



HPE Diagnostics

Software Version: 9.26

Release Notes

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE 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

Confidential computer software. Valid license from HPE 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 - 2016 Hewlett Packard Enterprise Development LP

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 in the Documentation directory on the product installation media.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<https://softwaresupport.hpe.com/group/softwaresupport/search-result?keyword=>.

This site requires an HPE Passport account. If you do not have one, click the **Create an account** button on the HPE Passport Sign in page.

Support

Visit the HPE Software Support website at: <https://softwaresupport.hpe.com>

This website provides contact information and details about the products, services, and support that HPE Software offers.

HPE Software Support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HPE Passport user and sign in. Many also require a support contract. To register for an HPE Passport ID, go to <https://softwaresupport.hpe.com> and click **Register**.

To find more information about access levels, go to:

<https://softwaresupport.hpe.com/web/softwaresupport/access-levels>

HPE Software Integrations and Solutions

Visit the Integrations and Solutions Catalog at <https://softwaresupport.hpe.com/km/KM01702731> to explore how the products in the HPE Software catalog work together, exchange information, and solve business needs.

Contents

Installation Notes	6
Diagnostics Components Compatibility	6
What's New in 9.26 IP2	7
Automatic Probe Mediator Limit	7
Expanded Support	7
Security Fix	7
What's New in 9.26 IP1	8
.NET Agent support for Web API	8
What's New in 9.26	9
What's New in 9.24 IP1	11
What's New in 9.24	13
What's New in 9.23 IP2	15
What's New in 9.23 IP1	16
What's New in 9.23	17
New Features in 9.23	18
What's New in 9.21	22
New Features in 9.21	22
What's New in 9.20	26
New Features in 9.20	26
System Requirements	37
Notes and Limitations	38
Java Agent	38
.NET Agent	41
Integrations (BSM, PC, LR, TV, SaaS)	42
Integration With APM 9.30	44
Collectors (Oracle, SAP, MQ, TIBCO EMS, SQL Server, webMethods, VMware)	44
Java Profiler	45
.NET Profiler	45
Diagnostics Server	46
User Interface	47
Documentation	48
Translated Versions	49

Localization and Globalization	50
Send Documentation Feedback	51

Installation Notes

Diagnostics Server, Agent and Collector installation requirements and detailed instructions are provided in the Diagnostics Server Installation and Administration Guide, Collector Guide, and Agent Guides.

Diagnostics 9.26 contains a full replacement of the Diagnostics components but note that there are special upgrade installation instructions you must follow. Please refer to the upgrade and patch installation instructions provided in the HP Diagnostics Server Installation and Administration Guide. The upgrade installation instructions are also provided as `Diagnostics_Upgrade_Patch_Install_Instructions.pdf` in the download .zip file or in the Documentation directory on the installation DVD as a convenience.

Diagnostics Components Compatibility

The Diagnostics components in your deployment have the following requirements:

- All Diagnostics Servers in the deployment must be at the same version.
- 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 HP 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.

Check the HP Software Download Center (<http://support.openview.hp.com/downloads.jsp>) for any updates and patches to the Diagnostics components. This web site requires an HP Passport login. Locate the Diagnostics information and choose the appropriate link for downloading the Diagnostics software.

The 9.26 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 9.1x, 9.2x
- .NET Agent 9.1x, 9.2x
- Python Agent 9.2x
- Collectors 9.1x, 9.2x

For information about Diagnostic Server compatibility with BSM, see the BSM System Requirements and Support Matrices guide.

What's New in 9.26 IP2

Automatic Probe Mediator Limit

Added a new property named "*force.auto.assign*" in the probe */etc/dispatcher.properties* file. Possible values as follows:

- **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.
- **False** – the system does not check whether the mediator is over the load limit.

Added the missing property "*mediator.load.query.time.frame*" in the *server.properties* file with a value of "1d".

Expanded Support

We have added support for the following:

- IBM WebSphere 8.5 Liberty profile
- .net 4.5.2
- Jetty 7

Security Fix

This IP contains a fix for several security issues that have been resolved.

What's New in 9.26 IP1

Nested Server Requests in Java Agent

Added support for monitoring nested server requests in Java. This feature allows users to define entry points in the application code to unconditionally start new server request instances.

.NET Agent Instrumented Location Throughput Throttling

Ignoring events which are over the throughput threshold causes Diagnostics to provide incorrect data, particularly counts and throughputs, without any way to identify which data is correct.

To resolve this issue, there is now an alert if the probe has been throttling events (red status). This solution was implemented in the past for the Java Probe, and has now been implemented for the .NET agent.

The probe processes only the first N events coming from a single location within each second.

.NET Agent support for Web API

.NET Agent now supports Web API in both IIS and non-IIS hosted environments.

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET framework.

Improved OnDemand Call Profiles

The OnDemand Call Profiles feature enables users to collect additional instance trees for call profile analysis. This is done by creating OnDemand Call Profiles Collections rules for the investigated service request.

The OnDemand Call Profile feature was improved to collect cross VM call profiles, if they exist, in addition to regular call profiles. Having many instances of cross VM call profiles that relate to the investigated server request can significantly improve a user's ability to troubleshoot problems.

What's New in 9.26

Improved Mediator Performance and Scaling

The maximum number of supported, concurrent user sessions has been significantly improved. The total amount of concurrent users sessions has been increased to 200.

A new, large size configuration has been tested and documented. For details, refer to the Diagnostics System Requirements and Support Matrixes Guide.

.NET Agent

Overhead and Instrumentation Control

A new feature enables you to view instrumentation points by layers and to dynamically enable/disable them, either individually or for an entire layer. For details, see "Managing Probe Overhead" in the .NET Agent guide.

Asynchronous HTTP Requests

Asynchronous HTTP requests and received HTTP responses are captured and correlated. Cross-VM instances visualize the async transaction in its entirety.

Assembly Instrumentation Exclusion

Added new ability to exclude specific assemblies (for example, assemblies related to sensitive processing such as encryption).

More Control on .Net Stack Trace Capture

Agent setting disables Stack Trace Sampling mechanism by default. Stack Trace Sampling can be enabled dynamically by updating the Agent configuration. An additional Stack Trace Sampling setting provides tuning options to control the amount of resources consumed by the mechanism.

Java Agent

New Server Request Classification

HTTP parameters captured by the Java Agent can be transformed before being reported by the probe - this is important for applications such as Guidewire. For details, see "Configuring URI and Parameter Capture" in the Java Agent guide.

Added Support for Frameworks

- Hessian API
- Apache CXF
- OpenJPA
- Guidewire

Java 8 Features

- Support for Java 8 Lambda expressions, including IBM and Oracle Java 8.
- New system metric - Java 8 MemoryPool Metaspace Usage.

Integration with AppPulse Mobile

HP AppPulse Mobile monitors real users experience for native mobile applications.

You can now integrate on-premise Diagnostics and SaaS-based AppPulse Mobile for an end-to-end mobile monitoring solution. AppPulse Mobile enables you to investigate problematic mobile requests which were served by instrumented application servers and to drill down to the relevant Server Requests in Diagnostics.

Platform Support Changes

Added Server Platforms

- Oracle Linux (OEL) 7
- SUSE Linux Enterprise Server 12

Systems That can be Monitored by Diagnostics

- WebMethods 9.x
- Collector support for WebMethods Broker 9.x

Enhancements and Defect Fixes

Diagnostics 9.26 includes over 20 customer encountered defect fixes and over 10 customer requested enhancements.

What's New in 9.24 IP1

Ease of Deployment

- Java: Auto-deployment is now supported for Tomcat as a Windows service and for WebSphere. For details, see the Java Agent Guide.
- .NET: Assorted improvements have been made to the .NET Agent installation.

Instrumentation

- Support for instrumentation of Java 8 lambda expressions.
- SQL Bind Variables capture. A new option enables you to configure the Java and .NET agents so that Bind Variables values in SQL queries are captured and displayed in the Call Profile view.

Method Data	
Arguments	IM patients t0 WHERE (t0.ssn = ? AND t0.status = ?)
Class	com.pointbase.net.netJDBCPreparedStatement
Default Name	java.sql.ResultSet com.pointbase.net.netJDBCPrepare
Layer	Database/JDBC/Execute
Type	JDBC
Name	netJDBCPreparedStatement.executeQuery()
Package	com.pointbase.net
Return Type	java.sql.ResultSet
SQL Parameters	?1=123456789, ?2=APPROVED
Call Argument	SELECT t0.id, t0.version, t0.email, t0.password, t0.us

The feature is disabled by default, as the values may be sensitive. For more information on how to enable this feature, see "Capturing SQL Parameters" in the Java Agent guide and "<sqlparsing> element" in the .NET Agent guide.

Enhanced Manageability

Enhanced Log viewer for Diagnostics Server log files through the admin web interface. The log messages can now be viewed in a tail-like manner, with additional filters and highlighting options, making Diagnostics system troubleshooting easier and quicker, especially for environments with large log files.

Long Requests Metric

As a result of customer requests, the Long Requests metric for Java probes has been re-introduced. Customers can use this metric to determine if there are long running transactions which may be stuck.

Platform Support Changes

Added support for the following platforms:

- Red Hat Enterprise Linux (RHEL) version 7
- OpenJDK versions 1.7.x, 1.8.x
- Wildfly 8
- Collector support for Tibco 8.x

Security

By default, you cannot use SSL for incoming HTTPS connections due to the POODLE vulnerability problem. Instead, use TLS. For details of the POODLE vulnerability problem, see <https://www.us-cert.gov/ncas/alerts/TA14-290A>. For further details, see "Configuring the Diagnostics Server for Incoming HTTPS Connections" in the HP Diagnostics Server Installation and Administration Guide.

What's New in 9.24

Java Instrumentation Management

- Switch between monitoring profiles from the UI, without the need for an application server restart. For example, turn on detailed monitoring when a problem occurs and return to high level monitoring when the problem ends.
- Instrumentation Analytics. The agent automatically disables high frequency instrumentation points for automatic overhead optimization and control. This is useful for protecting production application performance in situations where methods are called with high frequency. The agent automatically detects these situations, reduces data collection, and limits the overhead.
- Dynamically stop and start agent data collection from the UI for testing and overhead control purposes.

Manageability for Large Scale Environments

- Improved mediator performance and scaling. HP Diagnostics scales horizontally to support thousands of agents on a single deployment. The maximum capacity of a single mediator has been improved and more than doubled compared to previous versions. The server start up time for ultra large environments has been significantly reduced. A new large sizing configuration has been tested and documented (for details, refer to the HP Diagnostics System Requirements and Support Matrixes Guide).
- Rule Based mediator assignment. Increase flexibility of probe assignments in large, complex environments. Introduced the concept of Mediator groups for optimized load balancing based on business needs. Agents can be assigned based on environment type (such as "Testing", "Production"), geographical location (such as "East", "West"), and other relevant meta-data. For details, see the "Probe Registration Auto-Assignment for Large Deployments" section in the HP Diagnostics Server Installation and Administration Guide.
- Simplify the work of administrators with automatic application generation and assignment based on agent configuration. For details, see the "Automatically Assigning a Probe to an Application" chapter in the HP Diagnostics Java Agent Guide.
- Introduced the Mediator Load metric in the Diagnostics System Capacity View. This metric is used by the automatic load balancing mechanism and can also be used by Diagnostics administrators to better understand mediator load.

Granular Data Export

Support for more granular data for custom integrations and dashboards. Selected metrics can now be exported at 1 minute granularity.

Platform Support Changes

- Diagnostics version 9.24 does not include server packages for Solaris or 32 bit operating systems. If you need to install Diagnostic servers on such systems, use Diagnostics version 9.23.
- FireFox ESR 31 (EUI).
- Microsoft Internet Explorer 11 (EUI).
- Java Agent – monitoring for JRE 8 applications.
- Support for JSF framework instrumentation.
- Java Agent – monitoring for GlassFish 4.x application server.
- .Net Agent – collects Microsoft SharePoint disk I/O metrics.
- .Net Agent version 9.24 does not include a Win32 package. Use the .Net Agent version 9.23 Win32 package to monitor Windows 32 bit .Net applications.
- .Net Agent support for MVC 3, 4, and 5, including support for Entity Framework (ORM).
- Collector support for Microsoft SQL 2014, Oracle 12c, VMware 5.1, 5.5.

Server Robustness

- Embedded OM Agent updated to version 11.13 to enable IPv6 integration with HP Business Service Management.
- Upgraded server 3rd party software (Oracle Berkeley DB, Oracle JRE) for improved robustness and security. For upgrade details, refer to the HP Diagnostics Upgrade and Patch Installation guide.
- Many defects have been fixed to improve server robustness, security and ease of deployment.

Compatibility Changes

- Due to client JRE limitations with older versions of BSM, only BSM versions 9.23 and later are supported.
- Supports integration with LoadRunner/Performance Center version 12.01.

What's New in 9.23 IP2

Security Fixes

Provides an update that fixes the Heartbleed vulnerability issue associated with **OpenSSL**. For details, see <http://heartbleed.com> and the [HP Security Bulletin](#).

System Requirements Changes

Supports JRE version 8 for the Diagnostics Enterprise UI client.

Compatibility Changes

- Supports LoadRunner version 12.00
- Supports Performance Center version 12.00

RUM-Diagnostics Integration

Improvements have been made to the RUM-Diagnostics integration, using the .NET Agent to inject the RUM JavaScript. For details, refer to the RUM-Diagnostics Integration Guide available on the HP Software Support Online Product Manuals site.

What's New in 9.23 IP1

Platform Support Changes

- Added support for Microsoft Windows Server 2012 R2 64-bit
- Added support for Oracle Linux (OEL) 6.x

Documentation Changes

The following new PDF guides are available on the HP Software Support Online Product Manuals site:

- Diagnostics System Requirements Guide
- RUM-Diagnostics Integration Guide

RUM-Diagnostics Integration

Improvements have been made to the RUM-Diagnostics integration. For details, refer to the RUM-Diagnostics Integration Guide available on the HP Software Support Online Product Manuals site.

What's New in 9.23

Platform Support Changes

- **Subsequent releases of Diagnostics after version 9.2x will NO LONGER support Servers and Collectors on Solaris operating systems. After Diagnostics 9.2x, Servers and Collectors will only be supported on Windows and Linux operating systems.**
- Added support for Microsoft Windows Server 2012 Enterprise Edition (HP Operations Manager Event and Metrics Integration is not supported in Diagnostics running on Windows Server 2012)
- Added support for Mozilla Firefox 24.0 ESR
- Added support for Microsoft Internet Explorer (IE) 10.0
- Dropped support for Java Agents monitoring of Java 1.4 applications
Version 9.23 Java agents no longer support monitoring of applications running on version 1.4 JVMs. If you require monitoring of a Java 1.4 JVM, use the 9.21 Java agent.

Documentation Changes

The HP Diagnostics Installation and Configuration guide has been divided up into separate manuals, as follows:

- Server Installation and Administration Guide
- Collector Guide
- Java Agent Guide
- NET Agent Guide
- Python Guide
- BSM-Diagnostics Integration Guide
- LoadRunner/Performance Center-Diagnostics Integration Guide

This will allow readers to more easily find the information related to the installation and setup of a specific workflow or component. The guides are packaged with their respective components or available as downloads from HP Software Support Online. See Documentation Updates in the front matter of this guide for more information about downloading documentation.

Defect Fixes

Diagnostics 9.23 has a large number of defect fixes.

For details on fixed defects see the patch notes on the HP support site where you downloaded the patch:
<http://support.openview.hp.com/selfsolve/patches/>

New Features in 9.23

The 9.23 release includes the following new features.

- Java Agent Monitoring Profiles

Background: Diagnostics is a versatile and flexible product which can be used in different environments, ranging from development desktops to systems deployed in production. However, customization of all Java Agent configuration settings to match the expected scalability or overhead level vs. the amount of collected data can be a tedious task, especially for novice users.

Description: The Java Agent configuration is now managed by a monitoring profile. Monitoring Profiles offer a predefined set of settings that control the instrumentation points, metrics collection, and other properties for the probe from a single location. Monitoring profiles are a safe and easy way to manage the impact of the probe on the monitored system and still obtain the needed performance data.

See the new chapter "Monitoring Profiles" in the Diagnostics Java Agent Guide for more information.

Benefits: Usability

- Java Agent Auto-deployment on JBoss and Tomcat

Background: To enable monitoring of an application by the Java Agent, users perform the following: Instrument the JRE used by the application server (Diagnostics' JRE instrumentation does not modify the installed JRE, but rather places copies of instrumented classes under the Java Agent installation directory); and then modify the application server startup script to specify the JVM parameters to invoke the agent and use the instrumented JRE when the application starts. This procedure varies for each type of application server.

Description: The Java Agent Setup program can now perform the instrumentation and modifications to the application server startup script automatically for agents running on JBoss and Tomcat application servers. Users need only to restart the application server after the Agent Setup program completes to pick up the changes to the script.

See the chapter "Installing Java Agents" in the Diagnostics Java Agent Guide for more information.

Benefits: Usability

- Java Agents in Cloud Environments

Background: Previous versions of Diagnostics were difficult to use when deployed in a cloud environment, such as Stackato. There are two major sources of the difficulties:

- probe configuration challenges: manual probe id configuration hardly works in the cloud, hostnames do not exist, and the IP address of the box is not resolved by the Java runtime.
- dynamic assignment of IP addresses and Database URIs makes it not only difficult for the user to find the relevant data, but also causes unlimited growth of the Diagnostics symbol table.

Description: Java Agents can now monitor applications running in Stackato or HP aPaaS.

See the new section "Java Agents in Cloud Environments" in the Diagnostics Java Agent Guide for more information.

Benefits: Functionality, Usability

- Mediator Server Auto-Assignment for Large Deployments

Background: In a large deployment environment with many probes, deployments often share a Java Agent installation with multiple probe instances since they all use similar configuration. In this kind of setup, the probe.id and registrar.url is set and overridden in the startup JVM options. This allows multiple probes to share the same configuration, but still have unique probe.id and mediators they are reporting their data too.

Description: Probe auto-assignment is now available for these types of environments. Users do not have to assign each individual probe to a specific mediator. When enabled, the probe will always ask the commander for the registrar.url when it starts instead of relying on a the version specified in **dispatcher.properties**.

See the section "Configure Monitoring of Multiple Java Processes on an Application Server" in the Diagnostics Java Agent Guide for more information.

Benefits: Functionality, Usability

- .NET Agent Call Stack Sampling

Background: In previous versions, a user viewing the Call Profile view may see that sometimes the instrumented methods have long execution times, while they are sufficiently quick in most cases. Adding additional instrumentation for the callees of such a method is often not desirable, because the instrumentation overhead affects all invocations, not only those with long running times. Such overhead would be especially high for callees invoked many times during each invocation of a caller, i.e. in a loop.

Furthermore, adding instrumentation may require multiple steps (instrumenting callees of the first level callees, etc.), application re-starts, and sometimes may not be even possible, for example when the calls are made using reflection.

Description: Asynchronous thread stack trace sampling for .NET applications can now be used to discover what the application is doing when exceptionally long executing times for a method are observed. It is a collection technique in which some threads are periodically looked at and their state is recorded.

The main advantage of thread stack trace sampling is its ease of use, because no application restart is required, sampling can be turned on and off dynamically and there's no additional overhead while sampling is turned off.

See the new section "Asynchronous Thread Stack Trace Sampling for .NET Applications" in the Diagnostics Users Guide and "Configuring Thread Stack Trace Sampling" in the Diagnostics .NET Agent Guide for more information.

This feature is also available for the Java Agent.

Benefits: Functionality, Usability

- .NET Agent Thread Diagnostics

Description: The .NET Agent now offers Thread Diagnostics on its monitored applications. The Thread Diagnostics can be turned on or off on demand to diagnose issues as they occur.

The Threads window displays thread performance metrics for the threads that are running in a .NET probed application and provides a way for you to capture stack traces for the running threads. There is also a thread state analyzer that displays approximate thread state distribution percentage for each thread.

This page can be useful for helping to diagnose the following situations:

- Incorrect thread pooling or attempting to do too much in a single thread.
- Performance problems caused by deadlocks or concurrency-related issues.
- Problems that go deep into the interactions with the OS kernel where you need to see the CPU time broken into user and kernel times.

See the new section "Threads Windows Description" in the "Diagnostics Profiler for .NET" chapter in the Diagnostics .NET Agent Guide for more information.

This feature is intended to provide the metrics to feed features like "Thread State Analyzer," "Thread Dumps," and "Thread Monitoring," which are currently available in the Java Agent.

Benefits: Functionality, Usability

- .NET SharePoint Monitoring and New Views in the Enterprise UI

Background: In previous versions, a user would need to use the Probes view to view information about the .NET Agent monitoring a SharePoint application.

Description: Automatic discovery and recognition of SharePoint probes and assigned application server as "Microsoft SharePoint Server" consolidated into the following new views:

- Summary: High level dashboard view of SharePoint hosts and probes
- SharePoint Probes: List of all SharePoint Probes and related metrics
- SharePoint SQL: List of sorted (by latency) SQL calls executed within SharePoint environment

See the new section "Microsoft SharePoint Views" in the Diagnostics Users Guide for more information.

Benefits: Functionality, Usability

- .NET Parameter Capture

Description: You can now configure and aggregate HTTP requests on specified query parameters. This is useful where the latency of Server Requests may depend on the query parameters.

See the new section "<httpcaptureparams>" in the Diagnostics .NET Agent Guide for more information.

Benefits: Functionality, Usability

- .NET Automatic URI Collapsing

Background: In previous versions, each unique server requests was identified and reported in the Call Profile view. In some applications this resulted in the Diagnostics Server being flooded with a large number of unique server requests.

Description: The .NET Automatic URI Collapsing feature uses mathematical rules to collapse branches of frequently used URI paths common in REST environments which have resource (ids) at the end of the paths.

- Replaces frequently used URI branches with *
- On by default. Limits defined by simple numbers separated by '/' representing path limits n URI paths are retained over application restarts by persistence of applied rules and observed URI instances
- "Original URI" property available on "collapsed" Server Request call profile views

See the new section "<uriautocollapsing>" in the Diagnostics .NET Agent Guide for more information.

Benefits: Functionality, Usability

- MyBSM Diagnostics Dashboard views now contain CIs of type WebService and WebService Operations
Description: In the MyBSM Diagnostics Dashboard, it is now possible to select CIs of types Web Service or Web Service Operation in the View Explore and see new views for those CIs.

Benefits: Functionality, Usability

- VM Heap Utilization % Metric

Description: The Java Probes view shows a new metric related to heap utilization: **VM Heap Utilization %**. This metric reports the percentage of heap used by the selected application being monitored by this probe. The heap is measured immediately after the last GC cycle, therefore heap growth to be controlled by the GC cycle does not affect the reported percentage.

This metric is only available for probes running on an IBM or Hotspot JVM. For other JVMs, see the VM Heap Used metrics in the Details pane.

Benefits: Functionality, Usability

What's New in 9.21

Platform Support Changes

- **Subsequent releases of Diagnostics after version 9.2x will NO LONGER support Servers and Collectors on Solaris operating systems. After Diagnostics 9.2x, Servers and Collectors will only be supported on Windows and Linux operating systems.**

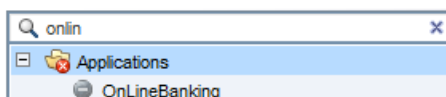
Diagnostics 9.21 has a large number of defect fixes and new features, details on the new features are provided below.

For details on fixed defects see the patch notes on the HP support site where you downloaded the patch:
<http://support.openview.hp.com/selfsolve/patches/>

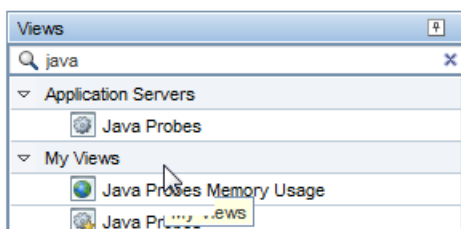
New Features in 9.21

The 9.21 release includes the following new features.

- A new Diagnostics Python Agent and a new Python Probes view in the Enterprise UI
Description: A new Diagnostics Python Agent is available in 9.21. The Python agent captures method invocations, server requests and system metrics from Python applications. Out of the box instrumentation is included for OpenStack Cloud Computer Platform, Django web framework and WSGI. Custom instrumentation can be added. The Python agent is installed on the same machine as the Python application under test. See the Python Agent Guide for details on Installing and Configuring the Python Agent.
Benefits: Functionality
- User can search for an application name/group in the welcome page & for views in sidebar
Background: In prior versions, it was very difficult to find applications when a large number of applications exists, it's very, very difficult to find the application of interest.
Description: The user can now enter a few characters in a box above the list of applications and the list of applications will be immediately filtered.



In addition, it is also now possible to search for a particular view. For example, if you would like to find all views with the word "java", just enter the word "java" in the Views search box.

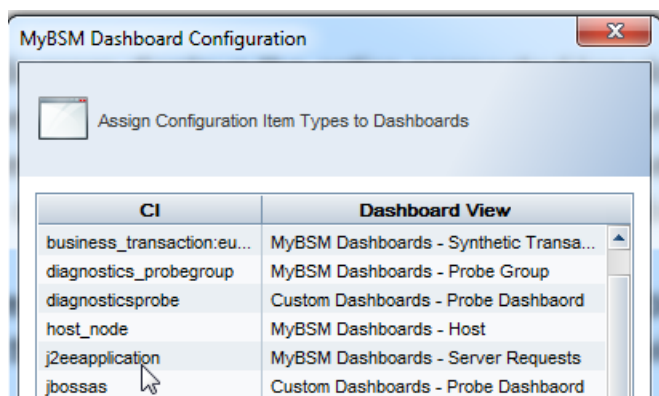


Benefits: Functionality, Usability

- User can display a custom dashboard within MyBSM based on CI type.

Background: In 9.20, a new feature was introduced that allows the creation of very rich and sophisticated custom dashboards. However, these dashboards could not be assigned to CIs to be displayed in BSM.

Description: In “MyBSM Dashboard Configuration”, it is now possible to assigned a CI to a custom dashboard.



In MyBSM, when a Diagnostics CI is selected (for example, jbossas), then the dashboard associated with jbossas (in the example above, it is the “Probe Dashboard” view inside the “Custom Dashboards” view group) is shown in its full glory.

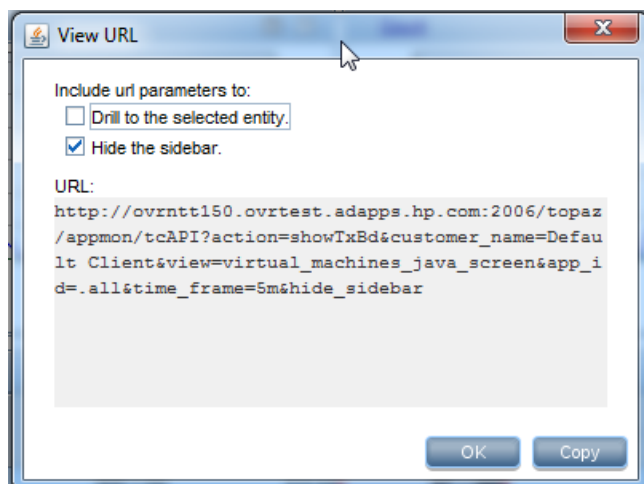
Benefits: Functionality, Usability

- Provide a menu item that will show the URL for the active view to make it easier to open to a view.

Background: In past releases, it was possible to jump directly to a screen if the right URL was created. However, the creation of the URL was manual and error prone. An example URL might have been:

`http://server:2006/topaz/appmon/tcAPI?action=showTxBd&customer_name=Default Client&view=virtual_machines_java_screen`

Description: A new menu has been added “Show URL for View” .



There are two options:

Drill to the selected entity – If you want the selected entity to be highlighted

Hide the side bar – To create a true dashboard effect

The Copy button copies the URL to the clipboard. The URL can then be used in portals like the BSM UI Mashup or any portal or webpage of your choosing. The new URL will work in the application context as well (which was not previously supported).

Benefits: Functionality, Usability

- Diagnostics can show internal metrics and Web Services for webMethods Integration Server.

Background: Prior to 9.21, Diagnostics did not display web services and metrics specific to webMethods Integration Server environments.

Description: With 9.21, a full range of metrics is collected and displayed for webMethods Integration Server environments, similar to WebLogic or WebSphere. In addition, select Inbound and Outbound Web Services are captured and displayed in the UI along with Consumer IDs, Soap Faults, and Soap Payloads.

Benefits: Functionality

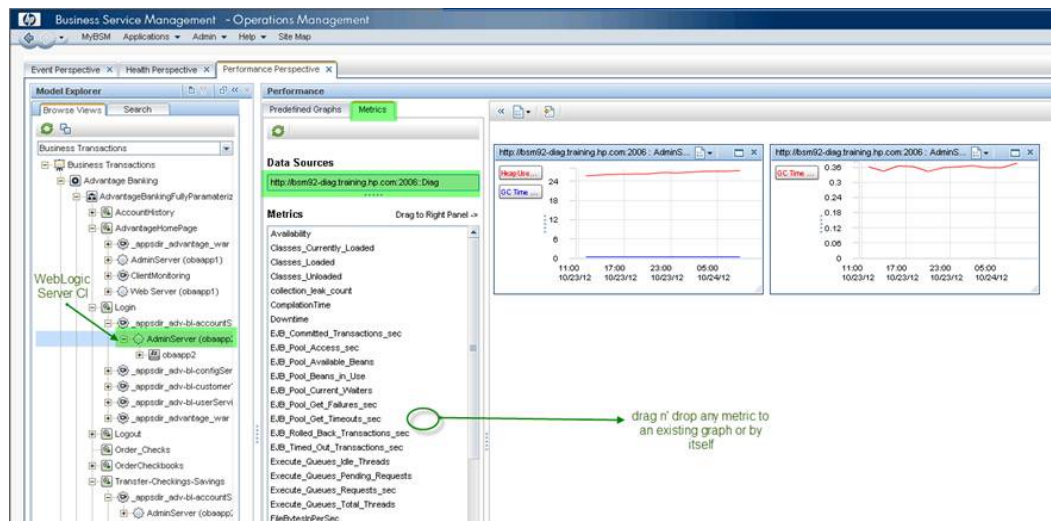
- Improved integration with BSM's Performance Grapher (also known as PMi).

Background: Starting in BSM 9.20, PMi/BSM Grapher is now decoupled from OMi and does not require an OMi Foundation license (unless you want to graph OM agent data).

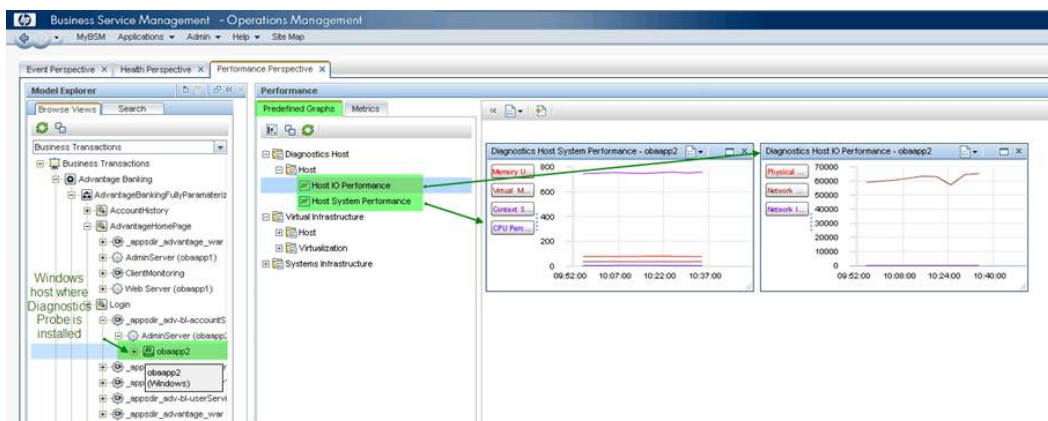
Description: The BSM Grapher appears in the BSM UI under the "Operations Management" application (Performance Perspective) and has a related BSM "Mash-Up" component. The BSM Grapher can graph OM Agent data, SPI data, OM Agent-RTM data, EUM data (BPM and RUM), SiteScope data and now Diagnostics data.

All that is needed is to attach the Diagnostics Commander to BSM 9.20. The rest is automatic. The BSM Grapher looks at the "Monitored By" list attribute of a CI to determine what data sources are available.

Example of Diagnostics Probe metrics from an application server:



Example of a predefined graph (host level shown):



Benefits: Functionality

- Client Monitoring changes

Background: The name of the "ClientMonitoring.war" files has been changed to "HPDiagCM.war".

Description: If you are using Client Monitoring from a previous version you should undeploy the old "ClientMonitoring.war" file and change the web server configurations to forward requests to the new context as described in the HP Diagnostics Java Agent Guide chapter on Client Monitoring.

Benefits: Usability

- Diagnostics tracks licenses by operating system instances now instead of application instances

Background: Prior to 9.21, Diagnostics tracked and reported on application instances in the licensing report.

Description: With 9.21, Diagnostics tracks and reports on operating system instances. The same license is used but now the license value that was entered previously represents the total number of OS instances running an agent (not a collector). For example, if previous to 9.21 the license count was 100 instances, then Diagnostics would report a license violation after 100 application probes or .NET processes was exceeded (the number of operating systems they were running on was not relevant). Starting with 9.21, Diagnostics will report a license violation only after it detects agents (not collectors) running on 100 different operating system instances.

Benefits: Licensing

What's New in 9.20

Platform Support Changes

- **Subsequent releases of Diagnostics after version 9.20 will NO LONGER support Servers and Collectors on Solaris operating systems. After Diagnostics 9.20, Servers and Collectors will only be supported on Windows and Linux operating systems.**

Diagnostics 9.20 has a large number of defect fixes and new features, details are provided below.

New Features in 9.20

The 9.20 release includes the following new features.


- Some Diagnostics views have improved default graphs plus new default views.

Background: Diagnostics has many out of the box default views. An effort is made with each release to improve the default views.

Description: The following views have improved graphs:

- Application Servers – Java Probes
- Application Servers – Application Explorer – Resource Utilization Metrics tab
- Java Probe Summary (drill down from Java Probe)
- .NET Probe Summary (drill down from .NET Probe)
- New view for JVM's based on Sun/Oracle – Click on any application server based on a Sun/Oracle JVM, right click and choose “Open in Custom View” and choose “JVM Health (Oracle)” to see a view with some interesting metrics unique to Sun/Oracle JVM's.

Benefits: Functionality, Usability

- Views are easier to navigate in new tree structure and numerous overall UI improvements.
 - The Outlook style left pane has been replaced with a hierarchical tree structure. The tree is made up of a number of view groups such as “Applications”, “Application Servers”, “Infrastructure”, etc. Some of the view groups have sub groups such as “Database → Oracle”. The bottom leaf of a tree is a view. This new approach makes it significantly easier to navigate to the many views within Diagnostics. Many of the views are restructured into new view groups, so it will take a little while to familiarize yourself with the new structure. Note that the view group nodes in the tree (for example “Infrastructure”) do not have any action or view associated with them, they are simple a way to structure the views.
- Actions have been consolidated into one menu . Go here for all possible actions, including Help and Configuration. In addition to this consolidated action menu, you can right mouse on a view group or view to get a context sensitive list of actions for that view.
- Graphing features have been enhanced and consolidated to the upper right of each graph. The list of options are:



These icons map to (from left to right):


- Layout charts in two columns
- Chart trends with least number of graphs
- Chart trends by metric
- Chart trends by entity
- Line charts
- Area charts
- Stacked Area charts
- Maximize/restore
- Toggle auto-hide

- Menu action items associated with the selected entity can be performed by right clicking on the graph OR on the selected entity.

- The chart column is now combined color and chart.

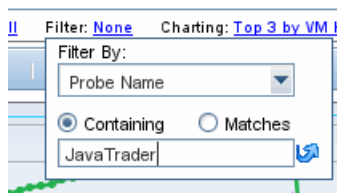
- New ICONS   for:

- Select the columns to display in the table
- Search the table

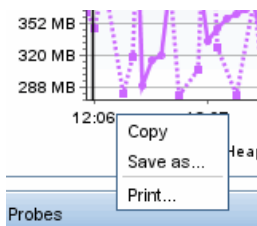
- Details Pane - New Search metrics box to find metrics . Type text to filter on metric names.

- The technology behind all the metric and topology graphs has been replaced. All the metrics and topology graphs have a different look and feel and a different color scheme.

- The Filter in a view now has the ability to Match a regular expression:



- You can right click on the time range of a graph bring up a menu to copy, save and print



For more information on using the Diagnostics UI, see “Common Controls in the Diagnostics UI” in Chapter 2 of the User's Guide.

Benefits: Usability, Functionality

- Search for Entities

Background: Prior versions of Diagnostics lacked any way of searching entities.

Description: A new search box has been added and is available from any view.



Simply type a string and either hit enter or click on the icon to execute the search. By default, the server will index the following entities into the search engine:

- Hosts
- Probes
- Business Transactions
- Synthetic Transactions
- Applications

Server Request indexing is disabled by default. If you wish to enable this on all of your servers, you need to set:

- `searchindexer.include.server_requests=true`

in:

- `server.properties`

on the commander and every mediator.

The search is enabled by default in 9.20 (except for Server Requests) and it only takes a few minutes to index the entire server. If you suspect that the index needs to be rebuilt (this should not happen), it can be manually rebuilt in seconds via:

- `http://<servername>:2006/search`.

By clicking on .

For more information, see “Common Controls in the Diagnostics UI” in the User Guide.

Benefits: Functionality, Usability

- Client Monitoring

Background: Diagnostics now provides client (browser) side instrumentation to measure the exact time a request takes from the client to the server.

Description: New in 9.20 is the ability to capture total, server and render time of a “page” as experienced by the user (browser). This feature will show the “true” user experience (includes download of components and render time). After you deploy the ClientMonitoring.war file to your application server and update your webserver configuration to provide external access, the client monitoring feature automatically enables itself within the Java Agent and no further configuration or user interaction is needed. After the JSP's are instrumented, performance data can be displayed in the UI in the following way:

- By URL
- By Browser Type
- By Location

Diagnostics has the ability to automatically instrument JSP's for select technologies. These include:

- Tomcat
- JBoss
- WebLogic
- WebSphere

For more details on this new feature, see “Client Monitoring Views” in the User Guide.

Benefits: Functionality

- Dashboard Creator

Background: In prior Diagnostics versions, it was possible to create a “Dashboard” or custom view by creating a new view and dragging and dropping completed views (either stock or custom) into the different sections. This capability was limited and did not allow complex, robust dashboard views to be created easily.

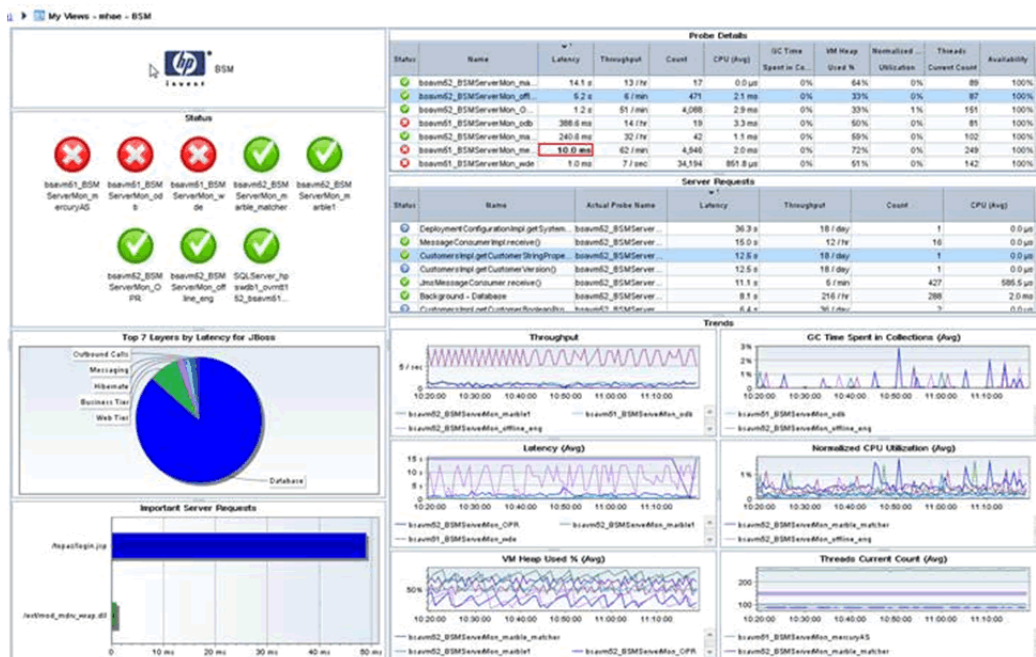
Description: In 9.20, there is a Dashboard View Creator feature which has a large number of new features. The dashboard can include any number of sections, any size or rectangular shape, each including one of the following choices:

- Time Series Chart as Line charts, Area charts or Stacked Area charts
- Summary Chart with Pie or Bar charts
- Table
- Status
- Image
- Probe topology

For each section or feature (except for the probe topology), the user is allowed to choose from the following entities:

- Application
- Business Transaction
- Host
- Layer
- Outbound Call
- Probe
- Server Request
- SQL
- Synthetic Transaction

See “Working with Custom Views and Dashboards” in the User Guide for more details on how to get started with this new feature. Here is an example of what can be created:



Benefits: Functionality, Usability

- Open in Custom View

Background: In prior Diagnostics versions, it was possible to select an entity such as a Java Probe and right mouse to select “Open in Custom View”. However, this feature was very limited before the addition of the Dashboard Creator feature.

Description: The best way to explain this feature is by showing an example. In this example, a custom dashboard was created that looks like this for Java probes:



The graph has the following components:

1. Top Left – Time Series chart with Throughput and Latency (Avg) for the application server
2. Top Right - A bar chart showing the top 7 server requests by Latency (Avg)
3. Bottom Left – This is an advanced component that has 3 tabs:
 - A tab with a table of SQL statements for the application server (not shown)

- A tab with a topology graph of the application server (shown)
- A tab with pie chart of the layers in the application server showing Latency (not shown)

4. Middle Right – A table of all the Server Requests for the probe

5. Bottom Right – Status of the probe

When the template is created, it's easiest to see what it would look like if you filter on a specific probe when you create each graph, but this is not a strict requirement. Since it is just a template, it does not matter. Do whatever it takes to make the template look as desired. Once the template is in place, to use the template, go to the Java Probe view and select any probe, right click and "Open in Custom View" and open it with the view you created. The result will look like this:



Get creative in using the Dashboard Creator and Open in Custom View features!

Benefits: Usability

- Critical and Warning Thresholds/Alerts and Normal Alerts

Background: Prior versions of Diagnostics allowed a single critical threshold.

Description: With this release, the user can set a Critical and a Warning threshold. Each threshold set via the Set Threshold dialog box presents the user with two choices now instead of one:

☒ Use custom threshold

☒ Critical > 10.00 Seconds

☐ Warning > 1.00 Seconds

Likewise when the user creates an alert via the dialogue box, new options are presented:

Alert Triggers

☒ Alert when entity status changes to critical
☐ Alert when entity status changes to warning
☒ Alert when entity status changes to normal
☐ Alert when no entity data has been received for 5 minutes

See "Working with Thresholds and Metrics in the Details Pane" and "Working with Alerts and Rules" in the User Guide for more details.

Benefits: Functionality

- Script Execution on Alert Trigger

Background: Prior versions of Diagnostics did not offer any script execution options when an alert was triggered.

Description: This version of Diagnostics allows the user to execute a custom script when an alert is triggered. There is a new option when creating an alert rule for a metric:

Optional Actions

☒ Execute the following scripts (comma-separated list) when the alert is triggered:

For more details on this feature, see “Working with Alerts and Rules” in the User Guide.


Benefits: Functionality

- PDF Export

Background: It is often desirable for the Diagnostics administrator to export information from the profiler and/or the Enterprise UI (EUI) to a format that can then be sent to an application developer or some other subject matter expert (SME). Previously, the only export format choices were html and CSV. These formats are not ideal for communicating the information to the developer or SME.

Description: With 9.20, a new PDF export format is available. This format is excellent for e-mailing detailed information to a developer or SME. In addition, an option is presented to append to an existing PDF file. So a Diagnostics administrator can append a large number of different screens into an existing file to e-mail to the developer or SME.

This feature is available in every view in the Enterprise UI through the tools menu .

The feature is even more powerful in the Java Profiler where you can export a single tab of data or all tabs of data in one step. Look for the Acrobat ICON in the action menu .

This feature is documented in “Common Control in the Diagnostics UI” in the User Guide.

Benefits: Functionality, Supportability

- New JRE Instrumenter Options

Background: Starting with Diagnostics 9.0, a new feature was added called Collection Leak Pinpointing (CLP). CLP automatically detects leaking collections and provides a stack trace of where the leak occurs. This helps identify issues early, while there is time to mitigate the issue (such as an eventual out of memory error/server crash), as well as saves developers time by avoiding the tedious task of analyzing heap dumps. In order to use this feature, it is required that the JRE instrumenter is run to pre-instrument every JRE that is used by the application (server) that is being monitored (including 1.5 and 1.6). Configuration was difficult for two reasons. The first difficulty was in trying to find the right JRE that the application (server) would use. The second was when the JRE was upgraded and the new JRE had to be instrumented again.

Description: With Diagnostics 9.20, two new methods for instrumenting the JRE have been added to make configuration more consistent. Here are the two new instrumentation methods (listed first) plus the existing two methods:

Automatic Explicit JRE instrumentation - Users explicitly invoke the JRE Instrumenter in an application server startup script (Recommended when possible).

Automatic Implicit JRE instrumentation - Users do not need to explicitly invoke the JRE Instrumenter, the JRE Instrumenter is implicitly invoked by the agent when the agent is used for the first time. (Recommended when Explicit is not possible).

Manual JRE instrumentation – Users need to manually invoke the JRE Instrumenter.

Basic instrumentation (-javaagent only) – The JRE Instrumenter is never invoked, the application server is instrumented but the Collection Leak Pinpointing feature (which needs the JRE instrumenter) is not available and the JVM performance is less than optimal.

Chapter 6 of the Install Manual describes these features in more detail and gives detailed instruction on how to use the new instrumentation methods.

Benefits: Usability

- Java Agent Support for Java 1.7 and Planned Obsolescence of Java 1.4

Background: Prior to 9.20, the Java agent worked with Java 1.4, 1.5 and 1.6.

Description: With 9.20, the Java agent will also work with Java 1.7. In addition, the 9.2x versions will be the last to support Java 1.4. In the next major version of Diagnostics, Java 1.4 will not be supported. However, Java agent versions 9.2x will continue to ship and support Java 1.4 for as long as the customer needs.

- Thread State Analyzer

Background: In previous versions of Diagnostics, the Java Profiler's Thread tab displayed thread performance metrics for the Java threads that are captured by the probe and provided a way to capture stack traces for the captured threads.

Description: New to 9.20 is an exciting new Thread State Analyzer feature. To navigate to Thread State Analyzer, select the Threads tab in the Java profiler. It is not necessary to begin profiling to access the thread data. Thread State Analyzer is available as the third tab within the Thread tool. All three tabs use internally the same data but display different aspects of them. The State Analyzer displays approximate thread state distribution percentage for each thread, over the specified time period. Each thread is represented by a single row.

The left panel provides the thread name. The center panel provides the thread state data. The total height of the colored bar represents 100%. If a thread has been in more than one state during the observation period, multiple colors are used to display the corresponding states, proportionally to the time spent in those states. For automatic updates, the observation period is the same as the configured refresh period.

The right panel displays the current method name with line number, if available. If the stack traces collected for the thread over the observation period are all the same, the method name is displayed using a bold font. If different stack traces were observed, the displayed method is the topmost common method for the collected stack traces, and its display uses a regular font. If no such common method could be found, nothing is displayed.

The following thread states are presented by the Thread State Analyzer:

- Deadlocked - the thread participates in a deadlock cycle
- Blocked - the thread is delayed (suspended) when trying to enter a Java monitor. This can happen when the thread tries to invoke a synchronized method, enter a synchronized block, or re-enter the Java monitor after being awoken from the waiting state, while another thread has not left the Java monitor yet.
- Running - the thread is actively consuming CPU time
- I/O - the thread is performing an I/O operation. It does not use any CPU time. The notion of I/O covers not only the traditional operations on files or sockets, but also covers any multimedia or graphics operations. In general, the thread is waiting for an external (out-of-process) event
- Sleeping - the thread is delayed after invoking the Thread.sleep() method
- Waiting - the thread is delayed, usually having executed Object.wait(). However, threads can get into this state by other means. In general, the thread is waiting for an internal (in-process) event

- Starving - the thread is runnable, i.e. not suspended by any I/O, wait(), sleep() or Java monitor operation, but is not running. This can be caused by insufficient number of CPUs available, Garbage Collection pauses, excessive paging, or by a virtual machine guest OS experiencing a shortage of resources
- Unknown - the Diagnostics Agent was unable to determine the state of the thread

Benefits: Functionality

- Developers Performance Report for their JUnit Tests

Description: Developers can add the Java agent when running their JUnit tests and configure it so that it generates a performance report for all of the unit tests.

This is useful to find out if the performance (latency/CPU) of a particular test has changed over time. When the unit test finishes, the probe creates a CSV file for each server request and a complete listing of all methods that were part of the test case. The CSV file can be opened in a Spreadsheet program to analyze and visualize performance characteristics (the Filter function in Excel is very helpful to select specific methods).

Benefits: Functionality, Performance

- Oracle RAC Support

Background: Prior versions of the Diagnostics Oracle collector did not support Oracle RAC.

Description: New to this release is the ability to connect to Oracle RAC and have the Oracle collector automatically monitor all instances of the cluster. See Chapter 4 of the HP Diagnostics Installation and Configuration Guide for more details.

Benefits: Functionality

- VMware CI and HI Population

Background: Prior to this release, HP Diagnostics did not populate Configuration Items (CI's) or Health Indicators (HI's) with VM Ware information from the VM Ware collector.

Description: If Diagnostics server is registered with BSM and the Diagnostics VMware Collector is installed and configured to collect from a vSphere instance, when the Diagnostics UI "VMware Hosts" and "VMware Guests" views contain instances, the user can view in BSM Apps -> SH -> Top View -> Virtualization_Infrastructure the VMware host instances as VMware ESX Server CI's in which each CI has a composition link to a vmware_hypervisor CI which in turn has execution_environment links to Computer CIs (or Windows or Unix sub-types) corresponding to the VMware guests contained in the VMware host as displayed by the Diagnostics UI.

In addition to the population of CI's, the following subset of the HIs are sent to VMware host and guest CIs:

- CPULoad
- MemoryUsageLevel
- HostDiskUtilization
- InterfaceUtilization

Benefits: Functionality

- New Online Help System

Background: HP Software is migrating to a common look and feel for the online help system. Diagnostics 9.20 has switched to this new help system along with BSM.

Description: You can launch the new online help system from the Tools menu .

Blue tabs on the bottom left of the help window let you switch between Contents, Search and Favorite pane.

Buttons at the top right let you go to next topic/prev topic, print and add to favorites.

Let us know if you have any feedback on the help or other documentation.

Benefits: Consistency with other HP products

- .NET Agent Azure Pack for 9.20

Background: The enhancements of the .NET Azure pack address some of the limitations of the 9.12 Azure pack and updated for changes in the 9.20 Agent.

Description: The following list reflects the new features available through these changes.

The AzurePack will install the metrics agent so that the Host Metrics can be reported to the Diagnostics Server

The AzurePack startup will be changed to accommodate the .NET Agent Http WebServer changes.

For a single host cloud deployment, the user can view the profiler using port 35000.

Benefits: Host metrics and limited profiler support in Azure deployed applications

- .NET Agent WCF Service Support for HTTP and TCP Transport Based Bindings

Background: Previous .NET Agent versions supported limited number of Bindings viz.

1. BasicHTTP
2. WSHTTP
3. netTCP.

We ran into other bindings with customer engagements like CustomBinding, WebHttpBinding, etc which we did not support. We need to expand our WCF binding support as the technology becomes more prevalent and popular.

Description: The WCF support is redesigned around transport type instead of binding type. This enables support for many more standard and custom bindings. The Webservice entity identification has also been modified to make it non-ambiguous over different types of transports and bindings.

Benefits: Wider support for WCF based services and applications.

- .NET Agent FIPS Compliance

Background: Previous .NET Agent versions used Rijndael cryptoservice provider to encrypt the profiler passwords. This algorithm is not FIPS compliant and the library not available on FIPS compliant windows OSes. To support FIPS compliant OSes, we added a default OOB encryption using TripleDESCryptoService Provider which is FIPS compliant.

Description: The default OOB encryption algorithm for profiler password is now the TripleDESCryptoService Provider. We maintain backward compatibility with the following configuration option.

`<profiler authenticate="true" usefipscryptoprovider="true"/>`.

Setting usefipscryptoprovider="false" will revert to pre 9.20 behavior. Also the PassGen utility which encrypts passwords will use the TripleDESCryptoService by default. To use the previous Rijndael encryption use the "\r" option.

Benefits: Acceptance and greater sales opportunities in Federal accounts where FIPS compliance is often mandated.

- .NET Agent Web Server Redesign, Replace TCP Based Webserver with HTTP Based

Background: Previous .NET Agent versions had a webserver which was built on simple TCPIP sockets interface and limited home-grown HTTP processing. This made it less efficient(and stable) and difficult to enhance in terms of new HTTP requests. This redesign will provide easier enhancements and integrations for the much desired eventual .NET profiler UI make-over.

Description: The new HTTP based web-server is more efficient, stable and easier to enhance than the older implementation. In this version there are no significant profiler changes. The one enhancement added is more support related, i.e. now you can access the .NET Probe log and configuration files remotely using the web interface

http://hostname:35000/profiler/logfile - Domain Log file

http://hostname:35000/profiler/ilogfile - Instrumentation log file

http://hostname:35000/profiler/config - probe_config.xml file

Benefits: Efficiency, Stability and Enhancement of the .NET Profiler. Better support tools due to availability of the log and configuration files using the web interface.

- .NET Agent URI Pattern Replacement Capability

Background: Static content served by a website creates a lot of uninteresting Server Requests for a probe. This was more apparent in SharePoint websites which deliver documents and other artifacts. Also as web-design is moving towards javascript based rendering and logic, the prevalence of static/uninteresting Server requests has increased. This also caused over-population in the Diagnostics Server symbol table. To reduce the amount of uninteresting Server Requests and to be able to aggregate certain types of static data, we have introduced the capability of replacing URI patterns with pre-configured and customizable rules.

Description: Added configuration options which will allow URI replacement based on rules. Pattern searching is defined using regular expression syntax. The option in the probe_config.xml file is noted below with the default options

```
<symbols>
```

```
<urireplacepattern enabled="true">
```

```
<pattern value="s#(;/?\\!).*$$$" />
```

```
<pattern value="s#(?<word1>^.*)(/.*\.(js|css|jpg|gif|png|pdf|html|doc|docx)$)#${word1}/Static Content" />
```

```
</urireplacepattern>
```

```
</symbols>
```

Benefits: Fewer uninteresting Server Requests. Better support for Sharepoint deployments due to the ability to rename URIs to consolidate static content and team site requests.

- Diagnostics is integrated with Continuous Delivery Automation

Description: The Continuous Delivery Automation (CDA) product provides a model driven approach to DevOps collaboration, automation of application deployment and monitoring. This enables customers to focus on their core applications and in driving business value while reducing costs, risks and time to perform provisioning and deployment tasks. The core features provided include:

- Application and Infrastructure modeling and Configuration management including version control, role based access, lifecycle management
- Application Deployment management using the tools of customer choice
- Infrastructure Provisioning management
- Embedded monitoring deployment in conjunction with application deployment

Benefits: Functionality, efficiency.

System Requirements

The [Diagnostics System Requirements and Support Matrixes Guide](#) contains information about system requirements for all Diagnostics components. This guide is available as a PDF on the HP Software Support site. Access requires an HP Passport login ([register](#) for an HP Passport).

For details about supported environments, refer to the Diagnostics Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp.

Notes and Limitations

Notes and limitations are provided for the following component areas (click the links below to see details). Note that the defect tracking number shown (for example 35266) is generally prefixed with QCCR1I.

- ["Java Agent" below](#)
- [".NET Agent" on page 41](#)
- ["Integrations \(BSM, PC, LR, TV, SaaS\)" on page 42](#)
- [Collectors \(Oracle, SAP, MQ, TIBCO EMS, SQL Server, webMethods, VMware\)](#)
- ["Java Profiler" on page 45](#)
- [".NET Profiler" on page 45](#)
- ["Diagnostics Server" on page 46](#)
- ["User Interface" on page 47](#)
- ["Documentation" on page 48](#)
- ["Translated Versions" on page 49](#)

Java Agent

- Long Requests metric is no longer available as a Java Probe metric (QCCR1I75560)
The Long Requests metric for Java probes is now available again.
- Unexplained problems with database outbound calls (OC) and SQL statements being attached to a Server Request (SR) for the Java agent (QCCR1I98747)
Note: The result of the fix for this issue is visible only when a fresh Diagnostics server installation is used, or for new SQL statements (not already persisted).
- When the dynamic property enable.stack.trace.sampling is set to "auto" (the default value), it activates stack trace sampling with the limitations described below: (38822) (43937)
 - On any Java 1.5 implementation with the following limitations:
 - For HotSpot JVM, version 1.5.0_04 or later is required
 - For JRockit JVM, JRockit version R27.4 or later is required
 - On any Java 1.6 implementation with the following limitations:
 - For PA-RISC HP-UX, JRE version 1.6.0.03 or later is required
 - "Too many open files" errors after deploying probe on an IBM Java 6 64-bit JVM (60289)
A bug in some versions of IBM's Java implementation may (APAR IZ 99243) cause the application to crash with a "too many open files" error when used with Diagnostics Agent. The affected JVMs do not run any object finalizers when invoked with a -javaagent option pointing to an agent which declares its capability to instrument classes dynamically ("Can-Redefine-Classes" is set to true in the manifest file).
Workaround: Invoke the Diagnostics Agent without using "-javaagent". After running the jreinstrumenter for the JRE in question, use the java options recommended by the jreinstrumenter

with the -javaagent option removed, and with
<probe_install_dir>/classes/boot directory added to the -Xbootclasspath.
For example:

```
-Xbootclasspath/p:/opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent/classes/IBM/1.6.0/instr.jre:  
/opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent/classes/boot -Xshareclasses:none
```

- JBoss application fails to startup with RMI instrumentation (62520)

Problem: After RMI instrumentation has been activated on a JBOSS v9 application server, there is an Exception on the JBOSS start-up log as seen below.

Caused by: java.lang.ClassNotFoundException: No ClassLoaders found for:
com.mercury.opal.capture.RMIRemoteProxy (no security manager: RMI class loader disabled)

Resolution: Customer should setup JBoss such that there is a security manager with an appropriate policy that includes permissions for codebase paths

```
codebase = "file:<probe_install_dir>/etc/  
file:<probe_install_dir>/classes/ file:<probe_install_dir>/lib/probe.jar  
file:<probe_install_dir>/lib/ThirdPartyLibs.jar".
```

The following example enables a security manager with all permissions. Customer should setup policy with restrictions as appropriate:

```
-Djava.security.manager -Djava.security.policy=<full_path_name_of_policy_file>  
on the client side.
```

```
grant {  
permission java.security.AllPermission;
```

- When using basic or automatic implicit JRE instrumentation with an IBM 1.6.0 2.6 JVM, a JVM crash will occur. (72394)

For example:

```
17:31:40.791 0x4839d09000 j9mm.141 * ** ASSERTION FAILED ** at ./ParallelScavenger.cpp:2312:  
((false))
```

JVMDUMP006I Processing dump event "traceassert", detail "" - please wait.

Workaround. To work-around this problem, use the new Diagnostics 9.20 Automatic Explicit JRE instrumentation mode or the older Manual JRE instrumentation mode to avoid the problem.

- Special configuration needed to support SAP JVM NetWeaver 7.1 and later (43388)

Because NetWeaver 7.1 runs in a clustered environment, you will need to do one of the following in order to distinguish the probes in the cluster:

- Find a way to set the probe.id for each individual server process in your cluster (an expert SAP administrator might be able to help).
- If the above is not possible, then you need to specify the keyword %0 in the probe name. Doing this will name each probe in the cluster uniquely. For example, if you specify "- Dprobe.id=myProbe%0" then the first probe that comes up will be called "myProbe0", the second one "myProbe1" and so on.
- There is a potential issue on HP-UX 11.23 with IBM WebSphere: When multiple probes are installed on the same system, it can happen that the probe's communication port (35000) will be overwritten with each startup of the probe. In this case, simply specify a different port range for each of the probes in etc/webserver.properties, for example:

- `jetty.port=36000`

- `jetty.max.port=36100`

Or specify the above parameters on the app server's startup configuration (via -D). (44062)

- Using the Heapwalker feature with JRockit JVM version 27.3 may cause the application to crash. Versions 27.2 and 27.4 and higher work as expected, so the problem is only with version 27.3. Please upgrade to a higher version of JRockit to fix this problem. (43993)
- When instrumenting an IBM 1.6 JVM on z/OS only, using the output of the JRE Instrumenter, the JVM will fail to start with a `java/lang/NoClassDefFoundError` exception. (50969)

To correct this issue, modify the `-Xbootclasspath` parameter by appending `":<probe_install_dir>/classes/boot"` substituting for `<probe_install_dir>` appropriately. In particular note the required colon after `instr.jre`.

For example, if the output of the JRE Instrumenter was:

```
"-Xbootclasspath/p:/opt/JavaAgent/DiagnosticsAgent/classes/IBM/1.6.0/instr.jre"  
"-javaagent:/opt/JavaAgent/DiagnosticsAgent/lib/probeagent.jar"  
-Xshareclasses:none
```

Change it to:

```
"-Xbootclasspath/p:/opt/JavaAgent/DiagnosticsAgent/classes/IBM/1.6.0/instr.jre:  
/opt/JavaAgent/DiagnosticsAgent/classes/boot"  
"-javaagent:/opt/JavaAgent/DiagnosticsAgent/lib/probeagent.jar"  
-Xshareclasses:none
```

- When the Diagnostics/TransactionVision Agent is run in "dual" mode, there is a limitation on JDBC calls. "Dual" mode means that both Diagnostics Java probe and TransactionVision Java sensor are enabled on the system. In this configuration, Diagnostics has the following limitations:
 - 1) No data will show up from nodes in "dual" mode in the "SQL Statements" view.
 - 2) When drilling down to a Call Profile, JDBC calls will not show the SQL statement in the Arguments of the call.

This problem will be fixed in a future release of Diagnostics. (42989)

- GC Time Spent in Collections metric may be inaccurate (JRockit JREs only)

The problem is caused by a bug in JRockit JRE which reports the time spent in GC in nanoseconds rather than in milliseconds, as specified.

Workaround: either upgrade JRockit to version 1.5.0_10, build R27.2.0 (or later) or modify the Rate setting as follows:

In `<probe_install_dir>/etc/metrics.config`, change `"[0.1]"` to `"[0.0000001]"` in the line which defines "GC Time Spent in Collections" metric. For example:

```
Java\Platform\java.lang:type=GarbageCollector,*.CollectionTime = RATE[0.0000001](GC Time Spent in  
Collections|percent|GC) (42344)
```

- If the following WARN message is found in the `probe.log` file on the probe system:
2006-11-28 07:07:27,171 WARN com.mercury.opal.capture [ExecuteThread: '8' for queue:
'weblogic.kernel.Default'] Maximum number of SQL queries cached (4096). The values of some prepared
SQL queries will be lost. See `sql.cache.size` in `capture.properties`.
Then you will need to increase the `sql.cache.size` in the `capture.properties` file on the probe system.
(40639)
- By default, Diagnostics does not monitor server requests that always execute in under 51 milliseconds.

These requests are trimmed-no information is captured about them. SQL statement executions made from these trimmed requests are not recorded, even if those same SQL statements are timed and recorded in the context of executing other server requests. In addition, the probe-level layer breakdown does not include time taken by trimmed server requests.

An exception to this is in a case where, on at least one occasion, a server request takes longer than 51ms to execute. Future executions of that server request are recorded, even if the subsequent requests are faster than 51ms. The reason for this behavior is to report accurate averages (and not mislead the user into thinking that this request only ran once or twice when in actual fact it is constantly running, just very quickly). This "Always Record" flag lasts for 1 hour from the last time an execution over 51ms was seen.

You can redefine the 51ms trimming threshold. For the Java Probe integrated with LoadRunner or Performance Center, you configure this setting in the <diagnostics_server_install_dir>\etc\trimming.properties file. For the Java Probe in all modes, you configure this setting in the minimum.fragment.latency property in the <probe_install_dir>\etc\dispatcher.properties file. (40631)

- A defect in the JVM on Linux (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6330858) results in the thread CPU times being the same for all threads in BEA WebLogic 9.0. This bug was fixed in JDK 1.5.0_07. (40391, 39387)
- The Java Probe causes core dump at startup of WebLogic 9.1 when used with JRockit VM version 150_04. The JRockit bug is fixed on jdk1.5.0_06. (38545)
- By default, for performance reasons, Diagnostics does not capture target information for Database calls.
- VM Heap Breakdown is based on an experimental api (JVMPi) in the JVM and is not expected to work in the following JVMs:
 - Sun 1.4.2_01: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4861809
 - Sun 1.4.2: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4899339 {open}
 - IBM JREs in general do not have a stable API implementation for this API
- Running the profiler directly on z/OS is not supported. The script "profiler.sh" has been removed in the 7.50 version of the z/OS probe, it will exist in previous versions but cannot be successfully run on z/OS. To run the profiler for a probe on a z/OS system, do so from a platform other than z/OS. (42369)

.NET Agent

- After installing the .NET agent, Microsoft SharePoint 2013 may not function correctly. To fix this you can apply the following workaround:
 - a. Open the SharePoint web.config file for editing. By default this file is located in **C:\inetpub\wwwroot\wss\VirtualDirectories\80**.
 - b. Change the **legacyCasModel** setting from true to false, as follows:

```
<trust level="Full" originUrl="" legacyCasModel="false" />
```

- c. Restart IIS by using either IIS Manager or the IISReset command-line utility. (82212)
- By default, Diagnostics does not monitor server requests that always execute in under 51 milliseconds. These requests are trimmed-no information is captured about them. SQL statement executions made from these trimmed requests are not recorded, even if those same SQL statements are timed and recorded in

the context of executing other server requests. In addition, the probe-level layer breakdown does not include time taken by trimmed server requests.

An exception to this is in a case where, on at least one occasion, a server request takes longer than 51ms to execute. Future executions of that server request are recorded, even if the subsequent requests are faster than 51ms. The reason for this behavior is to report accurate averages (and not mislead the user into thinking that this request only ran once or twice when in actual fact it is constantly running, just very quickly). This "Always Record" flag lasts for 1 hour from the last time an execution over 51ms was seen.

You can redefine the 51ms trimming threshold. For the .NET Probe in all modes you configure this setting in the <diagnostics_server_install_dir>\etc\trimming.properties file. All .NET Probe trimming configuration is done in the <.NET_probe_install_dir>\etc\probe_config.xml file. (40631)

- By default, for performance reasons, Diagnostics does not capture target information for Database calls. Therefore, these calls do not show up in the Outbound Calls view. You can enable it as follows:
 - For .NET Probes, set create.database.fragmentArcs=true in the server.properties file for the Diagnostics Server in Mediator mode. You should also use this workaround if enable.probe.aggregation in the Java probe has been changed from its default value of true to false.Once capturing of Outbound database calls has been enabled, these calls will be displayed in the Outbound Calls view. (40722)
- On platforms where Windows Process Activation (WPAS) is installed, the .NET Diagnostics agent uninstall has an implicit runtime dependency on up to four "listener adapter" services. During the uninstall, MSI pops up a "The following processes need to be closed before continuing..." dialog if any of the listener adapter services are started. Stopping them (temporarily) allows the uninstall to continue. However, the services do NOT have to be stopped in order for removal to end successfully. The user can select the "Ignore" button on this dialog and continue uninstallation. (38151)
- If you have a problem displaying probe metrics by .NET Agent, note that if you have done the configuration described in the KnowledgeBase article <http://support.microsoft.com/default.aspx?scid=kb;EN-US;282884> Diagnostics .NET agent will not be able to show probe metrics.

Integrations (BSM, PC, LR, TV, SaaS)

- The logs indicate OOM errors are happening to the embedded Diagnostics UI in LoadRunner Controller. (64938)

When you open the J2EE/.NET Diagnostics tab in LoadRunner Controller during a load test, the system CPU spikes and sometimes stays high.

Workaround: Upgrade the JRE on the LoadRunner system.

- Diagnostics Server does not work with IIS Basic Authentication with Business Availability Center and BSM Reverse Proxy Server. (43987)
- In the BSM-Diagnostics integration (Admin>Diagnostics) we get an error on short-name and there is no way to re-enter the FQDN. (62229)

You should use the FQDN when registering the Diagnostics Server with BSM. However if you entered a non-FQDN for the Diagnostics Server you could get stuck in a registration problem loop.

Workaround. Use the URL below and substitute the "topazhost" name in the URL with the FQDN BSM 9.10 Gateway Server to clear the Diagnostics Configuration page in BSM.

http://topazhost/opal/ui/bridge/action/registration.do?action=set_registration&customer_name=Default%20Client&opal_protocol=&opal_port=&opal_full_host=

- During installation of Operations agent components, certificate requests from Diagnostics (and IA, and SiteScope) nodes are blocked at Load Balancer/Reverse Proxy if BSM has been configured for HTTPS communication. (QCCR1A122094)

During installation of Operations agents, a certificate request is sent to the certificate server. The protocol used for the request is HTTP. If the BSM Reverse Proxy/Load Balancer is configured for HTTPS communication, it blocks all HTTP communication including the certificate request.

Workaround. Generate a certificate manually and install it manually on the IA/SiteScope/Diagnostics node. You can refer to the topic "Configure HP Operations Agent manually" in the HP BSM Integration Adapter Installation and Upgrade Guide in the BSM documentation library and do something similar for Diagnostics.

- Offline vs. Online. The following features and functionality in LoadRunner Offline Analysis are different from the Diagnostics Online.
 - Data in the profiler that is not sent into the Diagnostics Server will not be in the offline analysis after Performance Center/LoadRunner runs. This includes LWMD, Heap Breakdown, Allocation Analysis, Exceptions, SOAP Faults and SOAP payloads.
 - Oracle 10g data will not show up in offline analysis.
 - Instance trees will not be available in the offline analysis only aggregate trees after the needed drill down.
- When configuring Real User Monitoring in Business Availability Center to use a Page definition including parameters, Diagnostics will not be able to make a match unless parameter capturing has been enabled. To enable this option on the Java Probe, list all desired parameter names in reverse alphabetical order in the args_by_class property in auto_detect.points under the "[HttpCorrelation]" section. (40496)
- In BSM's Transaction Management Admin UI > Monitoring tab > Search Criteria, when the Type dropdown list is expanded, duplicate or triplicate Server Request types are shown. (60740)
- HP Performance Center offline files are kept by default. To manage offline files, you need to configure the Diagnostics Servers in Mediator mode so that they delete these files. You do this by setting the property distributor.offlinedelivery.preserveFiles to true in the <diagnostics_server_install_dir>/etc/server.properties file. When set to true, this property causes the run-specific "offline" files stored in the server's data directory to be retained for the amount of time specified in the facade.run_delete_delay property in the server's webserver.properties file (default period is 5 days). During this retention period, the run can be successfully collated. Sometime after the retention period has ended, the associated offline files will be deleted from the system. (40739)
- Performance Center users that drill to Diagnostics get full admin privileges to the Diagnostics Server and to the Probes that are connected to it. All restricted actions (for example, changing thresholds, alerts, and Custom Attributes) are accessible by all Performance Center users. Furthermore, every custom screen created by a Performance Center user is shared by all Performance Center users. (40133)
- Starting in Diagnostics 7.0, many instrumented methods have been flagged when-root-rename in auto_detect.points to prevent many spurious and uninteresting server requests from being created and displayed in the UI. Should these methods execute outside the context of another server request, they will still be recorded, but into a pseudo server request with the name "Background - <Layer>", where <Layer> is the layer name for the method.

For example, the Background JDBC connection testing that WebLogic runs used to be reported into a few separate server requests, such as `netJDBCPreparedStatement.executeQuery()`, but will now be recorded as invocations of a "Background – Database" pseudo server request.

In addition, when using LoadRunner, these new pseudo server requests will not appear in LoadRunner Offline Analysis. Should you wish to keep the old behavior for a particular instrumentation point and see these entries in LoadRunner Offline Analysis, carefully remove the `when-root-rename` detail parameter from the `auto_detect.points`. (42127)

- When Diagnostics is integrated with HP Software-as-a-Service (SaaS), you must log on through Business Availability Center when viewing data for customers other than "Default Client". Failing to do so will cause several subtle issues with how data is reported. You can still use the standalone login to view the "Default Client" data. (42178)
- LoadRunner Collate fails when the Diagnostics Server was restarted during Performance Center run/load test. (34682)

Integration With APM 9.30

In previous releases of BSM, Diagnostics sent CIs, metrics, and events to BSM. Health Indicator status (coloring) for the Application Infrastructure CIs was based on events sent to BSM through the OMi event channel.

In APM 9.30, due to the OMi removal, Diagnostics sends events to OMi and continues to send CIs and metrics to APM. As a result of this change, the Application Infrastructure CIs (Diagnostic Probe, Oracle/MSSql Database, IBM WebSphere MQ, WebSphere AS Dynamic Caching, IBM WebSphere MQ, IBM MQ, WebLogic AS, JBOSS AS, SQL Server, MSSQL Database, Oracle iAS, Oracle, SAP, SAP R3 Server, VMware ESX Server, Host Node) do not show the correct HI Status in APM 9.30.

The HI status for these CIs appears as *Undetermined* (blue question mark). If you are upgrading from BSM 9.2x to APM 9.30, the last status received for the CI before the upgrade will be displayed.

Use the workaround below to view the HI status of these CIs in OMi. (QCCR11117984)

Workaround: To enable sending events related to threshold violation in Diagnostics to OMi, in the `server\bin` directory of the Diagnostics server:

1. Run `cscript switch_ovo_agent.vbs -server <FQDN of OMi> -cert_srv <FQDN of OMi>`.
2. Go to **OMi Administration > Certificate Requests** and grant certificates.
3. Run `cscript switch_ovo_agent.vbs -server <FQDN of OMi> -cert_srv <FQDN of OMi>` again.
All tasks should now display correctly.

Collectors (Oracle, SAP, MQ, TIBCO EMS, SQL Server, webMethods, VMware)

- To maintain accurate MS SQL Server metrics, the database option `AUTO_CLOSE` must be set to OFF. If `AUTO_CLOSE` is set to ON, MS SQL Server metrics will show up with incorrect and negative values after certain database operations (`bcp`, `backup`, `shrink`, etc) (43075).
- NPE for legacy (pre-9.10) MQ collector data. (54829)

Workaround. MQ channel data will populate in the BSM RTSM when 9.10 MQ collector is installed.

Java Profiler

- Call Profile window opened by standalone profiler does not react to mouse clicks (QCCR1199448)
If the Java Profiler uses OpenJDK version 1.7, when viewing Call Profiles for a Server Request, you are unable to select a specific call in the call stack.
- Starting with Diagnostics 9.0, the JAVA Profiler requires JVM 1.6 to run. However, on HP-UX, the default JVM that is installed with the Java Profiler is 1.5. Therefore in order to run standalone Java profiler, the following must be performed. NOTE - The Java Profiler runs fine on HP-UX when launched from the enterprise UI. To run the Java profiler in standalone mode, do the following:
 - Set JAVA_HOME to your java 1.6 installation location
 - export JAVA_HOME=<java 1.6 home directory path>
 - example: export JAVA_HOME=/opt/java6
 - Run the java profiler: *\$PROBE_HOME/bin/profiler.sh &*
- NPE and HTMLEditorKit prevents UI from starting more than once (55258)
Fixed this JRE issue (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6993073) by implementing the workaround (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6993691) in Diagnostics so the JRE does not throw an exception.
But it is important to note that since the Java probe install includes the Profiler UI - then each probe where you run the Profiler UI also needs to be updated or patched to get this fix. Otherwise the Profiler UI applet will only load once in the browser. You can download a patch for Java probe versions 8.0x or 9.0x and the fix is also included if you upgrade the Java probe to version 9.10.
- Call Profile window opened by standalone profiler does not react to mouse clicks (QCCR1199448)
If the Java Profiler uses OpenJDK version 1.7, when viewing Call Profiles for a Server Request, you are unable to select a specific call in the call stack.

.NET Profiler

- When running the .NET Profiler under locales that use an Asian character set, the default font size may be too small. In such cases, change the text size used by the Web browser. (43070)
- Launching the .NET Profiler from the Diagnostics Enterprise UI fails for .NET Agents running the Probe Aggregator Service
Problem. Selecting the **View Profiler** command on a .NET probe that is using the Probe Aggregator Service prompts for credentials. However valid credentials are not accepted.
Workaround. Access the .NET Profiler in standalone mode.
If standalone mode access is not an appropriate solution, you can fix the access from the Diagnostic Enterprise UI as follows:
 - a. On the .NET Agent host, edit the **<probe_install_dir>/etc/probe_config.xml**.
 - b. Change the following entry:

```
<diagnosticsserver url="http://127.0.0.1:45000/commander" />
```

to

```
<diagnosticsserver url="http://<mediator>:<port>/commander" />
```

where <mediator> is the host name or IP address of the Diagnostics Mediator Server to which the probe is sending collected data to and <port> is the mediator port. For example, url="http://mymediatorhost:2006/commander".

- c. After a few seconds, try to access the .NET Profiler from the Diagnostics Enterprise UI. You do not need to restart the application, the probes will dynamically re-register using this new setting.

(QCCR1I92693)

Diagnostics Server

- Server Install fails - Permission denied. (51729)

If the Diagnostics commanding server will be integrated with Business Service Management 9.00 or later, root access is required during the Diagnostics Server installation. Root access is required for the installation of the OM Agent and IAPA component.

Workaround. If you need to install the Diagnostics Server without root access you can chose to not install these two components and install them later manually. When you see the dialog box: OM Agent and IAPA component installations leave the box unchecked and install later.

- When enabling HTTPS between Diagnostics components the user should monitor the <install_dir>/MercuryDiagnostics/server/log/jetty.log for warnings similar to:

```
2008-11-17 15:33:07,528: WARNING : WARN!! [RangeSocketListener-69]org.mortbay.http.SocketListener.isOutOfResources(SocketListener.java:358)22> OUT OF THREADS:
```

```
RangeSocketListener@0.0.0.0:8443
```

```
2008-11-17 15:33:35,582: INFO : EVENT [RangeSocketListener-14]org.mortbay.http.SocketListener.isLowOnResources(SocketListener.java:325)04> LOW ON THREADS ((200-198+7)<10) on RangeSocketListener@0.0.0.0:8443
```

Workaround. If these messages appear, increase the value of jetty.threads.max by increments of 100 in the <install_dir>/MercuryDiagnostics/server/etc/webserver.properties file.

Continue to monitor and increase if necessary. (44007)

- After configuring the Diagnostics Server in Mediator mode, if it appears missing or inactive in the Diagnostics Server in Commander mode view, then it is possible that "commander.url" in server.properties has an ending "/" on it. For example, if you open server.properties on the Diagnostics Server in Mediator mode and set to commander.url=http://amkisty01:2006/ instead of commander.url=http://amkisty01:2006 (Note that the difference is the appending /), then the Diagnostics Server in Mediator mode will appear missing or inactive. (40527)
- Broadly scoped allocation instrumentation may fill up perm-gen and cause the VM to crash. Users should avoid widely scoped instrumentation and instead make instrumentation as narrow as useful to minimize system impact. (39339)
- When average latencies are less than 50ms, users may see CPU time slightly higher than latencies because of resolution differences. (39690)
- Entity purging results in contribution/breakdown not summing up to the total because the deleted entities

data was already rolled up and is not recalculated at this point. (39344)

- Running the Diagnostics Server from a network drive, or configuring it to store the archive on a network drive, is not supported.
- When you configure the Diagnostics Server to run only in SSL mode, it only affects port 2006. The Diagnostics Server also has an embedded Diagnostics Probe (for internal troubleshooting purposes) that listens, as usual, on port 35000. If you are trying to lock your environment to SSL only for security purposes, you will also need to re-configure that embedded probe to listen only over SSL.
- (Documentation Correction) If you are importing threshold and alert definitions from a **script.thresholding** file, the **script.thresholding** file must be placed into the **<diagnostics_server_install_dir>/storage/thresholding/once/Default Client** directory. Be sure to include the space in the "Default Client" directory name, otherwise the import will fail. (QCCR1192662)

User Interface

- Change "Active Nodes" in UI to "Active Entities" . (99955)

The internal server's probe metric **Active Nodes** was renamed as **Active Entities**. After an upgrade, you may still see the old name in the **Details** pane (under **Online Cache**). If you add this metric as a chart, the new name is displayed as the chart's title. (Right click the metric in the **Details** pane and select **Start Charting this Metric.**)

- If a custom view, for example, is to be used for a Java probe and one of the features in that custom view shows Top 7 Business Transactions, because business transactions are not sub-entities of a Java probe, the graph feature will be blank.

Workaround. Save the opened view. The data will then be shown in the view.

- When a custom view is created and the view is used with "Open in Custom View", the UI will attempt to open all features with the appropriate filters for that entity. If a feature in the custom view does not match the filters for the entity, then the view will be blank. (71123)
- When a custom view is created from a detail view (as opposed to a dashboard view), when navigating away from the view via a drill or other navigation, often the UI will prompt the user if changes should be saved, even though nothing was changed. Always press "Yes" and continue. Pressing "No" will result in a blank screen and an error stating that the view is corrupt (although no permanent damage occurs). This situation will be fixed in a patch release of Diagnostics 9.20 in the near future. (71505)

Workaround. Press Yes if prompted.

- With IE 8 the Maintenance Link is missing on the Diagnostics Enterprise UI when launched from within BSM UI (BSM and Diagnostics are running on the same system). (55189)

Workaround. Close the BSM UI and the Diagnostics Standalone UI, and re-launch the Diagnostics Standalone UI. Also note it works fine when running with IE7 or launched stand-alone in IE8 (not inside of BSM).

- The ability to delete more than one alert rule at a time in the UI is currently not supported.

Workaround. Select and delete one alert rule at a time.

- NPE and HTMLEditorKit prevents UI from starting more than once (55258)

Fixed this JRE issue (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6993073) by implementing the workaround (http://bugs.sun.com/bugdatabase/view_bug.do?but_id=6993691) in Diagnostics so the JRE does not throw an exception.

But it is important to note that since the Java probe install includes the Profiler UI – then each probe where you run the Profiler UI also needs to be updated or patched to get this fix. Otherwise the Profiler UI applet will only load once in the browser. You can download a patch for Java probe versions 8.0x or 9.0x and the fix is also included if you upgrade the Java probe to version 9.10.

- Enterprise UI: OutOfMemoryError after repeatedly drilling down to Java profilers. (57043)

Workaround. Sometimes, when launching the profiler from the Enterprise UI, the entire UI (EUI, profiler and Java console) will hang and, shortly thereafter, the profile and Java console windows disappear and the remaining EUI window would not repaint. This problem can be eliminated by upgrading the JRE used by the user's browser to a more current version of 1.6.

- NPE when loading System Health. (57363)

Workaround. Open a new browser or you can use the new System Health view group available in the Enterprise UI.

- Prior to exporting an Application Explorer view to pdf or html you should examine all of the tabs to activate the population to the desired charts, otherwise the exported chart could be empty. (28801)
- Navigation menu items on the popup (context) menu are different when right-clicked alone vs. selected and then right-clicked. This occurs because only limited information is available when the first right-click on an entity occurs, and the context menu is populated with what is known at that time. However, the right-click operation immediately kicks off a query to the Diagnostics Server to find out more about this entity, and so subsequent right-click operations use the returned data and the popup menu includes more options. (29253)
- In a custom 2-way view, both of the Navigation controls are updated with the navigations for the selected entity. (29097)
- When an application is selected (Application Explorer view) or a Probe Group is selected (Server Summary view) you may see two "View Probe" navigations. This will happen when the Application or Probe Group contains both .Net and Java Probes. (26471)
- The navigation control only tracks the global time control. (29268)
- Drilling from some alerts to "View Threshold Violation" results in a view that is empty except for the message "Either no data is available for the selected time range, or the data at this level of detail has been purged from the database."

This generally happens when the violating entity is 2 levels below the alert entity. For example, a probe group alert from a server request threshold or a probe alert on a portal component–life cycle method alert.

Workaround: Notice the breadcrumb of the empty view. This indicates the view that shows the data causing threshold violation. Add the alert to a snapshot. From the snapshot, use the Navigations pane to drill-down. In most cases, you would navigate to Probes first and the destination view below that. For Life Cycle Methods, navigate to Portal Components (under Probes) first. For better navigation from alerts to threshold violations, keep alert rules closer to the entities being monitored. (27612)

- For server requests under 10 ms, Minimum Time will always be reported as zero. (34125)

Documentation

If you experience a JavaScript error when opening the HP Diagnostics online help system, disable the "Show Exception Dialog Box" in the Java Console and open the help again.

Translated Versions

Diagnostics has been translated into a number of languages. When using Diagnostics, occasionally text that should be translated will be shown in English. This is due to limitations in third party components that are used by Diagnostics. We apologize for any inconvenience that this may cause.

Localization and Globalization

Localization is the process by which software applications are adapted to suit a specific local market or *locale*. Globalization is the process of designing products or services for the global marketplace so that they can be sold anywhere in the world with only minor revision.

HP supplies Diagnostics localized software in these languages:

- English
- Simplified Chinese
- Korean
- Japanese
- French
- German
- Spanish
- Russian

HP supplies Diagnostics localized documentation in these languages:

- Japanese
- Simplified Chinese

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 Release Notes (Diagnostics 9.26)

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 sw-doc@hpe.com.

We appreciate your feedback!