starts with "CN", so the principle DNs can be determined without searching for them, but you want to search for them anyway.

```

Diagnostics {
  com.mercury.diagnostics.server.jaas.spi.LDAPLoginModule sufficient
  allowAnonymous="true"
  baseDN="OU=Users,DC=simple,DC=domain,DC=com"
  providerURL="ldap://simple.domain.com:389"
  searchFirst="true"
  uidAttribute="CN"
  ;
};

```

If "sally" logs in, then her principal DN will be "CN=sally,OU=Users,DC=simple,DC=domain,DC=com".

The following example is a configuration where users may be from anyplace in the world, but we are only interested in IT employees from three regions.

```

Diagnostics {
  com.mercury.diagnostics.server.jaas.spi.LDAPLoginModule sufficient
  searchFirst="true"
  searchReferral="follow"
  useSSL="true"
  serverCertificate="etc/key.store"
  searchProviderURL="ldaps://america.ns.root.com:636"
  searchDN="CN=Searcher,OU=Admins,DC=america,DC=ns,DC=root,DC=com"
  searchPassword="OBF:1fof1j1u1igh1ym51t331ym91idp1iz01fmn"
  searchFilter="(&(CN={USERNAME})(objectClass=IT))"

  searchBaseDNs="DC=america,DC=ns,DC=root,DC=com;DC=africa,DC=ns,DC=root,DC=com;DC=russia,DC=ns,DC=root,DC=com"
  ;
};

```

If "ororro" in Africa logs in, then her principal DN will be "CN=ororro,OU=IT,DC=africa,DC=ns,DC=root,DC=com".

The following example is a configuration where users may be from anyplace in the world and hosted on different LDAP servers. In addition, users CNs may be localized but their sAMAccountNames are guaranteed to be ISO 8859-1.

```

Diagnostics {
  com.mercury.diagnostics.server.jaas.spi.LDAPLoginModule sufficient
  searchFirst="true"
  searchReferral="follow"
  useSSL="true"
  serverCertificate="etc/key.store"
  searchProviderURL="ldaps://ns.root.com:636"
  searchDN="CN=Searcher,OU=Admins,DC=america,DC=ns,DC=root,DC=com"
  searchPassword="OBF:1fof1j1u1igh1ym51t331ym91idp1iz01fmn"

```

```

searchFilter="(sAMAccountName={USERNAME})"
searchBaseDNs="DC=ns,DC=root,DC=com"
;
};

```

If Σαλοθ in Greece logs in as his sAMAccountName "Saloth", then his principal DN used for authentication will be "CN=Σαλοθ,OU=Υσερζ,DC=greece,DC=ns,DC=root,DC=com".

LDAP Permission Handling

You define roleAttributes in the JAAS file and then configure group and user privileges on the Diagnostics commander server by selecting **Advanced Options > Security**.

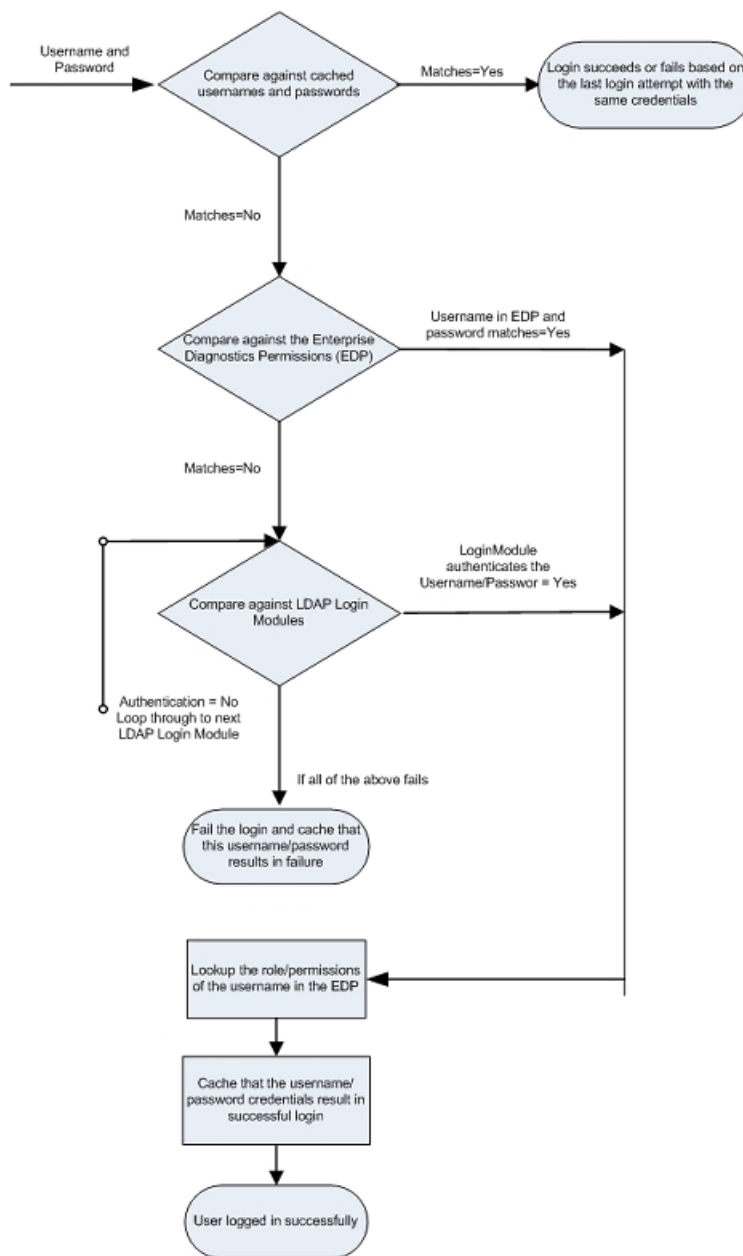
For example, if roleAttributes = "CN,Position" is defined in the JAAS file and you configure [Engineer] =view in the Enterprise Permissions file, this enables anybody with a position of Engineer to view Diagnostics.

For details on Roles and Privileges see "[Understanding Roles](#)" on page 102.

In general here is how Diagnostics handles permissions and how LDAP authentication is handled (also see the flow diagram that follows):

1. Accept a username and password from the user.
2. Compare the username and password against cached usernames and passwords.
 - a. If the username is in the cache and the password matches, then the login succeeds or fails based on the last login attempt with the same credentials.
 - b. Otherwise, proceed to the next step.
3. Compare the username and password against the Enterprise Diagnostics Permissions (EDP).
 - a. If the username is in the EDP and the password matches, then go to step 6.
 - b. Otherwise, proceed to the next step.
4. Loop through all the LDAP Login Modules configured in jaas.configuration.
 - a. If the LoginModule authenticates the username/password, then go to step 6.
 - b. Otherwise, proceed to the next Login Module.
5. If all the above fail, then fail the login and cache that this username/password results in failure.
6. Lookup the role/permissions of the username in the EDP.
7. Cache that the username/password credentials result in a successful login.
8. Return that the user logged in successfully.

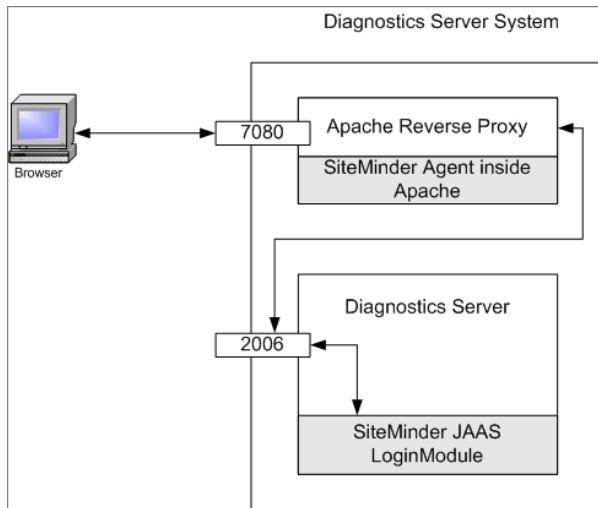
Diagnostics LDAP Authentication Flow



Using Reverse Proxy with SiteMinder JAAS LoginModule

The SiteMinder JAAS LoginModule requires a reverse proxy server in which the SiteMinder web agent is installed. A proxy server simply forwards HTTP/S requests to the Diagnostics server.

Example set-up:



In the above diagram, an Apache web server listening for requests on port 7080 is configured as a reverse proxy. It contains the SiteMinder web agent which performs the authentication and if successful, allows Apache to pass the request through to port 2006 (or whatever Diagnostics Server port is configured) on which the Diagnostics server listens.

Note: It is recommended that the login page is served up by another web server (different web server than the reverse proxy) to avoid conflicts with the redirect that the reverse proxy performs and the redirect that the SiteMinder module performs.

Alternatively, with Apache 2.2, the ProxyPass directive can be used to suppress proxying for certain URLs; for example, "ProxyPass /loginpage !". See the Apache 2.2 documentation for more information.

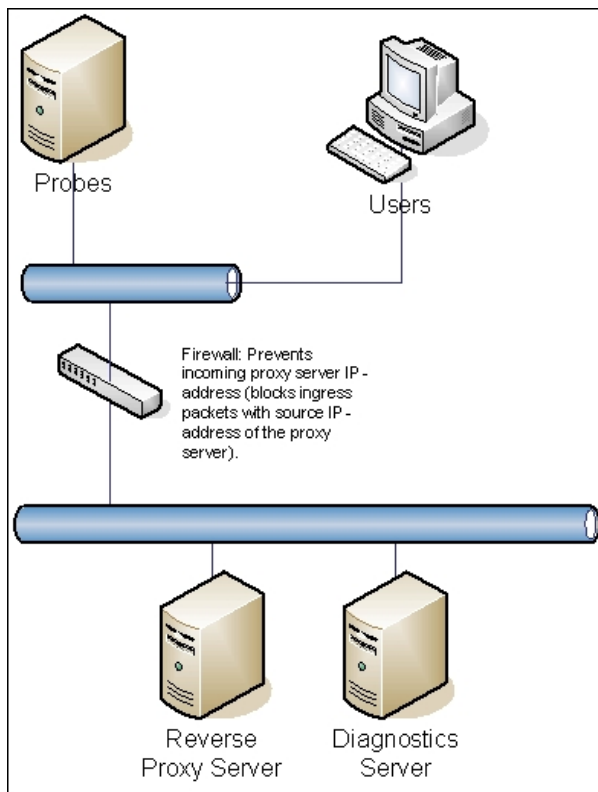
The Diagnostics Server detects requests from SiteMinder via the SiteMinder LoginModule.

Note: To use the SiteMinder JAAS authentication the users must go to the port of the reverse proxy, 7080 in this example, instead of port 2006. If the proxy server is not installed on the same system as the Diagnostics Server, the computer name for the proxy server must be used in the URL instead of the computer name of the Diagnostics server or localhost.

You can do the following as an optional step to provide additional security when you are concerned about spoofing:

- If the proxy server is not installed on the same system as the Diagnostics server, you can place the Diagnostics Server and the proxy server on the same subnet and configure an ingress filter for the proxy IP-address on the switch/router to prevent spoofing of the reverse proxy's IP-address from outside of the subnet.

See the diagram below:



Configuring SiteMinder JAAS Authentication

To configure SiteMinder authentication in Diagnostics you must configure the following on the Diagnostics commander server:

1. Configure Diagnostics to use JAAS (see ["Configuring Diagnostics to use JAAS "](#) on page 112).
2. Edit the `<INSTALL_DIR>/etc/webserver.properties` file.
 - a. Uncomment the `authentication.header.filter.username` property. Set the `authentication.header.filter.username` property to a field in the HTTP request header that should be used to get the username. By default this is set to `SM_UNIVERSALID` which is a field that SiteMinder creates in the HTTP request containing a user ID.
 - b. To use the Diagnostics roles, uncomment the `authentication.header.filter.roles` property (this is an optional step). Set the `authentication.header.filter.roles` property to a field in the HTTP header that should be used to get role information. This field can contain one role or many roles with commas separating them. If `defaultRoles` is also set, the resulting roles will be the union of `defaultRoles` and these roles.
3. Edit the Diagnostics JAAS realm (application) block in the `<INSTALL_DIR>/etc/jaas.configuration` file; for example:

```

Diagnostics
{
  com.mercury.diagnostics.server.jaas.spi.SiteMinderLoginModule sufficient
  defaultRoles="Role1,Role2"
  ip="16.228.25.40"
}

```

```

;
};

```

SiteMinder LoginModule Options

The following is a complete list of the options that can be specified for the SiteMinder LoginModule in the JAAS configuration file:

Option Name	Description	Required/Optional	Default Value	Example
IP	IP address of the reverse proxy server	required		ip="16.228.25.40"
defaultRoles	Comma-delimited list of roles to assign each authenticated user.	optional		defaultRoles="SuperUsers"

Note: After making any changes to server.properties, webserver.properties or the jaas.configuration file, you must restart the Diagnostics commander server.

Enable HTTPS Between Components

Information is provided on the configuration steps to enable HTTPS communications between the Diagnostics components and with BSM/APM.

This chapter includes:

- ["About Configuring HTTPS Communications" below](#)
- ["Filter Encryption Cipher Suites" below](#)
- ["HTTPS Checklist per Diagnostics Component" on the next page](#)
- ["Enable Incoming HTTPS Communication for Diagnostics Components" on page 126](#)
- ["Generate Client Certificate" on page 126](#)
- ["Enable Outgoing HTTPS Communication from Diagnostics Components" on page 132](#)
- ["Enabling HTTPS Communication for APM Integrations " on page 136](#)
- ["Configure Diagnostics Commander to Connect to a BSM/APM Server That Requires a Client Certificate" on page 137](#)

Note: The configuration instructions are intended for experienced users with in-depth knowledge of Diagnostics. Use caution when modifying any configuration settings for the Diagnostics components.

About Configuring HTTPS Communications

The instructions for configuring each type of component contain details of the following main steps:

- Generate a keystore on the component
- Export the certificate from the keystore
- Obfuscate passwords that provide access to the keystore
- Copy the component's certificate to the Diagnostics components that will initiate communication
- Configure the component's security properties to enable SSL and provide the passwords necessary for HTTPS communication to take place

Note: As you review this information it will be useful to reference the diagrams in ["Configure for HTTP Proxy and Firewalls" on page 52](#).

Filter Encryption Cipher Suites

A cipher suite defines the security algorithms and key sizes used for HTTPS/SSL encryption. The supported cipher suites are based on the version of Java used. Your organization may have a policy of filtering out certain ciphers either because they are not secure enough or are not allowed. There are two properties related to cipher suites that you can set in the **webserver.properties** file:

- **log.cipher.suites:** Prints messages related to cipher suites to the **server.log** file. These messages include the supported cipher suites and the cipher suites enabled after applying the **cipher.suites.filters**.
- **cipher.suites.filters:** A common-separated-value list of regular expression excludes and includes used to filter the cipher suites.

For example, suppose the following cipher suites are listed for your installation:


```

SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DH_anon_WITH_RC4_128_MD5

```

But you want to filter out the 40-bit, anonymous and DES-based cipher suites and you only want to include RC4-based encryptions. Then you can specify a filter that looks like this:\

```

cipher.suite.filters=\
  exclude:.*[0-9]?40[0-9]?.*, \
  exclude:.*_anon_.*, \
  exclude:.*_DES_.*, \
  exclude:.*_3DES_.*, \
  INCLUDE:.*_WITH_RC4_.*, \
  exclude:.*

```

HTTPS Checklist per Diagnostics Component

The following table summarizes the configuration steps that you must perform to enable HTTPS communications for each Diagnostics component:

Configuration step	Commander Server	Mediator Server	Java Probe	Collector	.NET Probe
Generate key and export certificate	Yes	Yes	Yes	Yes	No
Obfuscate passwords	Yes	Yes	Yes	Yes	No
Copy commander server certificate	No	Yes	No	No	No
Copy mediator server certificate	Yes	No	Yes	Yes	Yes
Copy Java Probe certificates	Yes	Yes	No	No	No
Copy Collector certificates	No	Yes	No	No	No
Edit security.properties: enablessl=true, keystorepassword, keystorepassword	Yes	Yes	Yes	Yes	No
Edit security.properties: add commander server certificate to trusted.certificates	No	Yes	No	No	No
Edit security.properties: add mediator server certificate to trusted.certificates	Yes	No	Yes	Yes	No

Configuration step	Commander Server	Mediator Server	Java Probe	Collector	.NET Probe
Edit security.properties: add Java probe certificate to trusted.certificates	Yes	Yes	No	No	No
Edit security.properties: add Collector certificate to trusted.certificates	No	Yes	No	No	No
Import mediator server certificate to Trusted Root Authority	No	No	No	No	Yes
Edit server.properties: set commander.url	Yes	Yes	No	No	No
Edit dispatcher.properties: set registrar.url	No	No	Yes	No	No
Edit collector.properties: set registrar.url	No	No	No	Yes	No
Edit probe_configuration.xml: set diagnosticsserver url, mediator host, metricport, and ssl	No	No	No	No	Yes
Edit metric.config: set metrics.server.uri	No	No	No	No	Yes

Enable Incoming HTTPS Communication for Diagnostics Components

This section includes instructions for configuring the Diagnostics Server, the Java Agent and the Collector to receive incoming HTTPS communications. The HTTPS communications can come from other Diagnostics components, from when the Diagnostics component is accessed using a Web browser, or when the component is accessed by other external applications.

This section includes the following topics:

- ["Configure the Diagnostics Server for Incoming HTTPS Connections" below](#)
- ["Configure the Java Agent for Incoming HTTPS Connections" on page 129](#)
- ["Configure the Collector for Incoming HTTPS Connections" on page 131](#)

Generate Client Certificate

Generate client certificate using advanced settings in Certificate Services and specify FQDN. Mark keys exportable and use the Friendly Name=CLIENT.

Configure the Diagnostics Server for Incoming HTTPS Connections

Note:

- By default, you cannot use SSL for incoming HTTPS connections due to a POODLE vulnerability problem. Instead, use TLS. For details of the POODLE vulnerability problem, see <https://www.us-cert.gov/ncas/alerts/TA14-290A>.

If you are unable to use TLS, you can either use a non-secure channel (HTTP) or enable SSLv3 for incoming HTTPS connections. To enable SSLv3, edit the **<diag_server_install_dir>\Server\etc\webserver.properties** file and remove **SSLv3** from the list of values in the **secured.connection.disabled.protocols** setting.

- If the Diagnostics Server communicates with a system running an old version of Java, you may encounter problems with HTTPS communication caused by the Java implementation. For details of this problem, see <http://www.oracle.com/technetwork/java/javase/documentation/tlsreadme2-176330.html>. We recommend that you use the latest version of Java 1.6 or later.
- To avoid issues with DNS and host name resolution, the Commander URL for the Diagnostics commander server should be configured as **localhost**. This can be accomplished by setting the `commander.url` property in **<diag_server_install_dir>/etc/server.properties** to `http://localhost:2006` (or the appropriate port number).
- When you enable HTTPS communications with a Diagnostics commander server, you must use port 8443 to run the Enterprise UI, for example: `https://<commander_server>:8443`.

To configure the Diagnostics Server for incoming HTTPS connections:

1. Generate a keystore in the **<diag_server_install_dir>/etc** directory. An example command is shown below:

```
<diag_server_install_dir>/JRE/bin/keytool -genkey -keystore <diag_server_install_dir>/etc/keystore -storepass <password> -alias SERVER -keyalg RSA -keypass <password> -dname "CN=<diagnostics_server_hostname>, OU=Diagnostics, O=Hewlett-Packard, L=Palo Alto, S=CA, C=USA" -validity 3650
```

To use this command example:

- Replace **<diag_server_install_dir>** with the path to the installation directory for the Diagnostics Server.
- Replace **<diagnostics_server_hostname>** with the machine name for the host of the Diagnostics Server (you should use the fully qualified domain name for the subject (CN) in the certificate).
- Replace each occurrence of **<password>** with the same password string. (You can assign different passwords to the **storepass** and the **keypass**, but we do not recommend doing this.)

After you execute this command, a keystore is created in **<diag_server_install_dir>/etc/keystore** with an entry called **SERVER** for the host of the Diagnostics Server.

2. Export the certificate for the SERVER entry in the keystore using the following command.

```
<diag_server_install_dir>/JRE/bin/keytool -export -keystore <diag_server_install_dir>/etc/keystore -storepass <password> -alias SERVER -rfc -file <diag_server_install_dir>/etc/<server_certificate_name>.cer
```

To use this command:

- Replace **<diag_server_install_dir>** with the path to the installation directory for the Diagnostics Server.
- Replace **<password>** with the string that you assigned as the **storepass** password when you created the keystore.
- Replace **<server_certificate_name>** with the name that you would like to assign to the certificate file. It is recommended that you assign a certificate name that will make it easy to recognize the component for which the certificate was created.

Use **diag_server_commander.cer** or **diag_server_mediator.cer**.

After this command runs, a certificate file with the name assigned in **<server_certificate_name>** is created in the **<diag_server_install_dir>/etc** directory for the Diagnostics Server, for example, **diag_server_commander.cer**.

Note: The certificate file must be imported to the host machines for each of the Diagnostics components that are expected to initiate communications with the Diagnostics Server. The instructions for importing the certificate file to each Diagnostics component are provided below.

3. Using the command in the following example, generate an obfuscated version of the **storepass** and the **keypass** passwords that you assigned when you created the keystore.
 - a. Replace **<diag_server_install_dir>** with the path to the installation directory for the Diagnostics Server.
 - b. Replace **<password>** with the string that you assigned as the password when you created the keystore.

```
<diag_server_install_dir>/JRE/bin/java -cp <diag_server_install_dir>/lib/ThirdPartyLibs.jar org.mortbay.util.Password <password>
```

The output from the obfuscation is shown in the following example. In this example, the password string was "testpass". The output consists of three lines. The original string that was to be obfuscated and two lines depicting the obfuscated password. Only the line that begins with "OBF" is used to set the properties in the following step of this process.

```
testpass
OBF:1ytc1vu91v2p1y831y7v1v1p1vv11yta
MD5:179ad45c6ce2cb97cf1029e212046e81
```

Note: If you did not use the same password for **keypass** and **storepass**, you must run this command twice to create an obfuscated version for each password.

4. Change the following properties in the file **<diag_server_install_dir>/etc/security.properties** for the Diagnostics commander server
 - a. Set **enableSSL=true**.
 - b. Set **keyStorePassword=<obfuscated_password>**.
 - c. Set **keyPassword=<obfuscated_password>**.

Note: The value entered for **<obfuscated_password>** must include the entire "OBF" line that was

output from the command in the previous step; for example:

```
keyStorePassword=0BF:1ytc1vu91v2p1y831y7v1v1p1vv11yta
```

- Restart the Diagnostics server.

Configure the Java Agent for Incoming HTTPS Connections

Note:

- Enabling SSL and HTTPS Communications for the Java Agent is supported on SUTs with the Sun, IBM, and JRockit JVMs. However, if you are using a JVM version prior to 1.4, you must download and install the Sun JSSE Optional Package onto the SUT server before you can enable SSL.

Other JSSE implementation, such as IBM's are not supported.

- When you enable HTTPS communications with a Java agent system, you must use port 45000 to run the Profiler UI, for example:
https://<my_probe_system>:45000.

Note: The location in which the agent is installed becomes the Diagnostics <probe_install_dir>. By default, the location is C:\MercuryDiagnostics\JavaAgent\DiagnosticsAgent on Windows and /opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent on UNIX.

To configure the Java Agent for incoming HTTPS connections:

- Generate a keystore in the <probe_install_dir>/etc directory using the following command:

```
/opt/MercuryDiagnostics/JavaAgent/JRE/bin/keytool -genkey -keystore <probe_install_dir>/etc/keystore -storepass <password> -alias PROBE -keyalg RSA -keypass <password> -dname "CN=<probe_hostname>, OU=Diagnostics, O=Micro Focus, L=Palo Alto, S=CA, C=USA" -validity 3650
```

To use this command example:

- Replace <probe_install_dir> with the path to the installation directory for the Java Agent.
- Replace <probe_hostname> with the machine name for the host of the Java Agent. This value cannot be the server's IP address. You should use the fully qualified domain name for the subject (CN) in the certificate.
- Replace each occurrence of <password> with the same password string. You can assign different passwords to the **storepass** and the **keypass**.

After you run this command, a keystore is created in <probe_install_dir>/etc/keystore with an entry called **PROBE** for the host of the Java Agent.

- Export the certificate for the PROBE entry in the keystore using the following command.

```
/opt/MercuryDiagnostics/JavaAgent/JRE/bin/keytool -export -keystore <probe_install_dir>/etc/keystore -storepass <password> -alias PROBE -rfc -file <probe_
```

```
install_dir>/etc/<probe_certificate_name>.cer
```

To use this command:

- Replace **<probe_install_dir>** with the path to the installation directory for the Java Agent.
- Replace **<password>** with the string that you assigned as the **storepass** password when you created the keystore.
- Replace **<probe_certificate_name>** with the name that you would like to assign to the certificate file. It is recommended that you assign a certificate name that will make it easy to recognize the component for which the certificate was created.

Include the type of the probe and the host name for the probe so that it will be easy to recognize the component for which the certificate was created; for example: **Java_probe_<probe_hostname>**.

After this command runs, a certificate file called **Java_probe_<probe_hostname>.cer** is created in the **<probe_install_dir>/etc** directory for the Java Agent.

Note: The certificate file must be imported to the host machines for each of the Diagnostics components that are expected to initiate communications with the Java Agent. The instructions for importing the certificate file to each Diagnostics component are provided below.

3. Using the command in the following example, generate an obfuscated version of the **storepass** and the **keypass** passwords that you assigned when you created the keystore.
 - a. Replace **<probe_install_dir>** with the path to the installation directory for the Java Agent.
 - b. Replace **<password>** with the string that you assigned as the password when you created the keystore.

```
/opt/MercuryDiagnostics/JavaAgent/JRE/bin/java -cp <probe_install_dir>/lib/ThirdPartyLibs.jar org.mortbay.util.Password <password>
```

The output from the obfuscation is shown in the following example. In this example, the password string was "testpass". The output consists of three lines. The original string that was to be obfuscated and two lines depicting the obfuscated password. Only the line that begins with "OBF" is used to set the properties in the following step of this process.

```
testpass
OBF:1ytc1vu91v2p1y831y7v1v1p1vv11yta
MD5:179ad45c6ce2cb97cf1029e212046e81
```

Note: If you did not use the same password for **keypass** and **storepass**, you must run this command twice to create an obfuscated version for each password.

4. Change the following properties in the file **<probe_install_dir>/etc/security.properties**.
 - a. Set **enableSSL=true**.
 - b. Set **keyStorePassword=<obfuscated_password>**.
 - c. Set **keyPassword=<obfuscated_password>**.

Note: The value entered for **<obfuscated_password>** must include the entire "OBF" line that was

output from the command in the previous step; for example:

```
keyStorePassword=0BF:1ytc1vu91v2p1y831y7v1v1p1vv11yta
```

Configure the Collector for Incoming HTTPS Connections

This section provides instructions for configuring the Collector to receive incoming HTTPS connections.

To configure the Collector for incoming HTTPS connections:

1. Generate a keystore in the **<collector_install_dir>/etc** directory using the following command:

```
<collector_install_dir>/JRE/bin/keytool -genkey -keystore <collector_install_dir>/etc/keystore -storepass <password> -alias COLLECTOR -keyalg RSA -keypass <password> -dname "CN=<collector_hostname>, OU=Diagnostics, O=Hewlett-Packard, L=Palo Alto, S=CA, C=USA" -validity 3650
```

To use this command example:

- Replace **<collector_install_dir>** with the path to the installation directory for the Collector.
- Replace **<collector_hostname>** with the machine name for the host of the Collector. This value cannot be the server's IP address. You should use the fully qualified domain name for the subject (CN) in the certificate.
- Replace each occurrence of **<password>** with the same password string. You can assign different passwords to the **storepass** and the **keypass**.

After you run this command, a keystore is created in **<collector_install_dir>/etc/keystore** with an entry called **COLLECTOR** for the host of the Collector.

2. Export the certificate for the COLLECTOR entry in the keystore using the following command.

```
<collector_install_dir>/JRE/bin/keytool -export -keystore <collector_install_dir>/etc/keystore -storepass <password> -alias COLLECTOR -rfc -file <collector_install_dir>/etc/<collector_certificate_name>.cer
```

To use this command:

- Replace **<collector_install_dir>** with the path to the installation directory for the Collector.
- Replace **<password>** with the string that you assigned as the **storepass** password when you created the keystore.
- Replace **<collector_certificate_name>** with the name that you would like to assign to the certificate file. It is recommended that you assign a certificate name that will make it easy to recognize the component for which the certificate was created.

Include the type of the collector and the host name for the collector so that it will be easy to recognize the component for which the certificate was created; for example: **collector_<collector_hostname>**.

After this command runs, a certificate file called **collector_<collector_hostname>.cer** is created in the **<collector_install_dir>/etc** directory for the Collector.

Note: The certificate file must be imported to the host machines for each of the Diagnostics

components that are expected to initiate communications with the Collector. The instructions for importing the certificate file to each Diagnostics component are provided below.

3. Using the command in the following example, generate an obfuscated version of the **storepass** and the **keypass** passwords that you assigned when you created the keystore.
 - a. Replace **<collector_install_dir>** with the path to the installation directory for the Collector.
 - b. Replace **<password>** with the string that you assigned as the password when you created the keystore.

```
<collector_install_dir>/JRE/bin/java -cp
<collector_install_dir>/lib/ThirdPartyLibs.jar org.mortbay.util.Password <password>
```

The output from the obfuscation is shown in the following example. In this example, the password string was `testpass`. The output consists of three lines. The original string that was to be obfuscated and two lines depicting the obfuscated password. Only the line that begins with "OBF" is used to set the properties in the following step of this process.

```
testpass
OBF:1ytc1vu91v2p1y831y7v1v1p1vv11yta
MD5:179ad45c6ce2cb97cf1029e212046e81
```

Note: If you did not use the same password for **keypass** and **storepass**, you must run this command twice to create an obfuscated version for each password.

4. Change the following properties in the file **<collector_install_dir>/etc/security.properties**.
 - a. Set **enableSSL=true**.
 - b. Set **keyStorePassword=<obfuscated_password>**.
 - c. Set **keyPassword=<obfuscated_password>**.

Note: The value entered for **<obfuscated_password>** must include the entire "OBF" line that was output from the command in the previous step; for example:

```
keyStorePassword=OBF:1ytc1vu91v2p1y831y7v1v1p1vv11yta
```

Enable Outgoing HTTPS Communication from Diagnostics Components

The following instructions provide you with the steps necessary to configure the Diagnostics components to send outgoing HTTPS communications to other Diagnostics components.

Note: The location in which the agent is installed is referred to as **<agent_install_dir>**.

To enable the Diagnostics commander server for outgoing communication to the Diagnostics mediator server via HTTPS:

1. Copy the certificate file from **<diag_server_install_dir>/etc/diag_server_mediator.cer** on the Diagnostics Server to **<diag_server_install_dir>/etc/diag_server_mediator.cer** on the Diagnostics commander server.
2. Change the value of the **trusted.certificate** property in the file **<diag_server_install_dir>/etc/security.properties** for the Diagnostics commander server.

3. Set **trusted.certificate=diag_server_mediator.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.
4. For incoming Diagnostics Server communication, indicate the URL for the Diagnostics Server by updating the following property in the file **<diagnostics_server_inst_dir>/etc/server.properties** on the Diagnostics commander server.

Set **commander.url** to **https://<diagserver_commander_hostname>:8443**

To enable the Diagnostics Mediator Server for outgoing communication to the Diagnostics commander server via HTTPS:

1. Copy the certificate file from **<diag_server_install_dir>/etc/diag_server_commander.cer** on the Diagnostics commander server to **<diag_server_install_dir>/etc/diag_server_commander.cer** on the Diagnostics mediator server.
2. Change the value of the **trusted.certificate** property in the file **<diag_server_install_dir>/etc/security.properties** for the Diagnostics mediator server.
3. Set **trusted.certificate=diag_server_commander.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.
4. For incoming Diagnostics Server communication, indicate the URL for the Diagnostics Server by updating the following property in the file **<diagnostics_server_inst_dir>/etc/server.properties** on the Diagnostics Server in Mediator mode.

Set **commander.url** to **https://<diagserver_commander_hostname>:8443**.

Note: When you enable HTTPS on the server, you must use port 8443 in the URL to run the Diagnostics UI.

To enable the Diagnostics Server (in Commander or Mediator mode) for outgoing communications to the probes via HTTPS:

1. Copy the certificate file from **<agent_install_dir>/etc/java_probe_<probe_host>.cer** for each probe to **<diag_server_install_dir>/etc/Java_probe_<probe_host>.cer** on the Diagnostics Server.
2. Change the value of the **trusted.certificate** property in the file **<diag_server_install_dir>/etc/security.properties** for the Diagnostics Server.

Set **trusted.certificate=Java_probe_<probe_host>.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.

To enable the Diagnostics Server in Mediator mode for outgoing communications to the collectors via HTTPS:

1. Copy the certificate file from **<collector_install_dir>/etc/collector_<collector_host>.cer** for each probe to **<diag_server_install_dir>/etc/collector_<collector_host>.cer** on the Diagnostics Server.
2. Change the value of the **trusted.certificate** property in the file **<diag_server_install_dir>/etc/security.properties** for the Diagnostics Server.

Set **trusted.certificate=collector_<collector_host>.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.

To enable the Java Agent for outgoing communications to the Diagnostics mediator server via HTTPS:

1. Copy the certificate file from **<diag_server_install_dir>/etc/diag_server_mediator.cer** on the Diagnostics mediator server to **<agent_install_dir>/etc/diag_server_mediator.cer** on the Java Agent.

2. Change the value of the **trusted.certificate** property in the file **<agent_install_dir>/etc/security.properties** for the Java Agent.

Set **trusted.certificate=diag_server_mediator.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.

3. For incoming Java Agent communication, indicate the URL for the Diagnostics mediator server by updating the following property in the file **<probe_inst_dir>/etc/dispatcher.properties**.

Set **registrar.url** to **https://<diagserv_mediatormode_hostname>:8443/registrar/**

Note: When you enable HTTPS on the server, you must use port 8443 in the URL to run the Diagnostics UI.

To enable the server/collector's embedded java probe for outgoing communications to the Diagnostics Server in Mediator mode via HTTPS:

1. Copy the certificate file from **<diag_server_install_dir>/etc/diag_server_mediator.cer** on the Diagnostics Server in Mediator mode to **<server/collector_install_dir>/probe/etc/diag_server_mediator.cer** on the embedded Java probe.
2. Change the value of the **trusted.certificate** property in the file **<server/collector_install_dir>/probe/etc/security.properties** for the embedded Java probe. Set **trusted.certificate=diag_server_mediator.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.
3. For incoming embedded Java probe communication, indicate the URL for the Diagnostics Server in Mediator mode by updating the following property in the file **<server/collector_install_dir>/probe/etc/dispatcher.properties**. Set **registrar.url** to **https://<diagserv_mediatormode_hostname>:8443/registrar/**.

To enable the Collector for outgoing communications to the Diagnostics mediator server via HTTPS:

1. Copy the certificate file from **<diag_server_install_dir>/etc/diag_server_mediator.cer** on the Diagnostics mediator server to **<collector_install_dir>/etc/diag_server_mediator.cer** on the Collector.
2. Change the value of the **trusted.certificate** property in the file **<collector_install_dir>/etc/security.properties** for the Collector.
Set **trusted.certificate=diag_server_mediator.cer**. If there are already other certificate files included in the value of this property, add the certificate file to the end of the list separated from the preceding value by a comma.
3. For incoming Collector communication, indicate the URL for the Diagnostics mediator server by updating the following property in the file **<collector_install_dir>/etc/collector.properties**.

Set **registrar.url** to **https://<diagserv_mediatormode_hostname>:8443/registrar/**

Note: When you enable HTTPS on the server, you must use port 8443 in the URL to run the Diagnostics UI.

To enable the .NET Agent for outgoing communications to the Diagnostics commander server via HTTPS:

1. Copy the certificate for the Diagnostics mediator server to the host for the .NET Agent. The certificate was generated when the Diagnostics mediator server was configured to receive HTTPS. See ["Enable Incoming HTTPS Communication for Diagnostics Components" on page 126](#) for instructions to configure the Diagnostics mediator server to receive HTTPS. If you followed the instructions in the referenced

section, the certificate can be found in `<diag_server_install_dir>/etc/diag_server_mediator.cer`.

2. On the Windows Taskbar, select **Start > Run**.
3. Run the Microsoft Management Console by typing `mmc`, and then clicking **OK**.
4. On the Microsoft Management Console menu, select **File > Add/Remove Snap-in** to display the Add/Remove Snap-in dialog.
5. Click **Add** on the Add/Remove Snap-in dialog.
6. Select **Certificates** from the Available Standalone Snap-in list and click **Add**.
7. In the Certificates Snap-in dialog box select **Computer account**, and click **Next**.
8. In the Select Computer dialog box, select **Local Computer: (the computer this console is running on)**, and then click **Finish**.
9. Click **Close** on the Add Standalone Snap-in.
10. Click **OK** on the Add/Remove Snap-in dialog.
11. On the Microsoft Management Console expand the listing for Certificates (Local Computer) in the left pane of the Console Root dialog.
12. Under Certificates (Local Computer), expand Trusted Root Certification Authorities.
13. Under Trusted Root Certification Authorities, right-click Certificates and select **All Tasks > Import** to start the Certificate Import Wizard.
14. Click **Next** to move past the Welcome dialog box of the Certificate Import Wizard.
15. Click **Browse** to navigate to the public keystore for the Diagnostics mediator server.
 - a. Select All Files (*.*) in Files of type.
 - b. Navigate to the directory where the keystore for the Diagnostics commander server was copied in step 1 and click Open. This should be: `<diag_server_install_dir>/etc/diag_server_mediator.cer`
16. Click **Next** to import the file.
17. Click **Next** to accept the default Certificate Store location of "Trusted Root Certification Authorities."
18. Click **Finish** on Completing the Certificate Import Wizard.
19. Click **OK** on the Certificate Import Wizard confirmation dialog.
20. Select **Certificates** under Trusted Root Certification Authorities to find the certificate you just added (it should be the hostname of the mediator server). Make a note of the value in the **Issued to** column. This value will be used for modifying the probe configuration files.
21. Edit the `<agent_install_dir>/etc/probe_config.xml` and change the `diagnosticsserver url` property to use the HTTPS URL: `<diagnosticsserver url="https://<diagnostics_mediator_server_host>:8443/commander" />`
22. Change the mediator host and port and add `ssl="true"`: `<mediator host="<diagnostics_mediator_server_host>" port="2612" metricport="8443" ssl="true"/>`
23. Edit `<agent_install_dir>/etc/metrics.config`. Change the `metrics.server.uri` value to specify the HTTPS URL: `metrics.server.uri = https://<diagnostics_mediator_server_host>:8443/metricdata/`

Note: For both the `probe_config.xml` and the `metrics.config` files, the `<diagnostics_mediator_server_host>` value must match the name that appears in the certificate. For example, if the hostname in the certificate is fully qualified, the hostname in the configuration files should also be fully qualified.

24. Restart IIS. For instructions on restarting IIS see the Diagnostics .NET Agent Guide.

To verify that you successfully configured the .NET probe for HTTPS communication with the Diagnostics commander server:

1. Browse to your .NET application to activate the .NET Agent.
2. Verify that the .NET Agent is available by checking the System Health view in the Diagnostics UI.

Enabling HTTPS Communication for APM Integrations

If the Diagnostics commander server is going to send data to a BSM/APM server in a hardened environment, you must configure the Diagnostics commander server to communicate securely with the BSM/APM Gateway Server.

The basic flow for any data collector connecting to secure BSM/APM is as follows:

- Obtain the appropriate root CA certificate(s) from the BSM/APM environment and import it into the JVM used by the data collector.
- Configure the connection to BSM/APM to use HTTPS.
- Make sure data flows over the secure connection.

Note: APM 9.40 and later (https) does not accept self signed certificates. When integrating Diagnostics with APM 9.40 or later, make sure the Diagnostics server certificates are not self signed.

To enable secure communications between the Diagnostics commander server and the BSM/APM Gateway Server:

1. (Optional) Enable the Diagnostics commander server for HTTPS communication as described in this chapter.
2. If your Diagnostics commander server is configured with SSL:
 - Copy the Diagnostics certificate file, **diag_server_commander.cer**, from the Diagnostics commander server installation directory, **<diag_server_install_dir>/etc/**, to the BSM/APM host.
 - Import the copied certificate, **diag_server_commander.cer**, into the BSM/APM server cacert keystore by running the following commands on all of the BSM/APM Gateway and Data Processing Servers, even if they are not accessed directly (for example, even if they are accessed using reverse proxy, or through a load balancer):

```

◦ <BSM/APM_server_install_dir>/jre/bin/keytool
  -import -file <copied_diag_certificate_directory>/diag_server_commander.cer
  -keystore <BSM/APM_server_install_dir>/jre/lib/security/cacerts -alias
  SERVER

```

```

◦ <BSM/APM_server_install_dir>/jre64/bin/keytool
  -import -file <copied_diag_certificate_directory>/diag_server_commander.cer
  -keystore <BSM/APM_server_install_dir>/jre64/lib/security/cacerts -alias
  SERVER

```

- Replace **<BSM/APM_server_install_dir>** with the path to the installation directory for the BSM/APM server host.
- Replace **<copied_diag_certificate_directory>** with the path to the copied Diagnostics certificate file.

Type **changeit** when you are prompted to enter the keystore password.

Type **yes** instead of the default **no** when you are asked if the certificate should be trusted.

- Restart all of the BSM/APM Gateway and Data Processing Servers.
3. Copy all root CA certificates (for example, for reverse proxy, Data Processing Servers, Gateway Servers) that were issued for the BSM/APM environment, **<BSM/APM_certificate_file.cer>**, to the Diagnostics Server host.
 4. Import the copied certificates into the Diagnostics Server cacert keystore by running the following command on the Diagnostics Server host.

```
<diag_server_install_dir>/jre/bin/keytool
-import -file <copied_BSM/APM_certificate_directory>/<BSM/APM_certificate_
file.cer>
-keystore <diag_server_install_dir>/JRE/lib/security/cacerts
```

- Replace **<diag_server_install_dir>** with the path to the installation directory of the Diagnostics commander server.
- Replace **<copied_BSM/APM_certificate_directory>** with the path to the copied BSM/APM certificate file.

Type **changeit** when you are prompted to enter the keystore password.

Type **yes** instead of the default **no** when you are asked if the certificate should be trusted.

5. The communication between the Diagnostics commander server and the BSM/APM Gateway Server is now secure. Select **HTTPS** for the Diagnostics Server protocol field when you register the Diagnostics commander server.

Configure Diagnostics Commander to Connect to a BSM/APM Server That Requires a Client Certificate

To configure a Diagnostics Commander to connect to a BSM/APM server that requires a client certificate, do the following:

1. Follow the steps described in ["Enabling HTTPS Communication for APM Integrations "](#) on the previous page to set up HTTPS communication between the Diagnostics Commander and BSM/APM.
2. Copy the APM server certificate (for example, **bsm_server.cer**) to the **<diag_server_install_dir>\etc** directory on the Diagnostics Commander host machine.
3. Modify the **<diag_server_install_dir>\etc\security.properties** file to add the server certificate as a trusted certificate. For example:

```
trusted.certificate=other.cer,bsm_server.cer
```

4. Obtain the APM client certificate and copy it to the Diagnostics Commander host machine (for example, **c:\client.pfx**).
5. Import the APM client certificate to the Diagnostics Commander by running the following command (with the applicable certificate names and paths) from the Diagnostics Commander host machine:

```
<diag_server_install_dir>\JRE\bin>keytool.exe -importkeystore -srckeystore c:\client.pfx -  
srcstoretype PKCS12 -destkeystore <diag_server_install_dir>\etc\keystore
```

6. Add required Java parameters to the **<diag_server_install_dir>\Server\nanny\windows\data\nanny\server.nanny** file. Edit the file and locate the line:

```
start=<diag_server_install_dir>\Server\jre\bin\javaw.exe^ -server
```

After this line, add the following two lines:

-Djavax.net.ssl.keyStore=<diag_server_install_dir>\Server\etc\keystore

-Djavax.net.ssl.keyStorePassword=<keystore password>

7. Restart the Diagnostics Commander server.

How to use System Views for Administrators

You can use the Diagnostics System views to monitor the health of the Diagnostics components and verify that they are working properly.

This chapter includes:

- ["System Views for Diagnostics' Administrators" below](#)
- ["System Health View Description" on the next page](#)
- ["System Capacity View Description" on page 141](#)

System Views for Diagnostics' Administrators

In large scale Diagnostics deployments you can use the System Views instead of the System Health Monitor. The specialized System Views allow you to quickly locate systems or groups of system based on various system attributes. Allowing you to more easily monitor system health and identify when systems are nearing capacity.

In many cases, the System views will be your first and only stop when you need to know information about the components in your Diagnostics deployment and the machines that host them. At a glance, you can determine which components are experiencing problems.

To access the System Views:

1. Open the Diagnostics UI as the System customer from `http://<Diagnostics_Commanding_Server_Name>:2006/query/`.
2. In the query page locate the System customer in the list and select the link to Open Diagnostics.
3. Log in to Diagnostics and on the Applications window select **Entire Enterprise** and select any link to open the Diagnostics Views.
4. In the Views pane you see the System Views view group. Open the view group and select either the **System Health** view or **System Capacity** view.

System Health View Description

The System Health view in the Diagnostics UI provides overall health information for the components you have installed in your Diagnostics environment.

Heartbeat	Name	Type	Host	Mediator	Port	Run	Events per sec	Ver
✓	CommandingServer	Commandi...	16.77.36.232		2006		0.9.25	
✓	server-ROSDIAGME...	mediator	ROSDIAGMED01.ovrtest...	Comma...	2612		1.9.25	
✓	server-rosdiagmed02	mediator	rosdiagmed02.ovrtest.adap...	Comma...	2612		0.9.25	
✓	OVRSHAREPOINT_...	probe	ovrsharepoint.ovrtest.adap...	Comma...	35005		0.9.25	
✓	OVRSHAREPOINT_...	probe	ovrsharepoint.ovrtest.adap...	Comma...	35001		0.9.25	
✓	ProbeAggregator-OV...	probe	ovrsharepoint.ovrtest.adap...	Comma...	45000		0.9.25	
✓	OVRSHAREPOINT_...	probe	ovrsharepoint.ovrtest.adap...	Comma...	35003		0.9.25	
✓	OVRSHAREPOINT_...	probe	ovrsharepoint.ovrtest.adap...	Comma...	35002		0.9.25	
✓	OVRSHAREPOINT_...	probe	ovrsharepoint.ovrtest.adap...	Comma...	35004		0.9.25	

System Capacity View Description

The System Capacity view in the Diagnostics UI provides information for managing capacity in your Diagnostics environment. This view shows the number of probe groups and probes that are assigned to each Diagnostics mediator.

Name	Host	Port	Probe Group Count	Probe Count								
server-ovresx1-vm5	ovresx1-vm5.ovrtest.adapps.com	2612	3	8								
<table border="1"> <thead> <tr> <th>Probe Group Name</th> <th>Probe Count</th> </tr> </thead> <tbody> <tr> <td>Sanity_LR_9_1_ovresx1-vm5</td> <td>4</td> </tr> <tr> <td>Sanity_CallChain_WebService_ovresx1-vm5</td> <td>3</td> </tr> <tr> <td>Sanity_Collectors_ovresx1-vm5</td> <td>1</td> </tr> </tbody> </table>		Probe Group Name	Probe Count	Sanity_LR_9_1_ovresx1-vm5	4	Sanity_CallChain_WebService_ovresx1-vm5	3	Sanity_Collectors_ovresx1-vm5	1			
Probe Group Name	Probe Count											
Sanity_LR_9_1_ovresx1-vm5	4											
Sanity_CallChain_WebService_ovresx1-vm5	3											
Sanity_Collectors_ovresx1-vm5	1											
server-SWROS018	WROS018.ovrtest.adapps.com	2612	4	8								
server-ovrntt100	ovrntt100.ovrtest.adapps.com	2612	4	8								
server-ovrsun28.rose.hp.com	ovrsun28.rose.com	2612	2	11								
CommandingServer	ovrntt150.ovrtest.adapps.com	2006	3	5								
server-ovrlxd14.ovrtest.adapps....	ovrlxd14.ovrtest.adapps.com	2612	1	9								
server-OVRNTT154	OVRNTT154.ovrtest.adapps.com	2612	3	5								
server-sw-vm118.ovrtest.adap...	hpsw-vm118.ovrtest.adapps.com	2612	1	10								
server-ovresx1-vm4.ovrtest.ada...	ovresx1-vm4.ovrtest.adapps.com	2612	6	17								

Diagnostics Data Management

Detailed information is provided on how Diagnostics data is managed and stored.

This chapter includes:

- ["About Diagnostics Data" below](#)
- ["Custom View Data" below](#)
- ["Performance History Data" on the next page](#)
- ["Data Retention" on page 146](#)
- ["Disk Space Issues on the Server" on page 149](#)
- ["Back Up Diagnostics Data" on page 149](#)
- ["To handle Diagnostics Data when Upgrading Diagnostics" on page 152](#)

About Diagnostics Data

There are two main types of Diagnostics data:

- The custom views that each user has created.
- Data collected by the probes and aggregated by the Diagnostics Servers. This data is stored in a time-series database on the Diagnostics Servers.

Each Diagnostics Server stores the data that is collected by the probes that report to it. In addition, the Diagnostics commander server stores the virtual transactions' data both for LoadRunner/Performance Center runs and Business Service Management as well as the application metrics. The organization and maintenance of the data files that make up the data base are described in this chapter.

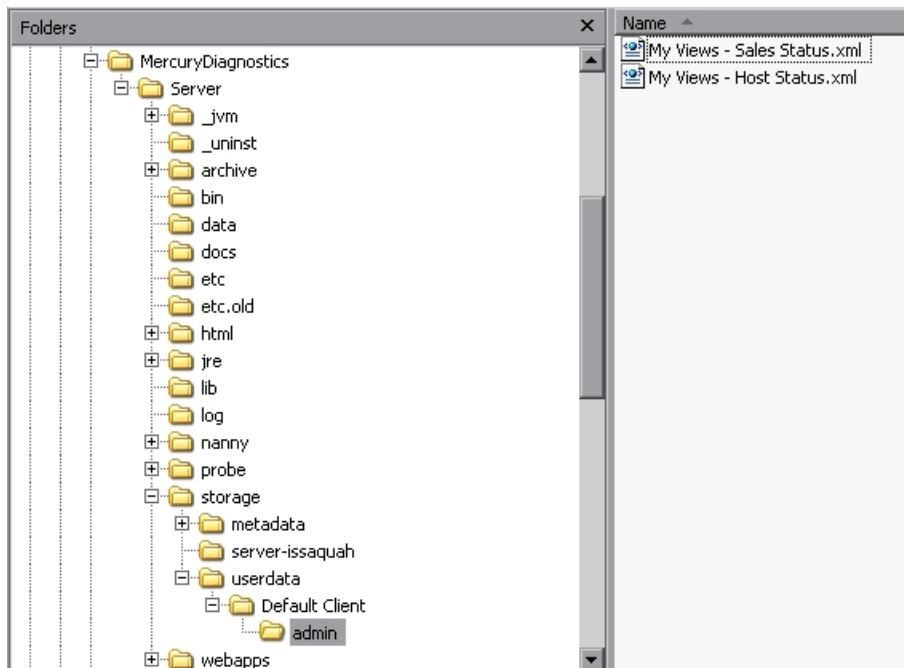
Custom View Data

Diagnostics users can create and save customized views as described in the chapter, "Customizing Diagnostics Views," in the Diagnostics User Guide. Diagnostics stores the customized views as XML files on the host for the Diagnostics commander server.

Custom View Data Organization

The user defined custom views are stored as XML files in the `<diag_server_install_dir>/storage/userdata` directory on the host for the Diagnostics commander server. The custom view files are relatively small.

Each user that has defined a custom view has their own custom view sub-directory in the **userdata** directory. For example, if the **admin** user created two custom views, Sales Status and Host Status, the two views would be stored as separate .xml files in the `<diag_server_install_dir>/storage/userdata/Default Client/admin` directory on the Diagnostics commander server as shown in the following example.



Performance History Data

Diagnostics stores the historical performance data in a time series database (TSDB) on the Diagnostics mediator server. If the Diagnostics Server has numerous probes reporting to it, the stored historical performance data can grow to many gigabytes of data. Although the amount of data collected for each application can vary in size, it is recommended that you plan for approximately 3 GB of data for each virtual machine that you are monitoring. For more information, see ["Data Retention" on page 146](#).

This section includes:

- ["Performance History Data Organization" below](#)
- ["Performance History Data File Types" on page 145](#)

Performance History Data Organization

The Diagnostics performance history data is in the `<diag_server_install_dir>/archive/mediator-<host_name>/persistence/<customer_name>_` directory of the Diagnostics mediator server where:

- **<host_name>** is the name of the host for the Diagnostics mediator server.
- **<customer_name>** is the customer name that you entered when you installed the Diagnostics mediator server. The name of this directory is the customer name with an appended underscore.

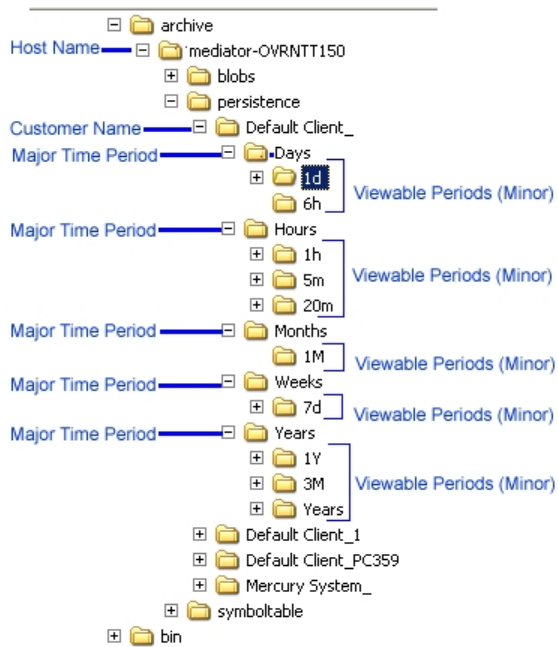
The archive **archive.dirname** property in **etc/server.properties** identifies where the archive should be stored (either absolute or relative to the `<diag_server_install_dir>`). If you would like to move or store the archive on a NAS drive (for example `//<hostname>/<sharename>`) the share you configure in **archive.dirname** needs to have read/write permissions for the user(s) Diagnostics will run as.

Note:

- Unless you are an Software-as-a-Service (SaaS) customer, the customer name should always be **Default Client**.

- The Diagnostics Performance history data for Performance Center or LoadRunner runs is stored in the **../persistence/<customer_name>_<run identifier>** directory.

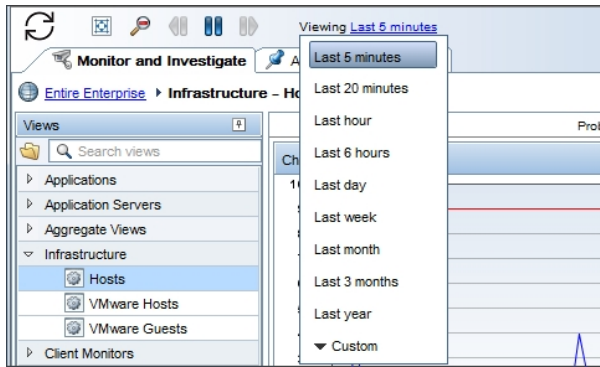
In the **/persistence** directory, the performance data is organized into directories as shown below:



The directory levels are referred to as **Major** time periods (Days, Hours Months, Weeks and Years) and the subdirectories are referred to as **Minor** time periods (1d, 6h, 1h, 20m, 5m, 1M, 7d, 1Y 3M, Years). These time periods are also referred to as data granularity.

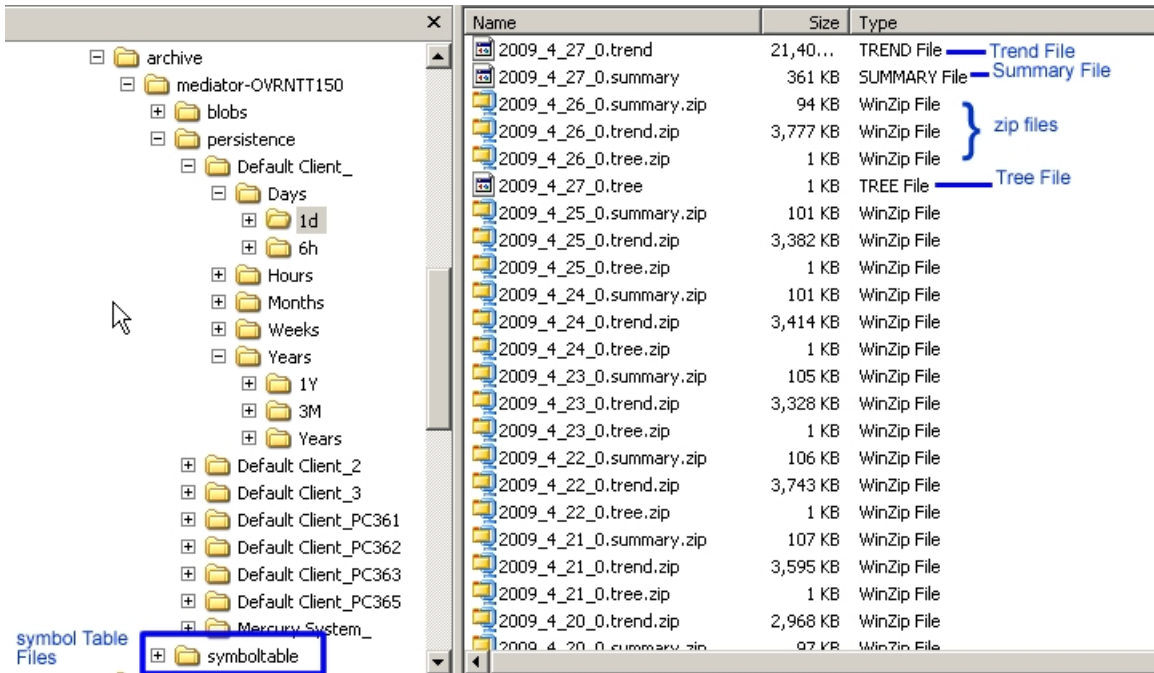
As seen in the directory example above, the Hours directory represents a major time period and has three subdirectories for the minor periods 5m, 20m, 1h.

These same minor time periods serve as the viewing periods in the UI. An example of the Viewing filter in Diagnostics is shown below:



Performance History Data File Types

The Diagnostics performance history data is stored in several types of files. An example of the directories with these files is shown below:



Symbol Table Files

The symbol tables contain string-to-integer mappings for small and fast data encoding of the other data files. For example, /login.do might be encoded as 1347854.

The symbol tables are stored in the `<diag_server_install_dir>/archive/mediator-<host_name>/symboltable` directory.

Summary Files

The summary files are accessed to display data in the Diagnostics View's entity tables and details pane. The Status shown in Diagnostics Views is based on summary files. Each viewable time period is stored in separate *summary* files.

Each summary file is named according to the minute (in GMT) at which Diagnostics started storing summary data in it. For example, a summary file that contained data beginning at midnight (GMT) on April 24, 2009 would be named **2009_4_27_0.summary**.

Trend Files

The trend files are accessed to display graph (charted) data in Diagnostics Views. To retrieve a trend for a minor time period, such as 5 minutes, only a small portion of the data in its major time period (1 hour in this case) trend file is read. Diagnostics stores the charted trend data for each major time period in *trend* files.

The trend files are named according to the minute (in GMT) at which the trended data that they contain was captured. For example, a file that contained hourly trend data starting at midnight April 24, 2009 would be named as follows: **2009_4_27_0.trend**.

When Diagnostics displays trended metrics in the Diagnostics views, it attempts to show approximately sixty data points for each viewable period so that the trend that is presented will be meaningful and easy to understand. To arrive at the data points needed for each viewable period, the Diagnostics Server consolidates the data from the appropriate tier in the data files. For example, when you are looking at trended data for the last hour, one data point per minute is shown in the graph. These data points were created by consolidating raw data points for twelve 5-second time periods.

Instance Tree Files

Instance tree files are accessed when you drill down to an instance call tree on the Diagnostics Server Requests view.

The *instance tree* files are similar to the trend files. There is a corresponding dump of the collected instance call trees for each major time period; for example, **2009_4_27_0.tree**.

Compressed Zip Files

Data in the instance tree, trend and summary files is for current time periods. The data in these files is uncompressed. The data files are quite large so they are automatically compressed after each time period is complete to save disk space. Compressed files have the same file names as the uncompressed file, but with a .zip extension; for example, **2009_4_26_0.summary.zip**.

Data Retention

Diagnostics uses data retention strategies that allow it to optimize its use of disk storage. Default settings for data retention are set out of the box. You should monitor your systems to check available disk space and change the data retention settings accordingly.

Data retention settings are taken into consideration in determining when purging occurs so if you see unexpectedly high disk space usage you should check your data retention settings to see if they need to be modified. See ["Symbol Table Purging" on page 148](#).

This section includes:

- ["Data Retention on the Mediators" below](#)
- ["Data Retention Configuration" on the next page](#)
- ["Symbol Table Purging" on page 148](#)

Data Retention on the Mediators

To make optimum use of disk space, historical performance data is stored in major and minor time periods with the data in each period retained based on the diagnostics data retention policy.

The data in the time periods with lower resolution data points is kept for longer periods of time to assist with such activities as capacity planning. The data in the time periods with high resolution data points is kept for shorter periods of time to assist with such activities as performance diagnostics. For this retention policy, measurements have shown that Diagnostics uses approximately 3 GB for each probed virtual machine.

In the table below you can see the Major time periods (directories as described in "[Performance History Data Organization](#)" on page 143) and the Minor time periods (viewable periods or subdirectories). The viewable periods under each directory are grouped together because they have the same resolution. So for example in the Days directory, 1 day and 6 hour data both have 5-minute resolution.

The following table illustrates the general Diagnostics data retention policy.

Major Time Period (Directory)	Minor Time Period (Viewable Period)	Trend Resolution	Data is kept for ... (Retention)
Hours	Hour, 20 minutes & 5 minutes	5 seconds	72 Hours
Days	Day & 6 Hours	5 minutes	93 Days
Weeks	Week	1 Hour	52 Weeks
Months	Month	6 Hours	24 Months
Years	Year, Quarter	1 Day	5 Years

The above table only applies when the entities do not change and are constantly available for the specified periods of time. See "[Symbol Table Purging](#)" on the next page

As shown in the table above, data for the last 1 hour, 20 minutes and 5 minutes viewing periods is kept for 72 hours while data for the last quarter is kept for 5 years.

Data Retention Configuration

You can configure data retention using the **server.properties** file on each mediator server. Default settings for data retention are set out of the box and the default settings for the major time periods are:

Major Time Period	Unit of Time	Default Time Period
0	Hours	72
1	Days	93
2	Weeks	52
3	Months	24
4	Years	5

The default settings suit most data retention requirements. If you must change the data retention policy, we recommended changing the major time period settings only. To do this, edit the **persistence.major.n.retention** setting for the required time period and change the value accordingly.

For example, the following graphic shows the default settings in the **server.properties** file for major time periods 0 and 1. To change the data retention period for time period 0 from 72 hours to 84 hours, change **persistence.major.0.retention=72** to **persistence.major.0.retention=84**.

```

persistence.major.durations.num=5

persistence.major.0.length=1
persistence.major.0.unit=h
persistence.major.0.retention=72
persistence.major.0.name=Hours
persistence.major.0.datapoints=720
...

persistence.major.1.length=24
persistence.major.1.unit=h
persistence.major.1.retention=93
persistence.major.1.name=Days
...

```

Symbol Table Purging

The purging mechanism must take a number of settings and other factors into account when determining when and how to purge data.

By default purging is set to run every 6 months (4320 hours). The purging interval can be modified in **server.properties** by changing the number of hours in the **persistence.major.4.total.length** parameter (requires restart of the server).

Note: Increasing the purging interval requires more server memory.

Data that is part of a snapshot is not purged since doing so renders the snapshot useless.

If a probe gets renamed or no data is received from the probe for 6 months (by default), Diagnostics automatically purges the probe and its data from the database.

A property can be specified for how much space you want the TSDB to use and data will typically be deleted to maintain a size less than this specified threshold. The **persistence.purging.threshold** property is set in the **<diag_server_install_dir>/etc/server.properties** file. But note that a number of other settings can take priority over this threshold value and can result in too much data being retained.

If you find that disk space is being exhausted this does not mean that purging isn't working it may mean that one of the following factors has affected the purging mechanism. For example if you have allocated 10GB of disk space on the server for Diagnostics but you see the archive at 20GB in danger of exhausting disk space on the system, this could be possible for any of the following reasons:

- Purging interval has not been reached yet. You can adjust to a shorter interval.
- There are a large number of snapshots on the system that by design do not get purged.
- The data retention settings may be requiring too much data to be retained. You may need to adjust data retention in order to save disk space (see ["Data Retention" on page 146](#)).

Data files that contain data for snapshots will not be purged. Since the TSDB is distributed and the snapshots only reside on the commanding server, a mechanism for determining what time ranges should be preserved is required.

When a snapshot is created, the commanding server will add the time range for the incident to the global list of preserved times. Upon snapshot deletion, this time range is removed. To allow multiple snapshots at the

same time, each snapshot creates a new preservation entry. Identical entries are not merged because the deletion handling would not be possible. The UI will inform the server that a time range needs to be preserved independently of snapshots. This will allow preservation of data for non-snapshot purposes.

When a server starts the purging process it will retrieve the preserved times from the commanding server. Failure to retrieve the list will cancel the purging process and it will be rerun at a later time. If the commanding server hasn't been contacted after 1 week, purging will be run without consideration for the preservation list to prevent unbounded growth on distributed servers in the case that the commanding server has been permanently taken offline.

No data will be deleted until the size of the archive has exceeded the purging threshold (**persistence.purging.threshold** property). After that, the policy defined below will be used to delete files until the archive is less than this threshold (provided other factors do not affect the purging mechanism). The threshold is set to 30 GB by default but you can change the value for **persistence.purging.threshold** in the **server.properties** file.

The purging process will scan the data files and identify all candidates for deletion. This process will operate on file sets. Since trend files are the largest consumer of disk space, and they require their summary files, data purging will be done based on trend file. For each trend file that exists, that does not have a preserved time range contained within it, a fileset containing all the files that are associated with that trend will be created. This will include the summary and tree file in the major summary that contains the trend file (only majors contain trend files), as well as all the minor summaries that index into that trend file.

Each fileset will have several values associated with it that will be used for determining which ones to purge. The "end time" of a fileset is the time of the last data point contained within the set. The "purge size" is the size of all the files in the summary that can be deleted.

Disk Space Issues on the Server

You can set up E-mail alerts to be sent to the Diagnostics administrator for disk space issues on the server. The Administrator E-mail address can be entered during the server installation or setup later using the Alert Properties page.

Alerts are issued if the server has less than 100MB of free disk space for the archive directory. If the server has less than 50MB of space the server stops collecting data and exits. The server will also not startup if there is less than 50MB of free disk space. These precautions help ensure the server stops collecting data before running out of disk space to maintain server stability.

The thresholds that determine these types of alerts for the Diagnostics administrator are factory configured in the server's **server.properties** file. For more details see the comments in this file for the various **watchdog** properties.

Back Up Diagnostics Data

It is recommended that you back up the Diagnostics data regularly so that it can be restored in the case of a disk or system failure.

If you use your own backup approach, you have to shutdown the server (because of Locked PathSymbolTable.pst). But use of the backup script provided with Diagnostics is recommended.

If your Diagnostics deployment requires that the Diagnostics Server have high availability, you can create a standby Diagnostics Server for each Diagnostics Server. The standby is then ready to be used during a hardware failure or other problem with the host of the Diagnostics Server. See ["Prepare a High Availability Diagnostics Server" on page 76](#).

This section includes:

- ["Backing up Data Remotely" below](#)
- ["Configure Symbol Table Backup" on the next page](#)
- ["Restore Data After a Failure" on the next page](#)

Backing up Data Remotely

Remote backup is possible by downloading the Diagnostics data files over HTTP to a local directory, and backing up that directory using your normal backup procedures.

The Diagnostics Server also supports the HTTP If-Modified-Since and Request-Range headers ("re-get") to allow standard HTTP mirroring software to download or incrementally update these files. If you choose to use your own HTTP mirroring software, work with your Micro Focus support representative to make sure the files are backed up in the proper order to ensure data integrity.

Diagnostics is installed with a remote backup script stored at `<diagnostics_server_install_dir>/server/bin/remote-backup.sh`. The UNIX script uses the `wget` utility (<http://www.gnu.org/software/wget/wget.html>) to download incrementally over HTTP. On Windows, Cygwin (<http://www.cygwin.com/>) can be used to run this script.

There is also a `remote-backup.cmd` for Windows. The `.cmd` script requires `wget.exe` to be located in the `<diag_server_install_dir>/server/bin/wget` directory (the `.sh` script just requires `wget` in the path).

The backup script can backup data remotely and from that directory you can do your traditional backup if you want. The script can also backup data to a local directory (ideally another drive on the same host).

Note: The backup script supplied with the Diagnostics Server backs up the data in a specific order. Failing to back up files in the correct order causes the restored backup to be unusable. It is therefore recommended to always use the supplied script to create data backups.

The following table lists the `remote-backup.sh` parameters:

Parameter	Description
<code>-h</code>	The host (or IP address) to download from
<code>-o</code>	The directory to store the backup in
<code>-u</code>	The HTTP username to use Default: admin
<code>-p</code>	The HTTP password to use Default: admin
<code>-P</code>	The HTTP port number to use (optional) Default: 2006
<code>-r</code>	The ID of the Diagnostics Server in Mediator mode being read from (for <code>rhttp</code> backups) (optional). For example, if you have 2 servers "commander" and "mediator", you could backup mediator over <code>rhttp</code> with: <code>-h commander -r mediatorId</code> .
<code>-c</code>	The clean option. When specified, files that exist in the output directory and do not exist on the server will be removed (the others need to be kept for better performance with the timestamping feature on download).

Parameter	Description
-v	Specified for more verbose output.

For example, to back up a Diagnostics Server running on the dragonfly machine into the **dragonfly-backup** directory:

```
% mkdir dragonfly-backup
% bin/remote-backup.sh -u admin -p secret -h dragonfly -o dragonfly-backup
```

The data is backed up in the following directories:

Data	Backup Directory
Server configuration	etc/
User custom views	storage/userdata
Raw performance history data	archive/.../persistence/
Symbol table	archive/.../symboltable/

Configure Symbol Table Backup

Sometimes the backup folder for the jdb files under symboltable use up a lot of disk space when there are a large number of symbol files. So you can configure backing up the symbol table as follows:

- You can enable or disable backing up the symbol table by setting the **symboltable.backup** property to true or false in the **server.properties** file.
- You can configure symbol table backup frequency by setting the **symboltable.backup.majors** property in the **server.properties** file.

Set the **symboltable.backup.majors** property using a comma separated list, to the desired backup frequency (Days, Weeks, Months). The frequency values are the same as defined by the **persistence.major.<n>.name** property (see "[Data Retention Configuration](#)" on page 147). For example, to backup the symbol table weekly, use the **persistence.major.2.name** which is Weeks.

The default configuration for symbol table backup as defined in **server.properties** is:

```
# Should the server backup the symboltable?
symboltable.backup = true
# Which majors should be backed up?
symboltable.backup.majors = Days,Weeks,Months
```

Restore Data After a Failure

The files in the backup directory are stored in the structure used by the Diagnostics Server.

To restore the time series database from the backup:

1. Install a clean Diagnostics Server. The Diagnostics Server is started automatically after the installation completes.
2. Shut down the Diagnostics Server.

3. Make sure that the Diagnostics Server has been shut down by verifying that there are no java/javaw processes in your process list. On Windows systems, you can use the Task Manager to do this and on UNIX systems, you can use ps.
4. Delete the `<diag_server_install_dir>/archive` directory from Diagnostics Server.
5. Copy the database backup to replace the `<diag_server_install_dir>/archive`.
6. If the host name for the Diagnostics Server has changed since the backup was taken you must update the directory name that is based on the Diagnostics Server host name to reflect the new host name.
Rename `<diag_server_install_dir>/archive/mediator-<host-name>` so that `<host-name>` reflects the new Diagnostics Server host name. For example, if host name in the backup was `oldhost` and the new host name is `newhost` you would change `<diag_server_install_dir>/archive/mediator-oldhost` to `<diag_server_install_dir>/archive/mediator-newhost`

Index Regeneration

When a restored Diagnostics Server is first started, the indexed data, which was not backed up, must be regenerated. Index regeneration is started automatically in the background and could take several hours to complete. While the indexes are regenerated, the Diagnostics Server is able to receive events from probes, but some historical data cannot be displayed in the Diagnostics views until the restoration is complete.

Known Limitation

In Diagnostics, binary data is written in the native byte order. This means that a Diagnostics data backup from a Big Endian machine cannot be restored and used on a Little Endian machine.

To handle Diagnostics Data when Upgrading Diagnostics

For details on handling Diagnostics data when upgrading, see "[Upgrade Overview](#)" on page 43.

General Reference Information

This section includes general reference topics.

This chapter includes:

- ["UNIX Commands" below](#)
- ["Regular Expressions" below](#)
- ["Multi-Lingual User Interface Support" on page 158](#)

UNIX Commands

When running an installation on a UNIX platform, you can usually follow the instructions that appear on the screen. The on-screen instructions can be confusing at times.

If something is unclear, use the following guidelines:

- To select an option from a list of options, type the number corresponding to the option and press **Enter**. Then type 0 and press **Enter** again to confirm your choice.
- When selecting multiple options, for each selection type the corresponding number and press **Enter**. After you finish selecting all your options, type 0 and press **Enter** again to confirm your choices.
- If you selected an option and want to clear it, retype the corresponding number, or type the number of another option, and press **Enter**. Then type 0 and press **Enter** again to confirm your choice.
- When entering information at a prompt:
 - To accept a default value that is displayed at the prompt, press **Enter**.
 - Type the information and press **Enter** to continue.
- To continue with the next step of an installation, type 1 to select **Next**, and press **Enter**.
- To go back to previous prompts to make changes, type 2 to select **Previous** and press **Enter**.
- To cancel an installation, type 3 to select **Cancel** and press **Enter**.
- To redisplay a prompt, type 4 to select **Redisplay** and press **Enter**.

Regular Expressions

When you specify the instrumentation definitions for each probe in the capture points file, you can use regular expressions for most of the arguments in a point.

A regular expression is a string that specifies a complex search phrase. By using special characters, such as a period (.), asterisk (*), caret (^), and brackets ([]), you can define the conditions of a search.

Note: Regular expressions in Diagnostics must be prefaced with an exclamation point.

By default, Diagnostics treats all characters in a regular expression literally, except for the period (.), hyphen (-), asterisk (*), caret (^), brackets ([]), parentheses (()), dollar sign (\$), vertical line (|), plus sign (+), question mark (?), and backslash (\). When one of these special characters is preceded by a backslash (\), Diagnostics treats it as a literal character.

Common Regular Expression Operators

This section describes some of the more common operators that can be used to create regular expressions.

Note: For a complete list and explanation of supported regular expression characters, see the Regular Expressions section in the Microsoft VBScript documentation.

Operator	Purpose
(\)	Rendering Special Characters Literal Creating Special Characters out of Literal Characters See "Use the Backslash Character" below
(.)	"Match Any Single Character" on the next page
([xy])	"Match Any Single Character in a List" on the next page
([^xy])	"Match Any Single Character Not in a List" on the next page
([x-y])	"Match Any Single Character within a Range" on the next page
(*)	"Match Zero or More Specific Characters" on page 156
(+)	"Match One or More Specific Characters" on page 156
(?)	"Match Zero or One Specific Character" on page 156
(())	"Group Regular Expressions" on page 156
()	"Match One of Several Regular Expressions" on page 156
(^)	"Match the Beginning of a Line" on page 156
(\$)	"Match the End of a Line" on page 157
(\w)	"Match Any AlphaNumeric Character Including the Underscore" on page 157
(\W)	"Match Any Non-AlphaNumeric Character" on page 157

Use the Backslash Character

A backslash (\) can serve two purposes. It can be used in conjunction with a special character to indicate that the next character be treated as a literal character. For example, \. would be treated as period (.) instead of a wildcard (see ["Match Any Single Character"](#) on the next page).

Alternatively, if the backslash (\) is used in conjunction with some characters that would otherwise be treated as literal characters, such as the letters n, t, w, or d, the combination indicates a special character. For example, \n stands for the newline character.

Here is an example:

- w matches the character w
- \w is a special character that matches any word character including underscore

- `\\` matches the literal character `\`
- `\(` matches the literal character `(`

For example, if you were looking for a file called:

```
filename.ext
```

the period would be mistaken as an indication of a regular expression. To indicate that the period is not part of a regular expression, you would enter it as follows:

```
filename\.ext
```

If a backslash character is used before a character that has no special meaning, the backslash is ignored. For example, `\z` matches `z`.

Match Any Single Character

A period (`.`) instructs Diagnostics to search for any single character (except for `\n`); for example:

```
welcome.
```

matches **welcomes**, **welcomed**, or **welcome** followed by a space or any other single character. A series of periods indicates the same number of unspecified characters.

To match any single character including `\n`, enter:

```
(.\n)
```

For more information on the `()` regular expression characters, see ["Group Regular Expressions" on the next page](#). For more information on the `|` regular expression character, see ["Match One of Several Regular Expressions" on the next page](#).

Match Any Single Character in a List

Square brackets instruct Diagnostics to search for any single character within a list of characters. For example, to search for the date 1967, 1968, or 1969, enter:

```
196[789]
```

Match Any Single Character Not in a List

When a caret (`^`) is the first character inside square brackets, it instructs Diagnostics to match any character in the list except for the ones specified in the string; for example:

```
[^ab]
```

matches any character except **a** or **b**.

Note: The caret has this special meaning only when it is the first character displayed within the brackets.

Match Any Single Character within a Range

To match a single character within a range, you can use square brackets (`[]`) with the hyphen (`-`) character. For example, to match any year in the 1960s, enter:

```
196[0-9]
```

A hyphen does not signify a range if it is displayed as the first or last character within brackets, or after a caret (`^`).

For example, `[-a-z]` matches a hyphen or any lowercase letter.

Note: Within brackets, the characters ".", "*", "[", and "\" are literal. For example, `[.*]` matches `.` or `*`. If the right bracket is the first character in the range, it is also literal.

Match Zero or More Specific Characters

An asterisk (*) instructs Diagnostics to match zero or more occurrences of the preceding character; for example:

```
ca*r
```

matches **car**, **caaaaaar**, and **cr**.

Match One or More Specific Characters

A plus sign (+) instructs Diagnostics to match one or more occurrences of the preceding character; for example:

```
ca+r
```

matches **car** and **caaaaaar**, but not **cr**.

Match Zero or One Specific Character

A question mark (?) instructs Diagnostics to match zero or one occurrences of the preceding character; for example:

```
ca?r
```

matches **car** and **cr**, but nothing else.

Group Regular Expressions

Parentheses (()) instruct Diagnostics to treat the contained sequence as a unit, just as in mathematics and programming languages.

Using groups is especially useful for delimiting the argument(s) to an alternation operator (|) or a repetition operator (*, +, ?, { }).

Match One of Several Regular Expressions

A vertical line (|) instructs Diagnostics to match one of a choice of expressions; for example:

```
foo|bar
```

causes Diagnostics to match either **foo** or **bar**.

```
fo(o|b)ar
```

causes Diagnostics to match either **fooar** or **fobar**.

Match the Beginning of a Line

A caret (^) instructs Diagnostics to match the expression only at the start of a line, or after a newline character.

Here is an example:

```
book
```

matches **book** within the lines **book**, **my book**, and **book list**, while

`^book`

matches **book** only in the lines **book** and **book list**.

Match the End of a Line

A dollar sign (\$) instructs Diagnostics to match the expression only at the end of a line, or before a newline character; for example:

`book`

matches **book** within the lines **my book**, and **book list**, while a string that is followed by (\$), matches only lines ending in that string; for example:

`book$`

matches **book** only in the line **my book**.

Match Any AlphaNumeric Character Including the Underscore

`\w` instructs Diagnostics to match any alphanumeric character and the underscore (A-Z, a-z, 0-9, _).

Here is an example:

`\w*` causes Diagnostics to match zero or more occurrences of the alphanumeric characters—**A-Z, a-z, 0-9**, and the underscore (_). It matches **Ab, r9Cj**, or **12_uYLgeu_435**.

Here is an example:

`\w{3}` causes Diagnostics to match 3 occurrences of the alphanumeric characters—**A-Z, a-z, 0-9**, and the underscore (_). It matches **Ab4, r9_**, or **z_M**.

Match Any Non-AlphaNumeric Character

`\W` instructs Diagnostics to match any character other than alphanumeric characters and underscores.

Here is an example:

`\W` matches **&, *, ^, %, \$, and #**.

Combine Regular Expression Operators

You can combine regular expression operators in a single expression to achieve the exact search criteria you need.

For example, you can combine the `.` and `*` characters to find zero or more occurrences of any character (except `\n`).

For example,

`start.*`

matches **start, started, starting, starter**, and so forth.

You can use a combination of brackets and an asterisk to limit the search to a combination of non-numeric characters; for example:

`[a-zA-Z]*`

To match any number between 0 and 1200, you must match numbers with 1 digit, 2 digits, 3 digits, or 4 digits between 1000-1200.

The regular expression below matches any number between 0 and 1200.

([0-9]?[0-9]?[0-9]|1[01][0-9][0-9]|1200)

Multi-Lingual User Interface Support

The Diagnostics user interface (UI) can be viewed in multiple languages in your Web browser. This applies, when Diagnostics is integrated with BSM/APM or running in standalone mode (no integration).

If Diagnostics is integrated with LoadRunner or Performance Center, the display language of the UI is determined by the client locale setting (defined in the Regional Settings of your operating system).

Note: Diagnostics does not support localization of agent names.

This appendix explains how to view the Diagnostics user interface in a specific language. The Diagnostics UI can be viewed in the following languages in your Web browser:

Language	Language preference in Web browser
English	English
Simplified Chinese	Chinese (China) [zh-cn], Chinese (Singapore) [zh-sg]
Korean	Korean [ko]
Japanese	Japanese [ja]

You use the language preference option in your browser to select how you view Diagnostics. The language preference chosen affects only the user's local machine and not the Diagnostics Server or any other user accessing the same Diagnostics Server.

To view the Diagnostics UI in a specific language:

1. Install the appropriate language's fonts on your local machine if they are not yet installed. If you choose a language in your Web browser whose fonts are not installed, the Diagnostics user interface uses the default language of your local machine.

Assume, for example, that the default language on your local machine is English and the Web browser is configured to use Japanese. If Japanese fonts are not installed on the local machine, the Diagnostics user interface is displayed in English.

2. If you are using Internet Explorer, configure the Web browser on your local machine to select the language in which you want to view the Diagnostics user interface. For details, see the Microsoft Web site, <http://support.microsoft.com/kb/306872/en-us>.

Continue with step 4.

3. If you are using FireFox, configure the Web browser on your local machine as follows:
 - a. Select **Tools > Options > Advanced**. Click **Edit Languages**. The Language dialog box opens.
 - b. Highlight the language in which you want to view Diagnostics.
If the language you want is not listed in the dialog box, expand the **Select language to add** list, select the language, and click **Add**.
 - c. Click **Move Up** to move the selected language to the first row.
 - d. Click **OK** to save the settings. Click **OK** to close the Language dialog box.

4. Close your existing browser and open Diagnostics in a new browser. The Diagnostics user interface is displayed in the selected language.

Data Exporting

The metric data collected by Diagnostics can be archived directly to a third-party database where it can be retained or it can be formatted into reports as supported by the database.

This data export is accomplished by using XPath-like queries to pull the desired metrics from the Diagnostics Time Series database (TSDB), which is the repository for all persistent Diagnostics data. For information about the TSDB, see ["Diagnostics Data Management" on page 142](#).

This chapter includes:

- ["Task 1: Prepare the target database" below](#)
- ["Task 2: Determine which metrics you want to export" on the next page](#)
- ["Task 3: Determine the frequency and the recovery period" on page 162](#)
- ["Task 4: Modify the data export configuration file for summary data" on page 163](#)
- ["Task 5: Modify the data export configuration file for trend data \(1 minute data granularity\)" on page 166](#)
- ["Task 6: Monitor the data export operation" on page 169](#)
- ["Task 7: Verify the results" on page 170](#)
- ["Task 8: Select the data from the target database" on page 170](#)
- ["Sample Queries" on page 171](#)

Task 1: Prepare the target database

The target database for the exported data can be an SQL Server or Oracle database to which the Diagnostics commanding server has access. For the most recent information on supported environments, see the *Diagnostics System Requirements and Support Matrix* guide on the [Software Support web site](https://softwaresupport.softwaregrp.com/) (<https://softwaresupport.softwaregrp.com/>) (missing or bad snippet).

The data export performed by the Diagnostics server automatically creates the schema and tables in the target database. The target database should have at least 1 GB of space available. During the first few export operations, you should monitor the size of the database to see if more space is needed.

Note: For specific database considerations when exporting trend data, refer to the "Notes and Limitations" section in ["Task 5: Modify the data export configuration file for trend data \(1 minute data granularity\)" on page 166](#).

To connect to the target database, you must specify the login credentials for a user that has read/write privileges to the database and has table definition privileges.

Note: If you are upgrading to Diagnostics 9.10 or later but you want to keep the old pre-9.10 database content you can alter the database to gain the following new functionality.

- In 9.10 or later the min and max values use doubles instead of integers allowing exported data to show decimal places. Alter the database after upgrading as follows:

Oracle:

```
ALTER TABLE RECORD MODIFY (  
REC_COUNT NUMBER(38),  
TOTAL FLOAT,
```

```
MINIMUM FLOAT,
MAXIMUM FLOAT)
```

SQL Server:

```
ALTER TABLE RECORD ALTER COLUMN REC_COUNT DECIMAL(19)
ALTER TABLE RECORD ALTER COLUMN TOTAL FLOAT
ALTER TABLE RECORD ALTER COLUMN MINIMUM FLOAT
ALTER TABLE RECORD ALTER COLUMN MAXIMUM FLOAT
```

Task 2: Determine which metrics you want to export

There are different ways to control which metrics are exported. You can specify to get all metrics for a particular entity type. You can exclude metrics from that grouping or you can specify to include only specific metrics. For more information on the Diagnostics data model see the *Diagnostics Data Model and Query API* document available on the DVD and in the help.

Note: The data export operation exports metric data only, that is counts, latencies, and averages. No instance data or status data is exported. Also you cannot export call profile data.

Metrics are grouped by the entity type to which they apply as well as other criteria. The following entity type groupings are the most commonly used:

Entity Type	Description
/probegroup/probe	Metrics for all probes across all probe groups.
/probegroup/probe/fragment	Metrics for all server requests across all probe groups and probes.
/probegroup/index[name='rollup_fragment']/fragment	Metrics for server requests rolled up by probe across all probe groups.
/probegroup/probe/index[name='services']/service	Metrics for Web services (excluding operation) across all probe groups and probes.
/index [equals(name,'apps')]/app/app_metrics	Metrics for a particular application.
/probegroup/probe[equals(probeType,'Oracle')]	Metrics for all Oracle collectors.
/probegroup/probe[equals(probeType,'SqlServer')]	Metrics for all SqlServer collectors.
/host -- Metrics for all hosts (various system metrics).	
/txn -- Metrics for all BPM transactions.	

The following tables give examples of types of metrics and the categories they belong to:

Category	Metric
Classes	Classes Currently Loaded Classes Loaded Classes Unloaded
Dynamic Caching	Caching Current Cache Size Caching Max Cache Size
EJB	EJB Activates EJB Activation Time EJB Committed Transactions / sec EJB Concurrent Active Methods EJB Concurrent Live Beans EJB Create Time EJB Creates EJB Drain Size EJB Drains From Pool EJB Frees EJB Gets Found EJB Gets From Pool EJB Instantiates EJB Load Time EJB Loads EJB Passivates EJB Passivation Time EJB Passive Beans EJB Pools Size EJB Ready Beans EJB Remote Time EJB Removes EJB Response Time EJB Returns Discarded EJB Returns To Pool EJB Rolled Back Transactions / sec EJB Store Time EJB Stores
EJB (Continued)	EJB Timed Out Transactions / sec EJB Total Method Calls EJB-Cache Access / sec EJB-Cache Beans Cached EJB-Cache Get Failures / sec EJB-Pool Access / sec EJB-Pool Available Beans EJB-Pool Beans in Use EJB-Pool Current Waiters EJB-Pool Get Failures / sec EJB-Pool Get Timeouts / sec

Category	Metric
Execute Queues	Execute Queues Idle Threads Execute Queues Pending Requests Execute Queues Requests / sec Execute Queues Total Threads
GC	GC Collections/sec GC Time Spent in Collections
Http Status	5xx-6xx
J2C Connections	J2C Connection Handles J2C Connection Released J2C Connections Allocated J2C Connections Closed J2C Connections Created
JDBC	JDBC Connections Created/sec JDBC Create Connection Delay JDBC Current Capacity JDBC Execute Statement JDBC Leaked Connections JDBC Reconnect Failures JDBC Requests Waiting for Connection JDBC Statement Cache Accesses / sec JDBC Statement Cache Hits / sec JDBC Statement Cache Size JDBC Total Connections Opened JDBC Wait Seconds High
Latency	latency total_cpu exception_count timeout_count throughput

You specify the group or individual metric to export as described in Task 4.

Note: When exporting trend data, which has a granularity of one minute, you must select specifically identified individual metrics to avoid overloading the mediator and database, which could negatively affect performance. For details on exporting trend data, see "[Task 5: Modify the data export configuration file for trend data \(1 minute data granularity\)](#)" on page 166.

Task 3: Determine the frequency and the recovery period

Each export operation has a specified frequency which controls how often it occurs and therefore the granularity of the returned metrics. The recommended frequency is 1h (hourly) which means that every hour the export operation is run. Other options for frequency are: 5m and 1d.

Note:

- When exporting trend data, the granularity is set to one minute by default and cannot be changed, regardless of the configured frequency.
- The data export operation can be run as frequently as desired however the data export operation affects the Diagnostics Server performance. The higher the frequency, the greater the load on the server. You can configure the export processing of the mediators in batches to reduce the load on the commanding server (**servers-per-query** attribute set in the **etc/data-export-config.xml** file on the server).

You can also specify a frequency recovery period. The recovery period is used only when the commanding Diagnostics Server is shut down or becomes otherwise unavailable. This value tells the commanding Diagnostics Server how far back to go to resume running the data export operations when it resumes operation.

The frequency recovery period formula is:

```
(current time) - (recovery-periods * frequency)
```

For example, assume a commanding Diagnostics Server was not active for 24 hours. A data export operation with an hourly frequency has missed a minimum of 23 executions. By default, the data export operations would start querying at the time the outage occurred (24 hours in the past). The metrics for the hourly data were aggregated into larger buckets and, therefore, the returned metrics are not meaningful.

However, if the recovery periods is specified as 6h, the hourly task would go back 6 hours in time (instead of 24) to start its querying against the TSDB. These metrics are meaningful.

Set the `<frequency>` and `<recovery-periods>` elements as described in Task 4.

Task 4: Modify the data export configuration file for summary data

The queries that export the Diagnostics data are defined in `<diag_server_install_dir>/etc/data-export-config.xml` file of the Diagnostic Commander Server. You can configure both summary and trend data queries at the same time in the `data-export-config.xml` file. For details on configuring trend data queries, see ["Task 5: Modify the data export configuration file for trend data \(1 minute data granularity\)" on page 166](#).

Follow these steps to set up this file:

1. Make a backup copy of the `<diag_server_install_dir>/etc/data-export-config.xml` file if desired.
2. Open the `<diag_server_install_dir>/etc/data-export-config.xml` file for editing.
3. Locate the `<enabled>` element and set it to true:

```
<enabled>true</enabled>
```

This element is used to turn on or off the data export operation. You should disable the data export operation when it is not needed to avoid unnecessary system overhead. By default the data export operation is disabled.

4. Locate the `<customer name>` element and set it to the customer name:

Unless you are a SaaS customer, the customer name should always be Default Client.

```
<customer name='Default Client'>
```

5. Locate the **<db-target>** element and enter the driver name, connection URL, user name and password (encrypted or plain text) for the target database.

For example, for SQL Server with an encrypted password:

```
<db-target>
  <driver>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver>
  <connection-
url>jdbc:sqlserver://testapps.mycompany.com:1433;databaseName=DIAG</
connection-url>
  <user>sa</user>
  <encrypted-password>0BF:1ym51y0s1uo71z0f1unr1y0y1ym9</encrypted-password>
  <batchsize>200</batchsize>
</db-target>
```

For example, for Oracle with an unencrypted password:

```
<db-target>
  <driver>oracle.jdbc.driver.OracleDriver</driver>
  <connection-url>jdbc:oracle:thin:@testapps.mycompany.com:1521:ORCL</connection-
url>
  <user>diagfan</user>
  <password>tiger2</password>
  <connection-property name="oracle.jdbc.defaultNChar" value="true"/>
</db-target>
```

For Oracle databases, the **oracle.jdbc.defaultNChar** property must be set to true when UTF8/UTF15 character support is required.

To encrypt a database password, use the Diagnostics password encryptor. See "Password Obfuscation" in the Diagnostics Collector Guide.

For the **<batchsize>** element, specify the batch size in units used for optimal JDBC PreparedStatement execution. By default, this is set to 100. Large implementations with large payloads require adjustments to the default.

6. For each set of metrics that you want to export, specify the following in the **<query>**:
 - **id=** A name which identifies the query being defined. Must be unique to this data-export-config.xml file. Spaces are not allowed in the id value.
 - **frequency=** A string value that specifies how often to run the query. Options are: 1h, 5m, and 1d.
 - **recovery-periods=** specifies how far back in time to start querying after an outage occurs.
 - **<entity-path>** One of the entity path groupings described in Task 2.
 - **<init-query-time>** or **<init-query-periods>** A time in the past at which the query starts. **<init-query-time>** is a time value specified in standard XSD time format. **<init-query-periods>** is an integer that is multiplied by the frequency to determine the query time. If omitted, the query runs at the next frequency boundary.

For example, the following entry creates a query that runs every hour and returns all of the metrics for all probes in all probe groups. If an outage occurs, only recover back 2 hours from current time:

```
<query id="Probes" frequency="1h" recovery-periods="2">
  <entity-path>/probegroup/probe</entity-path>
</query>
```

The following entry creates a query that runs every hour and returns every hour's rollup fragment latency metrics for all probe groups:

```
<query id="Aggregate-SRs" frequency="1h" recovery-periods="2">
  <entity-path>/probegroup/index[name='rollup_fragment']/fragment</entity-path>
</query>
```

The following entry creates a query that runs every hour and returns every hour's web services latency metrics for all probe groups starting from April 22 at 3pm:

```
<query id="Web-Services" frequency="1h" recovery-periods="2">
  <entity-path>/probegroup/probe/index[name='services']/service</entity-path>
  <init-query-time>2009-04-22T15:00:00</init-query-time>
</query>
```

7. Optionally, you can use the **servers-per-query** attribute on a query as a way to reduce the load on the server.

servers-per-query= specifies processing the export query in batches.

```
<query id="Probes" frequency="1h" recovery-periods="2" servers-per-query="10">
  <entity-path>/probegroup/probe</entity-path>
</query>
```

For example, if there are 30 mediator servers and you set servers-per-query=10, then the export gets results for 10 mediators, exports these results and then processes the next 10 mediators and so on.

This attribute should only be set when using more than 10 mediators.

8. Optionally, each query can have an include or exclude filter applied to the query specified in the **<entity-path>** element. The include filter elements must be specified first before the exclude filter elements.

For either filter, specify:

- name= A regular expression to match against the metric name to filter. A value of "" matches all metrics.
- category= A regular expression to match against the category name to filter. A value of "" matches all categories.
- <order>: For multiple include or exclude filters, the order in which to process the filters.

For example, the following entry returns metrics for Database metrics only:

```
<query id="Probes" frequency="5m" recovery-periods="2">
```

```
<entity-path>/probegroup/probe</entity-path>
<metric-include-filter order="1" name="" category="Database" />
<metric-exclude-filter order="1" category="" />
</query>
```

The following example returns all metrics for EJBs excluding EJB-Poll metrics.

```
<query id="EJBStats" frequency="5m" recovery-periods="2">
  <entity-path>/probegroup/probe</entity-path>
  <metric-include-filter order="1" name="" category="EJB" />
  <metric-exclude-filter order="1" name="EJB-Pool" />
</query>
```

For more information about regular expressions, see ["Regular Expressions" on page 153](#).

- Optionally, specify the data retention rules for the extracted data by specifying the **<purge>** element. This prevents the database from running out of storage.

For example, the following entry causes data that is over 24 hours old (`retention="1d"`) to be deleted from the target database. The purge operation is initiated every hour (`frequency="1h"`) and any needed purge operations use 1hr increments (`purgeInterval="1h"`) to purge the data, thus reducing overall load on the system:

```
<purge id="Default.Client.Purger" frequency="1h" retention="1d"
  purgeInterval="1h"/>
```

- Save the changes to the **data-export-config.xml** file.

Task 5: Modify the data export configuration file for trend data (1 minute data granularity)

The queries that export the Diagnostics data are defined in **<diag_server_install_dir>/etc/data-export-config.xml** file of the Diagnostic Commander Server. You can configure both summary and trend data queries at the same time in the `data-export-config.xml` file. For details on configuring summary data queries, see ["Task 4: Modify the data export configuration file for summary data" on page 163](#).

Caution: Exporting trend data is intended for specific, individual metrics and should be kept to a minimum as it can add significant overhead to mediators and databases. When planning what trend data to export, refer to ["Notes and Limitations" on page 168](#).

Follow these steps to set up this file:

- Make a backup copy of the **<diag_server_install_dir>/etc/data-export-config.xml** file if desired.
- Open the **<diag_server_install_dir>/etc/data-export-config.xml** file for editing.
- Locate the **<enabled>** element and set it to true:

```
<enabled>>true</enabled>
```

This element is used to turn on or off the data export operation. You should disable the data export operation when it is not needed to avoid unnecessary system overhead. By default the data export operation is disabled.

4. Locate the **<customer name>** element and set it to the customer name:

Unless you are a SaaS customer, the customer name should always be Default Client.

```
<customer name='Default Client'>
```

5. Locate the **<db-target>** element and enter the driver name, connection URL, user name and password (encrypted or plain text) for the target database.

For example, for SQL Server with an encrypted password:

```
<db-target>
  <driver>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver>
  <connection-
url>jdbc:sqlserver://testapps.mycompany.com:1433;databaseName=DIAG</
connection-url>
  <user>sa</user>
  <encrypted-password>0BF:1ym51y0s1uo71z0f1unr1y0y1ym9</encrypted-password>
  <batchsize>200</batchsize>
</db-target>
```

For example, for Oracle with an unencrypted password:

```
<db-target>
  <driver>oracle.jdbc.driver.OracleDriver</driver>
  <connection-url>jdbc:oracle:thin:@testapps.mycompany.com:1521:ORCL</connection-
url>
  <user>diagfan</user>
  <password>tiger2</password>
  <connection-property name="oracle.jdbc.defaultNChar" value="true"/>
</db-target>
```

For Oracle databases, the **oracle.jdbc.defaultNChar** property must be set to true when UTF8/UTF15 character support is required.

To encrypt a database password, use the Diagnostics password encryptor. See "Password Obfuscation" in the Diagnostics Collector Guide.

For the **<batchsize>** element, specify the batch size in units used for optimal JDBC PreparedStatement execution. By default, this is set to 100. Large implementations with large payloads require adjustments to the default.

6. For each set of metrics that you want to export, specify the following in the **<trendquery>**:
 - id= A name which identifies the query being defined. Must be unique to this data-export-config.xml file. Spaces are not allowed in the id value.
 - frequency= A string value that specifies how often to run the query. This is limited to 5m only.
 - recovery-periods= specifies how far back in time to start querying after an outage occurs.

- <entity-path> One of the entity path groupings described in Task 2.
- <metric-selection name> The name of the specific metric you want to export.
- <metric-selection type> The data type of the specific metric you want to export.

For example, the following entry creates a query that runs every 5 minutes and returns 1m data points latency for all fragments. If an outage occurs, only recover back 5 minutes from current time:

```
<trendquery id="fragmetrics" frequency="5m" recovery-periods="1">
  <entity-path>/probegroup/probe/fragment</entity-path>
  <metric-selection name="latency" type="average"/>
</trendquery>
```

The following entry creates a query that runs every 5 minutes and returns selected probe metrics for all probe groups:

```
<trendquery id="Probemetrics" frequency="5m" recovery-periods="0">
  <entity-path>/probegroup/probe</entity-path>
  <metric-selection name="ProcessCpuUtil" type="average"/>
  <metric-selection name="HeapUsedPct" type="average"/>
  <metric-selection name="latency" type="timeout_count"/>
  <metric-selection name="latency" type="exception_count"/>
  <metric-selection name="Threads Current Count" type="average"/>
</trendquery>
```

7. Optionally, you can use the **servers-per-query** attribute on a query as a way to reduce the load on the server.

servers-per-query= specifies processing the export query in batches.

```
<query id="Probes" frequency="1h" recovery-periods="2" servers-per-query="10">
  <entity-path>/probegroup/probe</entity-path>
</query>
```

For example, if there are 30 mediator servers and you set servers-per-query=10, then the export gets results for 10 mediators, exports these results and then processes the next 10 mediators and so on.

This attribute should only be set when using more than 10 mediators.

8. Optionally, specify the data retention rules for the extracted data by specifying the **<purge>** element. This prevents the database from running out of storage.

For example, the following entry causes data that is over 24 hours old (retention="1d") to be deleted from the target database. The purge operation is initiated every hour (frequency="1h") and any needed purge operations use 1hr increments (purgeInterval="1h") to purge the data, thus reducing overall load on the system:

```
<purge id="Default.Client.Purger" frequency="1h" retention="1d"
  purgeInterval="1h"/>
```

9. Save the changes to the **data-export-config.xml** file.

Notes and Limitations

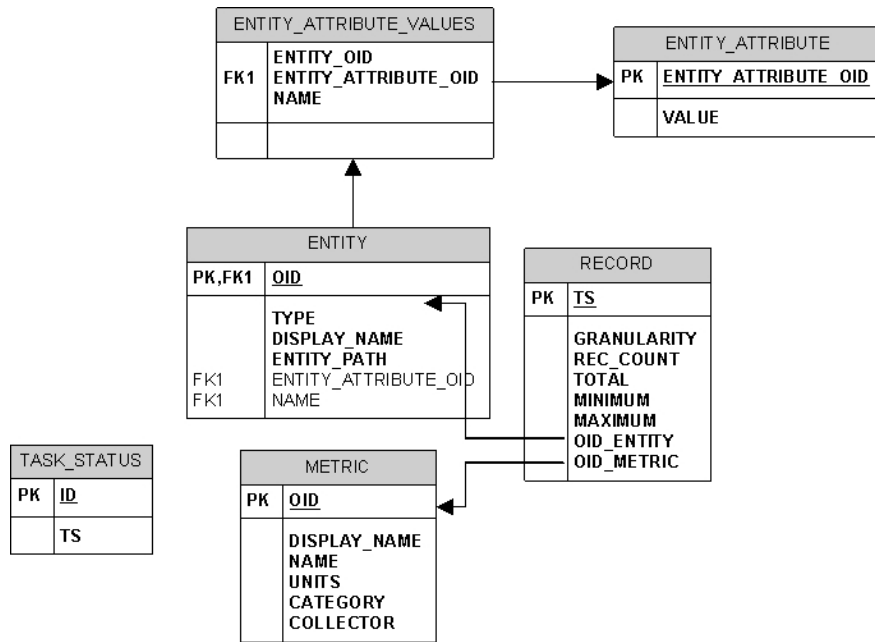
- Exporting trend data using the <dataquery> tag enables you to export fine granularity of data, which it can do at 1m intervals (by default and not changeable). Exporting trend data is not intended for entire database exports and is intentionally constrained to be used only on specific identified individual metrics, for which more granularity is required.
- Exporting trend data as opposed to summary data places a much higher overhead on the mediator and database and must be taken into consideration, especially if your mediator is already handling significant data load from probes. The mediator will basically be doing very time consuming queries every five minutes.
- The mediator requires a connection to a database that can handle the insert load that such granular metric exporting generates. For example, if you have 5000 server requests for which you want one metric exported for each one, the number of rows that are inserted in the database every 5 minutes for this one trend export task is $1m * 5 * 5,000 = 25,000$. There are also associated entity definitions for each of these server requests which, although they are only one time inserts, can still add significant overhead. For example, in the above scenario, there are 6 additional ENTITY_ATTRIBUTE rows for each server request which means another 30,000 ($6 * 5,000$) must be inserted.
- Exporting trend data does not support minimum and maximum metrics and the value in these columns is always 0.

Task 6: Monitor the data export operation

Assuming that the corresponding commanding Diagnostics Server is started, the entries in the saved configuration file take effect immediately. You do not need to restart the commanding Diagnostics Server.

The **data-export-config.xml** is parsed and verified as follows:

1. Each [db-target] specified in the XML is verified by connecting and verifying the database tables are available. If they are not available they are created with the following data relationships:



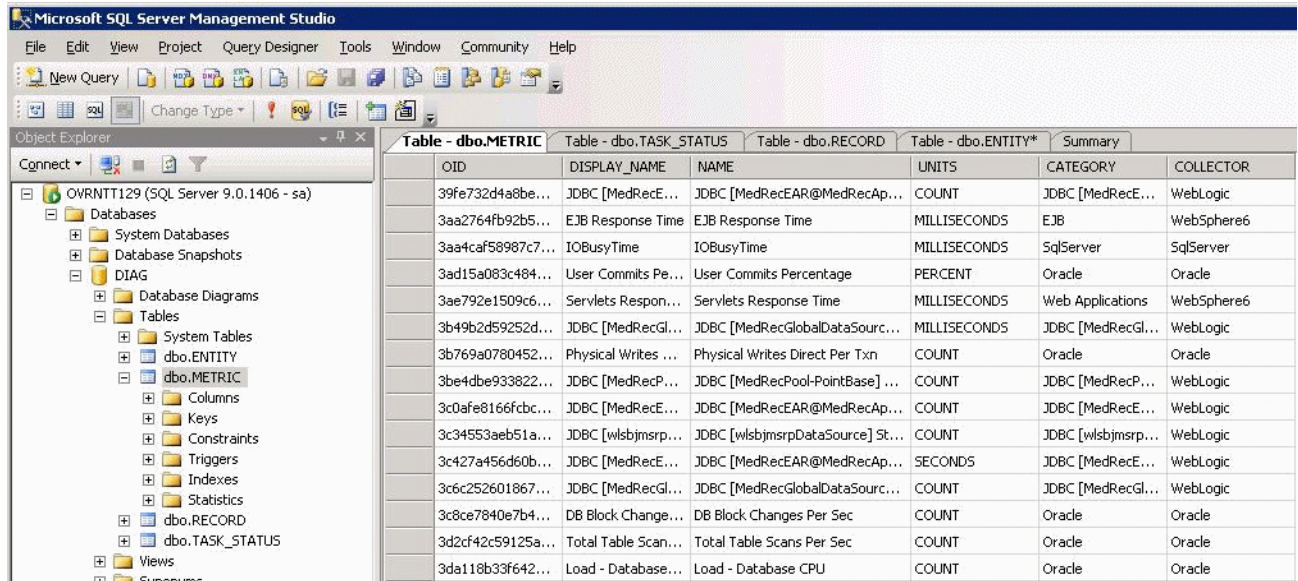
2. The Diagnostics Server schedules when to run each query based on the <frequency> specified. For example, a daily report (frequency = 1d) is run once a day after the last hour of the day has been aggregated into that daily summary. This means, in particular, that the queries are automatically aligned at the existing granularity boundaries.

- When the scheduled query executes, the results of the query are stored in the database tables as follows:
 - Entity descriptions, such as probe, fragment, or host are stored in the ENTITY table and the unique key is a MD5 hash of the TSDB entity key and the distributed source value to make it unique within a federated environment.
 - Metrics descriptions are stored in the METRIC table and the unique key is a MD5 hash of the metric data name, metric data collector name and the distributed source to make it unique within a federated environment.
 - The metric values are stored in the RECORD table and have the foreign key values pointing to the ENTITY and METRIC table unique keys. This is done to reduce overall size of data storage as the descriptions of both entities and metrics would be duplicated on every RECORD table row.
 - The 2 additional tables, ENTITY_ATTRIBUTE_VALUES and ENTITY_ATTRIBUTE, serve as lookup tables to further describe the ENTITY dimension.

Task 7: Verify the results

You can use the database tools of your target database to verify the expected results. For example, the following image shows the results stored in the METRIC table of a SQL Server database. This set of metrics is returned based on the query statement shown:

```
<query id="Probes" frequency="1h" recovery-periods="2">
  <entity-path>/probegroup/probe</entity-path>
</query>
```



Task 8: Select the data from the target database

Once the exported data is stored in the target database, you can query it as desired. However, a pivot manipulation is required to get the data into a useful reporting format. The pivot uses the foreign key

references to combine the dimension table data into a flatten row with the description data joined to the fact table.

Sample SQL scripts, queries and reports are included in `<diag_server_install_dir>/contrib/dataexport/` as follows:

- **sql_server_sample_view.sql.** SQL Server views used to denormalize and pivot exported data into a more friendly reporting format.
- **oracle_server_sample_view.sql.** Oracle DB Server views used to denormalize and pivot exported data into a more friendly reporting format.
- **oracle_view_query_samples.sql.** Various examples of querying the Oracle views.
- **sql_server_reports directory.** A directory containing SQL Server Reports project with various sample reports. Using the SQL Server Reporting tool, open the `sql_server_reports` directory and the "Diagnostic Fragments.sln" file.

Sample Queries

This section includes some query examples for dealing with time, querying totals and querying averages.

Dealing with Time

To query for data from 8 am to 5 pm, your query should specify the start time as 8:00 and the end time as 16:59. If you specify 17:00 as your query's end time, you will include data from the next time bucket which could be 17:00 to 18:00. This is reflected in the examples that follow.

Querying for Totals

Use the `sum()` function to calculate totals of metrics.

Example:

This query will calculate the total number of threshold violations between 6pm and 8pm for the Server Request with the entity path: Default Client / Default / ROS54770TST_Diag80_JDK_15.

```
select entity_display_name as Server_Request, sum(total) as Avg_Latency,sum(total) as
Tot_Latency, sum(rec_count) as count, metric_name, units, name, entity_path
from DIAG.DBO.REG_FRAGMENT_TYPE_METRICS_VIEW
where metric_name = 'threshold_violations'
and ts between '2009-06-11 18:00:00.000' and '2009-06-11 19:59:00.000'
and entity_path = 'Default Client / Default / ROS54770TST_Diag80_JDK_15 / '
group by entity_path, entity_display_name , metric_name,units, name
order by entity_path, entity_display_name
```

Querying for Averages

To calculate the average metric value, divide the total metric value by its count. The `rec_count` field will contain the count.

Note: The count for the Soap Fault metric is set to its total and hence it is not possible to calculate an average value for this metric. Only a total calculation is possible for this metric. In Diagnostics version 7.5, this was also true for the Threshold Violations metric. From Diagnostics 8.0 onwards, the count is

available for Threshold Violation and hence it is possible to calculate an average for this metric.

Example:

This query will calculate the average latency between 6pm and 8pm for the Server Request with the entity path: Default Client / Default / ROS54770TST_Diag80_JDK_15.

```
select entity_display_name as Server_Request, sum(total)/sum(rec_count) as Avg_
Latency,sum(total) as Tot_Latency, sum(rec_count) as count, metric_name, units, name,
entity_path
from DIAG.DBO.REG_FRAGMENT_TYPE_METRICS_VIEW
where metric_name = 'latency'
    and ts between '2009-06-11 18:00:00.000' and '2009-06-11 19:59:00.000'
    and entity_path = 'Default Client / Default / ROS54770TST_Diag80_JDK_15 / '
group by entity_path, entity_display_name , metric_name,units, name
order by entity_path, entity_display_name
```

Example:

This query will calculate the average number of threshold violations between 6pm and 8pm for the Server Request with the entity path: Default Client / Default / ROS54770TST_Diag80_JDK_15.

```
select entity_display_name as Server_Request, sum(total)/sum(rec_count) as Avg_
Latency,sum(total) as Tot_Latency, sum(rec_count) as count, metric_name, units, name,
entity_path
from DIAG.DBO.REG_FRAGMENT_TYPE_METRICS_VIEW
where metric_name = 'threshold_violations'
    and ts between '2009-06-11 18:00:00.000' and '2009-06-11 19:59:00.000'
    and entity_path = 'Default Client / Default / ROS54770TST_Diag80_JDK_15 / '
group by entity_path, entity_display_name , metric_name,units, name
order by entity_path, entity_display_name
```


Uninstallation

Uninstall on Windows

This section contains instructions for uninstalling the Diagnostics Server on Windows.

Steps to uninstall

1. Uninstall IAPA components (if installed as part of Diagnostics Server installation).
2. Uninstall OM agent (if installed as part of Diagnostics Server installation).
3. Uninstall Diagnostics.

To uninstall the IAPA component

The OM agent and IAPA components are not uninstalled when you uninstall the Diagnostics Server. If you want to uninstall the OM agent and IAPA components they must be uninstalled before you uninstall the server because the uninstaller for these components is under the server directory. The components must be uninstalled in this order: first the IAPA component and then the OM agent.

1. For Windows systems, change directory to **<diag_server_install_dir>/server/setup/ovo-iapa/win64**.
2. From the command line in the win64 directory execute:

```
cscript.exe <install_dir>\server\bin\install_ovo_iapa.vbs /x HPOprIAPA-09.00.111-Win5.2_64-release.msi uninstall.log
```

To uninstall the OM Agent

1. For Windows systems, change directory to **<diag_server_install_dir>/server/setup/ovo-agent/win64**.
2. From the command line in this directory execute:

```
cscript.exe oainstall.vbs -r -a
```

To uninstall the Diagnostics Server

1. Select **Start > All Programs > Micro Focus Diagnostics Server > Uninstall Micro Focus Diagnostics Server**.

Note: Make sure the Tomcat service is up and running before you start to uninstall Diagnostics.

2. Rename the server install directory with the suffix **_OLD**.

Alternatively, you can run the following command, and then rename the server directory:

```
<diag_server_install_dir>\Uninstall\HPEDiagserver\setup.exe
```

Uninstall on Linux

This section contains instructions for uninstalling the Diagnostics Server on Linux.

Steps to uninstall

1. Uninstall IAPA components (if installed as part of Diagnostics Server installation)
2. Uninstall OM agent (if installed as part of Diagnostics Server installation)
3. Uninstall Diagnostics

To uninstall the IAPA component

The OM agent and IAPA components are not uninstalled when you uninstall the Diagnostics Server. If you want to uninstall the OM agent and IAPA components they must be uninstalled before you uninstall the server because the uninstaller for these components is under the server directory. The components must be uninstalled in this order: first the IAPA component and then the OM agent.

1. Change the directory to **<diag_server_install_dir>/server/setup/ovo-iapa/Linux64**.
2. As root user, from the command line in the Linux 64 directory execute:

```
rpm -e HPOprIAPA
```

To uninstall the OM Agent

1. Change the directory to **<diag_server_install_dir>/server/setup/ovo-agent/Linux64**.
2. Execute the following command as a root user

```
./oainstall.sh -r -a
```

To uninstall the Diagnostics Server

You can uninstall the Diagnostics Server in console mode or graphical mode.

1. Stop the Diagnostics Server. For instructions, see ["Start and Stop the Diagnostics Server" on page 37](#).
2. Change the directory to the root directory.
3. Enter the following at the UNIX command prompt:

In console mode:

```
<diag_server_install_dir>\Uninstall\HPEDiagserver\setup.sh
```

In graphical mode:

Export your display before running in graphical mode.

```
export DISPLAY=<hostname>.0.0  
<diag_server_install_dir>/Server/_uninst/uninstaller.bin
```

4. Rename the server install directory with the suffix **_OLD**.

Troubleshoot

Diagnostics Installers Do Not Work on Linux

If the Server or Collector installation program on a Linux host terminates abruptly, make sure that the appropriate X libraries are installed on the host. See "[Library Packages Required on Linux](#)" on page 179.

Install the libraries and run the installation program again.

Java Agent Fails to Operate Properly

If the Java Agent does not operate properly, check whether the **ClassLoader.class** file located in the folder `<agent_install_dir>\classes\boot\java\lang\` was created during the installation process.

If the file was not created, make sure you have instrumented the JRE as this is what creates it. See "Preparing Application Servers for Monitoring with the Java Agent" in the Java Agent Guide.

Error During WAS Startup with Diagnostics Profiler for Java

Symptoms:

Class Loader errors occur when starting WAS with the Diagnostics Profiler for Java.

Reason:

Additional classes need to be excluded from the instrumentation.

Solution:

1. Open the property file, `<agent_install_dir>\etc\inst.properties`
2. Update the **classes.to.exclude** property to exclude `!com\.ibm\.*` by appending the class to the end of the existing values.

```
classes.to.exclude=!iaik\.security\.*,!c8e\.*,!org\.jboss\.net\.protocol\.file\  
.Handler,!org\.jboss\.net\.protocol\.file\.URLConnection,!*ByCGLIB.*,  
!com\.ibm\.*
```

Missing Server-Side Transactions

Symptoms:

The server requests for each probe are displayed in Diagnostics but the BPM transactions that are associated with the server requests are not displayed.

There are two symptoms to look for in the **server.log** file:

"not dropping at least one transaction that timed out" – this indicates that a transaction has not received any data for a period of time (10m by default) and has not received the ELT. This warning is issued

infrequently, and only when the transaction times out. After this warning you should see the transaction data in the UI. For more information on ELT see ["Reduce the Diagnostics Server Memory Usage" on page 70](#).

"Late data received for time period that was already persisted. Adjusting data by..." – this indicates that the server received an ELT unreasonably late, but before the transaction timed out. The data will be reported, but not at the same time that APM or SaaS reported it.

Reason:

If you do not see either of the log messages listed above and there is no transaction data the most likely cause is the BPM is not running the scripts.

Solution:

1. Verify that Business Process Monitor is running in BSM/APM or Software-as-a-Service (SaaS) and that the monitor is running.
2. Verify the state of the profile in the Business Process Monitor Console.

Event Capture Buffer Full Warning

Symptoms:

Some Diagnostics data loss is occurring and the following error appears in the probe log file:

```
"The event capture buffer is full, at least one event dropped."
```

Reason:

The log entry indicates that the application load is too high, or that the application is excessively instrumented.

Solution:

In some cases, increasing the value of the **event_buffer.size** property in the `etc/capture.properties` file can help avoid dropping events, but often reducing the application instrumentation is necessary.

WebSphere Application Server Startup Issue

Symptoms:

With the Java probe enabled, the WebSphere application server throws exceptions such as `"java.lang.NoClassDefFoundError: javax.xml.rpc.handler.Handler"` during application startup.

Reason:

The SOAP Handler cannot be loaded in this configuration.

Solution:

Turn off the SOAP Handler (impacts SOAP consumer ID and payload capture) by changing the property below to false in the probes's **etc\inst.properties** file. Then restart the application server.

```
details.conditional.properties=\  
mercury.enable.SOAPHandler=false  
mercury.enable.autoLoadSOAPHandler= false, \  

```

OR

Add the missing classes to the application server's class path so they can be accessed by the probe's SOAP Handler.

For example from `<WebSphere>/lib/j2ee.jar` there are several jars that contain the missing classes.

Java Agent Support Collector

The `runSupportSnapshot` utility creates a .zip file containing the entire set of files relevant to troubleshooting one or more instances of the Java agent in a Diagnostics deployment environment.

The .zip file contains the following:

- Files from the `<Diagnostics_agent_install_dir>\etc` directory
- Files from the `<Diagnostics_agent_install_dir>\log` directory
- Probe instance information, including property settings, environment variables, stack dumps, and class loader information.

To run `runSupportSnapshot`:

1. Navigate to `<Diagnostics_agent_install_dir>\contrib\JASMUtilities\Snapins`.

Note: The utility is also available in the `<agent_install_dir>\bin` directory.

2. Execute `.\runSupportSnapshot.cmd -console` on Windows, or `./runSupportSnapshot.sh -console` on UNIX or Linux.
3. A .zip file is created. The default location of the saved zip file is the `.../DiagnosticsAgent/` folder.

File Permissions on Linux

When you start Diagnostics on Linux using the nanny (recommended), by default new files created by the Diagnostics process are world-writable files. You can configure the nanny to use the `umask` command to change the default permissions assigned to new files that are created by the Diagnostics process.

To use the `umask` command to set file permissions:

Edit the `/opt/MercuryDiagnostics/Server/nanny/linux/dat/mdrv.dat` file.

In the nanny section, edit the line `ExtCmdLine=-nanny_mask 000` and change the permissions setting (by default, 000) with the required mask. For example:

```
ExtCmdLine=-nanny_mask 022
```

This sets the file permissions so that only the owner (the user who ran the nanny) has write permissions to newly created files.

Note: If you start Diagnostics manually not using the nanny (for example, for troubleshooting purposes), any `umask` setting you may have previously set to configure permissions in Linux is used.

Appendix A: Library Packages Required on Linux

Diagnostics Servers and Collectors on Linux machines require certain library packages to run. Installing a Server or Collector by running the installation program in graphical mode on Linux requires additional library packages.

RedHat Enterprise Linux, 64-bit

Note: Be sure to use the 32-bit version of these packages even when installing on a 64-bit platform. The installation program runs as a 32-bit application even when installing a 64-bit version of the Diagnostics component.

Package	How to Check If it is Installed
glibc.i686	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'glibc\.(i686 i386)'</code>
compat-libstdc++-33.i686	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'compat-libstdc++-33\.(i686 i386)'</code>
libstdc++.i686	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'libstdc++\.(i686 i386)'</code>
libgcc.i686	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'glibc\.(i686 i386)'</code>
libXp.i686 ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'libXp\.(i686 i386)'</code>
libXtst.i686 ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'libXtst\.(i686 i386)'</code>
libXrender.i686 ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'libXrender\.(i686 i386)'</code>
libXmu.i686 ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep -E 'libXmu\.(i686 i386)'</code>
libXp-1.0.0-15.1 ²	<code>rpm -qa grep libXp-1.0.0-15.1</code>

¹ Only required for running the Server or Collector installation program in graphical mode.

² Only required for Red Hat Enterprise Linux versions 6.x, 7.x.

If the **rpm** command to check if a library package is installed returns the library name, this indicates that the library is installed. To install a missing library, run the following command:

```
yum install <package_name>
```

Novell SUSE Linux Version 10, 64-bit

Note: Be sure to use the 32-bit version of these packages even when installing on a 64-bit platform. The installation program runs as a 32-bit application even when installing a 64-bit version of the Diagnostics component.

Package	How to Check If it is Installed
glibc-32bit	<code>rpm -qa --qf '\${NAME}\n' grep glibc-32-bit</code>
libstdc++33-32bit	<code>rpm -qa --qf '\${NAME}\n' grep libstdc++33-32-bit</code>
libstdc++-32bit	<code>rpm -qa --qf '\${NAME}\n' grep libstdc++-32-bit</code>
libgcc	<code>rpm -qa --qf '\${NAME}\n' grep libgcc</code>
xorg-x11-libs-32bit ¹	<code>rpm -qa --qf '\${NAME}\n' grep xorg-x11-libs-32-bit</code>
¹ Only required for running the Server or Collector installation program in graphical mode.	

If the **rpm** command to check if a library package is installed returns the library name, this indicates that the library is installed. To install a missing library, enter the following command:

```
yast2 -i <package_name>
```

Novell SUSE Linux Version 11, 64-bit

Note: Be sure to use the 32-bit version of these packages even when installing on a 64-bit platform. The installation program runs as a 32-bit application even when installing a 64-bit version of the Diagnostics component.

Package	How to Check If it is Installed
glibc-32bit	<code>rpm -qa --qf '\${NAME}\n' grep glibc-32-bit</code>
libstdc++33-32bit	<code>rpm -qa --qf '\${NAME}\n' grep libstdc++33-32-bit</code>
libstdc++43-32bit	<code>rpm -qa --qf '\${NAME}\n' grep libstdc++43-32-bit</code>
libgcc43-32bit	<code>rpm -qa --qf '\${NAME}\n' grep libgcc43-32-bit</code>
xorg-x11-libs-32bit ¹	<code>rpm -qa --qf '\${NAME}\n' grep xorg-x11-libs-32-bit</code>
xorg-x11-libXp-32bit ¹	<code>rpm -qa --qf '\${NAME}\n' grep xorg-x11-libXp-32-bit</code>
xorg-x11-libXrender-32bit ¹	<code>rpm -qa --qf '\${NAME}\n' grep xorg-x11-libXrender-32-bit</code>
xorg-x11-libXmu-32bit ¹	<code>rpm -qa --qf '\${NAME}\n' grep xorg-x11-libXmu-32-bit</code>
¹ Only required for running the Server or Collector installation program in graphical mode.	

If the **rpm** command to check if a library package is installed returns the library name, this indicates that the library is installed. To install a missing library, enter the following command:

```
yast2 -i <package_name>
```


Novell SUSE Linux Version 12, 64-bit

Note: Be sure to use the 32-bit version of these packages even when installing on a 64-bit platform. The installation program runs as a 32-bit application even when installing a 64-bit version of the Diagnostics component.

Package	How to Check If it is Installed
glibc-32bit	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep glibc-32bit</code>
libstdc++33-32bit	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep libstdc++33-32bit</code>
libstdc++43-32bit	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep libstdc++43-32bit</code>
libgcc43-32bit	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep libgcc43-32bit</code>
xorg-x11-libs-32bit ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep xorg-x11-libs-32bit</code>
xorg-x11-libXp-32bit ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep xorg-x11-libXp-32bit</code>
xorg-x11-libXrender-32bit ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep xorg-x11-libXrender-32bit</code>
xorg-x11-libXmu-32bit ¹	<code>rpm -qa --qf '%{NAME}.%{ARCH}\n' grep xorg-x11-libXmu-32bit</code>

¹ Only required for running the Server or Collector installation program in graphical mode.

If the **rpm** command to check if a library package is installed returns the library name, this indicates that the library is installed. To install a missing library, enter the following command:

```
yast2 -i <package_name>
```

Appendix B: Manual Installation of OM Agent and IAPA Components

The installer for the Diagnostics commander server includes the option to install the OM agent and IAPA components used for sending Health Indicator status events to BSM/APM. You can choose to skip installing these components during the Diagnostics Server installation and instead install the components manually on the Diagnostics commander server at a later time as described below.

To manually install OM agent on Windows systems:

1. For Windows systems, change directory to **<diag_server_install_dir>/server/setup/ovo-agent/win64**.
2. From the command line in this directory execute:

```
cscript.exe oainstall.vbs -i -a -minprecheck
```

To manually install OM agent on Linux systems:

1. For Linux systems, change directory to **<diag_server_install_dir>/server/setup/ovo-agent/Linux64**.
2. As root user, from the command line in this directory execute:

```
./oainstall.sh -i -a -minprecheck
```

To manually install the IAPA component on Windows systems:

1. For Windows systems, change directory to **<diag_server_install_dir>/server/setup/ovo-iapa/win64**.
2. From the command line in the win64 directory execute:

```
cscript.exe <install_dir>/server/bin/install_ovo_iapa.vbs /i  
HPOprIAPA-09.00.111-Win5.2_64-release.msi <log file>
```

Where <log file> is a file where the results of the install are logged, path is optional.

To manually install the IAPA component on Linux systems:

1. For Linux systems, change directory to **<diag_server_install_dir>/server/setup/ovo-iapa/Linux64**.
2. As root user, from the command line in the Linux64 directory execute:

```
rpm -ivh HPOprIAPA-09.00.111-Linux2.6_64-release.rpm
```

To complete the OM agent installation

To complete the OM agent configuration you must perform the steps to register Diagnostics with BSM/APM. See the APM-Diagnostics Integration Guide for details including troubleshooting information.

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 Server Installation and Administration 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!