HP Diagnostics

For the Windows ®, UNIX and Linux operating systems

Software Version: 9.20

Release Notes

Document Release Date: May 2012

Software Release Date: May 2012

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

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:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online 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 web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP 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 HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

Introduction

This file provides the following information about HP Diagnostics:

Documentation

The first page of this Release Notes document contains the following identifying information:

- Version number, which indicates the software version.
- Publish date, which changes each time the document is updated.

In addition to this document, please see the Upgrade_Patch_Install_Instructions.pdf (in each download package) for important upgrade instructions.

Note the documentation now includes a Frequently Asked Questions document, a Diagnostics Data Model (Query) API document and a link to Diagnostics videos on YouTube. In the online help you can access the following documentation:

- “Help on View” (from the new Tools and Information menu) – provides a help topic for the currently displayed view or page in the UI
- “Help” (from the new Tools and Information menu) – opens the Diagnostics documentation library and online help contents
- Installation and Configuration Guide (pdf format)
- User’s Guide (help topics provided in pdf format)
- Diagnostics FAQ document (pdf format)
- Diagnostics Data Model and Query API Guide (pdf format)
- Link to video recordings

The full Diagnostics documentation set was updated this release and is included with the product. The manuals are also posted on the SSO documentation download site for this release. To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

You must have Adobe® Reader installed to view files in PDF format (*.pdf). To download Adobe Reader, go to the Adobe web site.

To retrieve a document, select the:

1. **Product** name.
2. **Version** list.

3. **Operating System**.
4. Preferred **Language**.
5. Document title.
6. Click **Open** or **Download**.

You must have Adobe® Reader installed to view files in PDF format (*.pdf). To download Adobe Reader, go to the Adobe web site.

Installation Notes

Diagnostics Server, Agent and Collector installation requirements and detailed instructions are provided in the *HP Diagnostics Installation and Configuration Guide*. For the most recent information on supported environments refer to the Diagnostics Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp.

Diagnostics 9.20 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 Installation and Configuration Guide* in Appendix G.

The upgrade installation instructions are also provided in a document available with each download zip (Upgrade_Patch_Install_Instructions.pdf). These upgrade install instructions are a duplicate of Appendix G, provided in each download zip as a convenience.

Note on Using the Windows Installers in Diagnostics:

With the Diagnostics 9.10 or later releases - if you want to use command line options with any of the following Windows installers you must precede the options with –a.

HPDiagCollector_<releasenumber>_win.exe

HPDiagTVJvaAgt_<releasenumber>_win.exe

HPDiagServer_<releasenum>_win32.exe

HPDiagServer_<releasenumber>_win64.exe

For example for a silent install of the Diagnostics Server on 32-bit Windows system:

```
HPDiagServer_9.10_win32.exe –a –silent –options c:\responsefile1.rsp
```

Note on Probe Backwards Compatibility:

Diagnostics 9.02, 9.10, 9.12, and 9.20 probes are not compatible with 9.0/01 or earlier Diagnostics servers.

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.

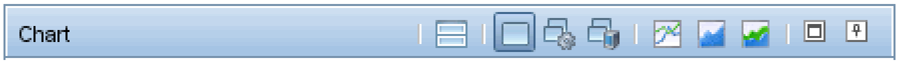
  Benefits: Functionality, Usability

- 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.

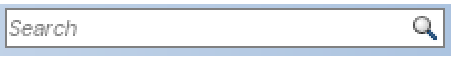
• The chart column is now combined color and chart.

• New ICONS for:

- Select the columns to display in the table

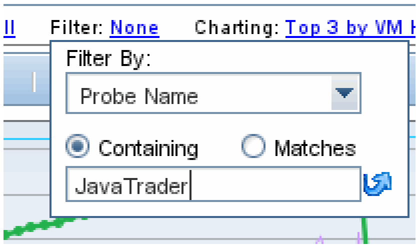
- Search the table

• 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:



Filter: None Charting: Top 3 by VM I

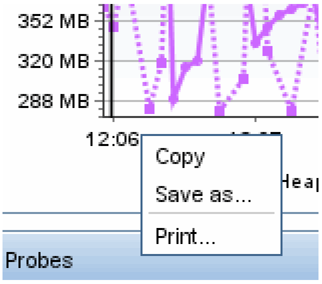
Filter By:

Probe Name

● Containing ○ Matches

JavaTrader

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



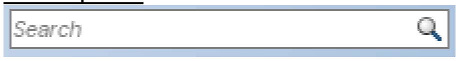
For more information on using the Diagnostics UI, see “Common Controls in the Diagnostics UI” in Chapter 2 of the User’s Guide.

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

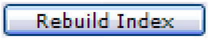
  in:

  • 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 Chapter 2 of the User’s 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 captures 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, please see “Client Monitoring Views”, Chapter 39 of the User’s 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

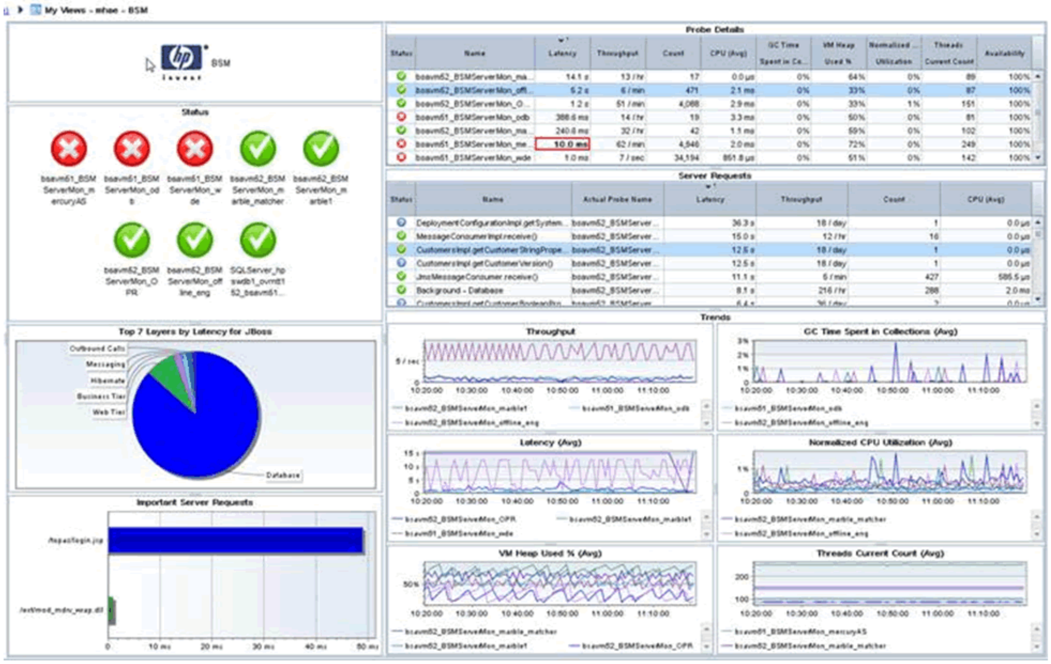
• Probe

• Server Request

• SQL

• Synthetic Transaction

See “Working with Custom Views and Dashboards” in Chapter 7 of the User’s Guide for more details on how to get started with this new feature. Here is an example of what can be created:



Probe Details

Status	Name	Latency	Throughput	Count	CPU (Avg)	GC Time Spent in Co...	VM Heap Used %	Normalized ... Utilization	Threads Current Count	Availability
✓	baavm62_BSMServerMon_ma...	14.1 s	13 / hr	17	0.0 µs	0%	64%	0%	89	100%
✓	baavm62_BSMServerMon_off...	6.2 s	6 / min	471	2.1 ms	0%	33%	0%	87	100%
✓	baavm62_BSMServerMon_O...	1.2 s	51 / min	4,088	2.9 ms	0%	33%	1%	151	100%
✗	baavm61_BSMServerMon_odb	388.6 ms	14 / hr	19	3.3 ms	0%	50%	0%	81	100%
✓	baavm62_BSMServerMon_ma...	240.8 ms	32 / hr	42	1.1 ms	0%	59%	0%	102	100%
✗	baavm61_BSMServerMon_me...	10.0 ms	62 / min	4,946	2.0 ms	0%	72%	0%	249	100%
✗	baavm61_BSMServerMon_wde	1.0 ms	7 / sec	34,194	861.8 µs	0%	51%	0%	142	100%

Server Requests

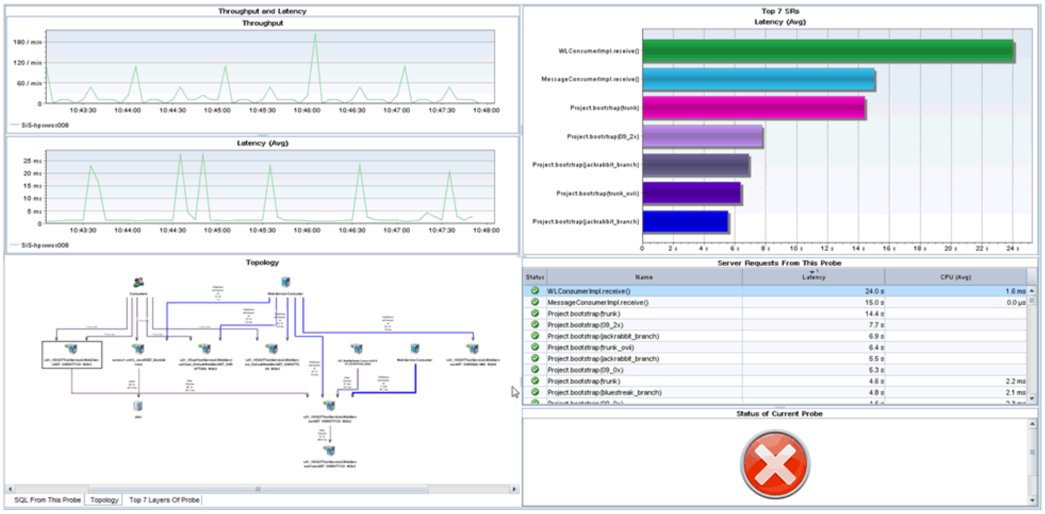
Status	Name	Actual Probe Name	Latency	Throughput	Count	CPU (Avg)
○	DeploymentConfigurationImpl.getSystem...	baavm62_BSMServer...	36.3 s	18 / day	1	0.0 µs
✓	MessageConsumerImpl.receive()	baavm62_BSMServer...	15.0 s	12 / hr	16	0.0 µs
✓	CustomersImpl.getCustomerStringPrope...	baavm62_BSMServer...	12.6 s	18 / day	1	0.0 µs
○	CustomersImpl.getCustomerVersion()	baavm62_BSMServer...	12.5 s	18 / day	1	0.0 µs
✓	JmsMessageConsumer.receive()	baavm62_BSMServer...	11.1 s	6 / min	427	586.5 µs
✓	Background - Database	baavm62_BSMServer...	8.1 s	216 / hr	288	2.0 ms
○	CustomersImpl.getCustomerBooleanPro...	baavm62_BSMServer...	6.4 s	36 / day	2	0.0 µs

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:



Server Requests From This Probe

Status	Name	Latency	CPU (Avg)
✓	WLConsumerImpl.receive()	24.0 s	1.6 ms
✓	MessageConsumerImpl.receive()	15.0 s	0.0 µs
✓	Project.bootstrap(trunk)	14.4 s	
✓	Project.bootstrap(09_2x)	7.7 s	
✓	Project.bootstrap(jackrabbit_branch)	6.9 s	
✓	Project.bootstrap(trunk_ovii)	6.4 s	
✓	Project.bootstrap(jackrabbit_branch)	5.5 s	
✓	Project.bootstrap(09_0x)	5.3 s	
✓	Project.bootstrap(trunk)	4.6 s	2.2 ms
✓	Project.bootstrap(bluestreak_branch)	4.8 s	2.1 ms

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)

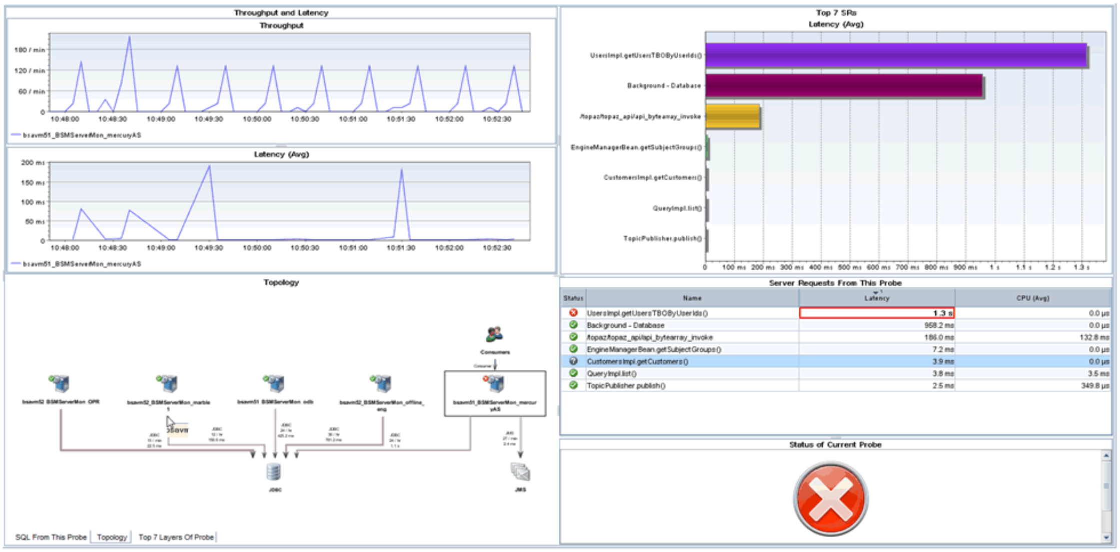
  - A tab with a topology graph of the application server (shown)

  - 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:



Server Requests From This Probe

Status	Name	Latency	CPU (Avg)
✕	UsersImpl.getUsersTBOByUserIds()	1.3 s	0.0 µs
✓	Background - Database	958.2 ms	0.0 µs
✓	/topaz/topaz_api/api_bytearray_invoke	186.0 ms	132.8 ms
✓	EngineManagerBean.getSubjectGroups()	7.2 ms	0.0 µs
?	CustomersImpl.getCustomers()	3.9 ms	0.0 µs
✓	QueryImpl.list()	3.8 ms	3.5 ms
✓	TopicPublisher.publish()	2.5 ms	349.8 µs

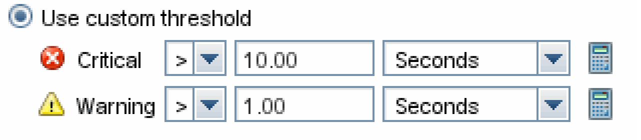
Get creative in using the Dashboard Creator and Open in Custom View features!

Benefits: Usability

- Critical and Warning Thresholds/Alerts and Normal Alerts

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

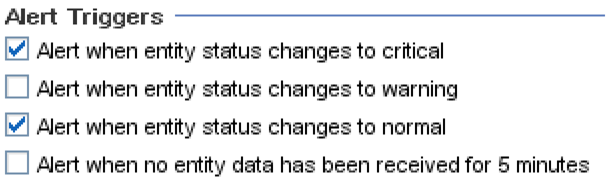
  Description: With this release, the user can set a Critical and a Warning threshold. Each threshold set via the Set Threshold dialog box presents the user with two choices now instead of one:



● 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

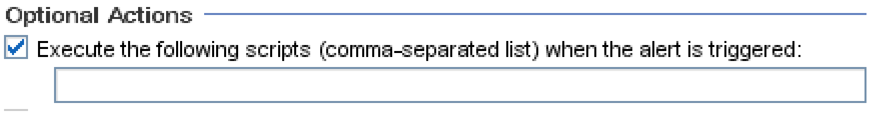
  See “Working with Thresholds and Metrics in the Details Pane” in Chapter 4 and “Working with Alerts and Rules” Chapter 5 of the User’s Guide for more details.

  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 Chapter 5 of the User’s 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 export (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”, Chapter 2 of the User’s Guide.

  Benefits: Functionality, Supportability

- New JRE Instrumener 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 awaken 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 us 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.

What's New in 9.12

Diagnostics 9.12 has a number of defect fixes and new features, click the following links for details:

- 9.12 New Features
- 9.12 Defect Fixes

New Features in 9.12

- Improved support for Software as a Service.

  Background: In Diagnostics 8.x, HP offered a Software as a Service (SaaS) solution which included an HP Diagnostics component through BSM. Earlier versions of BSM 9 did not support SaaS.

  Description: Starting with a later release of BSM 9.12, HP will offer an upgraded version of SaaS and Diagnostics 9.12 has a number of features to improve the SaaS offering. If you are interested in HP's SaaS offering, see http://www8.hp.com/us/en/software/software-solution.html?compURI=tcm:245-936891 and talk to your sales representative.

  Benefits: Performance

- A user can install and monitor an Azure application using the Diagnostics .NET instrumentation agent.

  Description: Microsoft provides Windows Azure SDK for developers to create and deploy Azure applications to the Windows Azure Cloud Infrastructure. The HP Diagnostics .NET Agent leverages the Azure SDK to provide seamless deployment of the HP Diagnostics .NET Agent into the Azure Infrastructure. Once deployed the .NET Agent monitors applications running in the Azure Cloud; collecting performance data and reporting to an HP Diagnostics Server for analysis and problem detection. Details and installation bits can be found with the Diagnostics packages in a special file named AzurePack_9.12.x.yyyyy.zip.

  Benefits: Functionality

- LDAP configuration needs a simple authentication type that requires a binding username and password instead of anonymous access, before normal LDAP user login (58734).

  Background: In versions prior to 9.12, the following features were NOT available:

  - LDAPLoginModule only supported anonymous searches of the directory, which opens a security hole.
  - LDAPLoginModule had a fixed filter format.

  - LDAPLoginModule had a fixed authentication DN.
  - LDAPLoginModule did not support search referrals.

  Description: In this release:

  - Users can now specify credentials to use for searching the directory.
  - Users can now specify any RFC 2254 compliant search filter.
  - LDAPLoginModule uses the DN returned from the search as the authentication DN.
  - Users can now specify that the LDAP server should refer search requests to other servers, if it cannot fulfill the search request.

  Benefits: Functionality, Security

- SaaS: Trim server requests after n path segments (default = 2) (64168).

  Server Request URIs can be trimmed after n path segments.

  Background: In prior versions of Diagnostics, URI lengths could be trimmed via the max length (maximum.uri.length) or pattern replacement (uri.pattern.replace).

  In certain cases (such as RESTful web services) it is useful to trim server request URIs based on the number of path segments, aggregating the data as well as reducing the number of server requests (which is limited by maximum.unique.uris).

  Description: A new property was added (maximum.uri.pathsegments) with a default value of -1 (no trimming) to capture.properties.

  The probe will trim excess fragments after applying maximum.uri.length & pattern.replace.

  For example, with a setting of 2, urls like /foo/bar/1, /foo/bar/2 will be trimmed to /foo/bar.

  Benefits: Functionality, Performance, Usability

- Diagnostics will feed data into BSM’s Service Health Analyzer.

  Description: New to the BSM 9.12 release is a new product called Service Health Analyzer (SHA). SHA is a system that can automatically detect anomalies by comparing a variety of metrics from different sources without setting thresholds on each metric. Diagnostics feeds a number of different metrics (probe and host metrics) into SHA to enable anomaly detection on Diagnostics data. See the SHA documentation with BSM 9.12 for details.

  Benefits: Functionality

- The Diagnostics collector can connect and extract metric from TIBCO Enterprise Message Service (EMS).

  Background: Prior to 9.12, Diagnostics was able to extract and display metrics from WebSphere Message Queue, but not TIBCO EMS.

Description: Starting with 9.12, the collector install contains a TIBCO EMS option. This new collector collects metrics very similar to WebSphere MQ. Metrics on servers, queues and topics are displayed and can be used to monitor and manage TIBCO EMS. See the *HP Diagnostics Installation and Configuration Guide* for details on configuring TIBCO EMS data collection.

Benefits: Functionality

- Diagnostics can display key TIBCO operational metrics along with other standard metrics.

  Background: Prior to 9.12, Diagnostics could only extract JMX attribute metrics. Many TIBCO metrics that are desirable for the user to see are operational metrics.

  Description: Starting with 9.12, Diagnostics can extract operational metrics. Out of the box, a number of key TIBCO JMX metrics are extracted and displayed: Active Process Count, Process Count, Threads, Uptime, Total Errors, New Errors, Uptime, Tibco Memory - Free Bytes, Percent Used, Total Bytes, Used Bytes. Further metrics desired by the user can be configured as necessary in the metrics.config file.

  Benefits: Functionality

- TIBCO ActiveMatrix Service Bus 3.1.2 users will see thread-switching server requests in one piece.

  Background: Prior to 9.12, Diagnostics supported TIBCO ActiveMatrix Service Bus 2.x only.

  Description: Starting with 9.12, Diagnostics supports TIBCO ActiveMatrix Service Bus 3.1.2 similar to the 2.x support, displaying Web Services including multi-threaded Web Services.

  Benefits: Functionality

- In some environments, the MQ collector will not collect metrics, fails to connect (62825).

  Background: In versions prior to 9.12, Diagnostics came with MQ API jar files included with the product. This saved a step for the customer to copy API files to a location needed by the collector. These jar files were from an older version of MQ, but have been working with newer MQ versions until just recently.

  Description: Starting with 9.12, the collector installer will prompt the user for the location of the IBM MQ jar files provided by IBM and will copy them to the location needed by the collector. This way, the collector is guaranteed to be using the correct, up to date API jar files.

  Benefits: Reliability

- Running a .NET Agent install session does not produce a log file on some systems (62247).

  Background: This will occur on Windows systems running Windows Installer versions earlier than version 4. In particular, Windows Server 2003 does not ship with Windows Installer version 4.

  Description: To create a log file, use the following syntax on a command line:

  ```
  <name of agent package>.msi /L*v <name of log file>
  ```

For example:

HPDiagTV.NetAgt_9.10.55.38542_win32.msi /L*v C:\mylog.txt

Benefits: Supportability

- Create an alert to warn about symbol table overflow conditions (61540).

  Background: Diagnostics Servers should recognize when the symbol table is going to overflow and block/limit the data coming in from a particular probe in order to lessen the impact on the server itself.

  Description: Diagnostics now provides a way to receive email notifications for conditions like too many server requests and running out of disk space. On the Alerting page in the UI under Maintenance > alerting, you can enter an email address for administration related notifications.

  Benefits: Supportability, Performance, Reliability

- Alerts Events not generated for Availability Metrics (60705).

  Background: Customer wants to see a change in the way we alarm. Here is the use case:

  On a particular probe, the availability threshold is set for 99%. Stop the JVM and wait for that threshold to be breached. It is currently @ 70% availability and there are no events listed under the Alert Events view and no email alerts received.

  If data is not received from a monitored component for a 5 minute period, the status goes from green -> yellow -> blank. Availability is calculated based on uptime in a 5 minute period. Blank status (no data) overrides threshold breaches (red alerts).

  So if the availability metric is needed to send an alert when the availability drops below 85%, the software could not send that alert because it would take more than 5 minutes for that threshold of 85% to be breached. Because it would take longer than 5 minutes, the status would go blank and no further alerting or thresholds would be measured because there is no more data being collected.

  There should be a "no-data override" where it can be specified that if no data is received within a 30 minute timeframe, that the status should go blank and an alert sent. The idea being that there are some metrics that need to be calculated based on more than 5 minutes of data not being received.

  In this particular case, the threshold is set to 99%, but there could be other thresholds that are configured which may require no data being collected for longer than 5 minutes to reach the proper level of breached thresholds and alerting.

  Description: A new flag in server.properties (threshold.evaluation.status.red.for.availability) now allows alerting on the availability metric. It requires a restart of the server.

  Benefits: Functionality

- Enhance CPU metrics in Solaris Zones to return CPU for that guest zone only and not the host system (60492).

Background: Prior to 9.12, Diagnostics displayed the CPU utilization for the host system and not the guest zone in a Solaris Zone environment.

Description: Now the CPU displayed is for the zone only and not the host.

Benefits: Performance, Reliability

- Add support for specific JMX "generational" heap metrics which are a complex JMX metric type (60336).

  Background: Prior to 9.12, Diagnostics could only collect “simple” JMX metric types. It would not extract the “generational” complex JMX metric types.

  Description: Composite Type mbeans are now supported. See the comments at the top of metrics.config for the updated grammar to select the fields from the composite type. Default Memory Pool metrics were added to support Hotspot, IBM, Jrockit, and Azul JVMs. If additional metrics are needed, see the comments in metrics.config under "Java Platform Metrics" OR uncomment the line "#default.dump.available.metrics = true" to get all supported metrics output to a file in the probe's log folder.

  Benefits: Functionality

- [JAX-WS-Outbound] instrumentation is very inefficient (47556).

  Background: In prior versions of Diagnostics, the JAX-WS-Outbound instrumentation point is very inefficient, taking more CPU time than necessary.

  Description: The point and code snippet was redesigned to be more efficient.

  Benefits: Performance

- If only the total size of all instances of one certain class is >1000 bytes, it will then be shown in the heap dump table (64285).

  Background: In prior versions, not all classes were shown in the profiler, some were missing. By design, the heap dump did not report classes with less than 1000 bytes of total footprint. This was hard-coded.

  Description: A dynamic property was added to dynamic.properties, heapdump.class.bytes.min, with a default of 1000, which can be modified if the desire is to show small classes.

  Benefits: Functionality, Performance

- Enable correlation map monitoring and memory use reduction by default (63482).

  Background: Cross-VM correlation problems can lead to a mediator requiring a large amount of heap space and it can even cause out of memory problems. As of 9.10 there is a monitor that is disabled by default which can significantly alleviate this problem by cleaning up the map periodically.

Description: The monitor is now enabled by default in server.properties. The frequency and aggressiveness of the monitor can be configured dynamically in server.properties.

# Monitor the correlation maps for statistics and to age off old entries.

# This can prevent excessive memory usage and OutOfMemory on some mediators.

# Use frequency=0 to disable. Set frequency and age limit to enable.

# Age limit of zero allows for statistics to be tracked without doing aging.

correlation.map.monitor.frequency=5m

correlation.map.age.limit=1h

Benefits: Performance

- Include documentation on the Diagnostics Query API (63261).

  Description: A new Diagnostics Data Model API document (pdf format) is included in the Diagnostics 9.12 documentation library available by selecting Help from the Help menu in the Enterprise UI.

  Benefits: Functionality

Defect Fixes in 9.12

Diagnostics 9.12 has a number of defect fixes shown below. Note that the defect tracking number shown (for example 35266) is generally prefixed with QCCR1I.

Java Agent

- 63487- Java probe crashes Glassfish 3.1

  Problem: When instrumenting Glassfish 3.1, the following error occurs:

  ```
  [#|2011-09-14T14:44:46.895-0400|SEVERE|glassfish3.1|
  javax.enterprise.system.core.com.sun.enterprise.v3.server |_ThreadID=1;_
  ThreadName=Thread-5;|com/mercury/opal/capture/proxy/MethodCaptureProxy
  ```

  Resolution: This issue can only be solved by modifying Glassfish configuration files. This is documented in the installation manual and repeated here for your convenience:

  Edit files:

  <as-install>/osgi/equinox/configuration/config.ini

  <as-install>/osgi/felix/conf/config.properties

  and add

  com.mercury.opal.capture.proxy

  to the property

  org.osgi.framework.bootdelegation

in both files.

- 61977 - Consumer ID extraction is not working correctly

  Problem: During service request execution, when using the ProbeSOAPHandler, the consumer id is extracted before the web service is recognized. However, the web service name is stuck to the thread, and therefore will be incorrectly used during customer id capture for the NEXT server request.

  Resolution: Using web service name provided by SOAP Handler. This is independent from our web service name detection, but it should always match.

- 30682 - Java Probe - Aggregating negative values into AggregateTimedRecord due to an inconsistent native timer

  Problem: SEVERE error below is seen in the java probe log file:

  ```
  2009-05-13 11:56:39,622 SEVERE common [Buffer write thread] Aggregating negative values into AggregateTimedRecord due to an inconsistent native timer,nodeData [latency (MICROSECONDS)] time [-180796] count [1] exceptionCount [0] totalTimeouts [0] min [-180796] max [-180796] sumTimesSquared [3.2687193616E10]

  java.lang.Exception

  at com.mercury.diagnostics.common.data.graph.impl_oo.AggregateTimedRecord.aggregate (AggregateTimedRecord.java:517)

  .

  .

  .

  at java.lang.Thread.run(Thread.java:619)
  ```

  Resolution: There is an ongoing effort to minimize the impact of “time drift” in virtual environments. Previous fixes and enhancements have been implemented to avoid these errors most of the time. However, they still occasionally occur. The current fix is to make this a WARNING instead of a SEVERE when it appears to be a small time drift. Also the negative numbers will be aggregated as zeros to minimize the impact after aggregation. There will still be a SEVERE log message if the time drift is larger.

.NET Agent

- 62589 - HP.MetricsAgent reports large negative network bytes/sec

  Problem: The .NET metrics agent on VMware guests occasionally report extremely large negative values for Network Bytes/Sec. Even though these values are in the MB/sec range, they are shown as Bytes/sec in the EUI.

  No errors were found in the .NET agent logs.

Resolution: Microsoft's signed 64-bit raw counters rollover at the unsigned 32-bit boundary on 32-bit machines. The code was changed to use different data types to hold the data.

- 63493 - .NET agent - any upgrade from 9.0 forward adds duplicate entries to probe_config.xml

  Problem: .NET agent - any upgrade from 9.0 forward adds duplicate entries to probe_config.xml. Does not affect runtime operation, but could cause customer confusion.

  Resolution: Duplicate entries to probe_config.xml have been eliminated.

Server

- 65181-Instrumentation errors in server.sh

  Problem: When starting the server using bin/server.sh, you get this:

  ```
  Running jreinstrumenter for /opt/MercuryDiagnostics/Server/jre/jre

  /opt/MercuryDiagnostics/Server/jre/jredoes not exist

  Sun 1.5.0_17 (/usr/java/jdk1.5.0_17/jre)

  mv: cannot stat `/opt/MercuryDiagnostics/Server1778/probe/classes/*/1.6.*': No such file or directory
  ```

  A check using the jreinstrumenter shows that the JVM supplied with and used to run the server has not been instrumented.

  On Windows, when starting the server with bin\server.cmd, this appears to work fine.

  Resolution: The script has been corrected for Linux and Solaris to properly instrument the server JRE.

- 62591 - After instant-on expires this message is logged every 30 seconds on a mediator (commander is licensed). WARNING license : Error getting license details (any): No license is found in license file.

  Problem: License checking should happen on the commander and not the mediators. The logs should not fill up with license messages.

  Resolution: Made the license checker only run on the commander and reduced the frequency of log messages.

- 62746 - Metric and threshold values in Diagnostics events sent to BSM look incorrect.

  Problem: Customer sees data that doesn't make any sense when they look at the reason for a status change in BSM.

  Resolution: Old information was cached that got used when new status changes occurred. Diagnostics was changed to create accurate information when status changed.

- 62670 - Alert Rules Not Displayed in Diagnostics UI

  Problem: In some situations, Diagnostics was not getting the list of alerts configured in the server under Alerts ->Alert Rules view.

  However, it is possible to create new alerts and also edit old one. Alerts are also generated.

  In the Diagnostics UI, there are no Alert rules visible, its blank.

  Resolution: Alert rules are now visible and correct in the UI.

- 64906 - No data in Scenario Summary View with .NET wrong IP Address detected

  Problem: No data in Scenario Summary View with .NET wrong IP Address detected

  Resolution: The server was unable to send start/stop run requests to the .NET probe because the detected IP address (due to something on the network) was not usable.

  Set use.host.for.addr=true in server.properties. This will allow the server to communicate start/stop runs with the .NET probe without using the incorrect auto-detected IP address. In this case, the customer was using the correct IP address as the hostname. The hostname would also work if the name resolves correctly from the server.

- 51524 - "BT to Application" dependency reporting to RTSM needs to be timelier in Diagnostics

  Problem: A Business Transaction being synthetically generated by BPM and monitored by Diagnostics probes was composed of 5 applications across multiple tiers (Java and .NET).

  After the Business Transaction executed for over 3 hours, only one of the applications (a .war) file was related in the RTSM although there were between 5-10 server requests in the scope of the business transactions across 3 different probes.

  Diagnostics had the correct data in its GUI but this data was not shown in the RTSM (Service Health, TM Topology).

  This problem happened for all 20 business transactions that were part of this application.

  NOTE: It was consistent that each BT only had one dependent application. After R&D performed a hard-sync the data was correctly recorded in BSM

  Resolution: In order to improve this situation, Diagnostics will perform a hard synchronization every 12 hours instead of every 24 hours.

What's New in 9.10

Diagnostics 9.10 has a number of defect fixes and new features, click the following links for details:

- 9.10 New Features
- 9.10 Defect Fixes

Platform Support Changes

- As of Diagnostics 9.00 the .NET Agent requires .NET Framework 2.0 or later. If you must support .NET Framework 1.1, use an earlier version of the .NET Agent (8.x) which will continue to be supported via patches and is compatible with later Diagnostics Servers 9.xx.
- Subsequent releases of Diagnostics after version 9.10 will NO LONGER support the following:
    - Servers and collectors on HP-UX operating systems

Notes on Integration with BSM

- Diagnostics 9.10 can be integrated with Business Availability Center 8.0x, Business Service Management 9.00/9.01 and 9.10.
- Diagnostics 9.10 integration with BSM 9.01/9.00 requires the following changes. These changes are not required when integrating Diagnostics 9.10 with BSM 9.10.
    - Install the OMI_00005 patch on the BSM 9.01 Gateway Server(s) and Data Processing Server(s). You can get this patch from the SSO patch site by selecting the Operations Manager i product and downloading the OMI_00005 patch.
    - After you complete installing Diagnostics 9.10, save a copy of the server.properties file. Stop the Diagnostics Server. In the <Diagnostics_Server_Install_Dir>/etc/server.properties file uncomment the “#pre-9.1#” prefix and comment out everything below it (see the example below). Then start the Diagnostics Server.

BEFORE

```
## Provides mapping support to allow renaming metric category names based specific metrics.
## ** Note this takes precedence over "bachi.cat.map" mapping below.
bachi.metric.map.HeapUsed|bytes|Probe = Heap
bachi.metric.map.HeapFree|bytes|Probe = Heap
bachi.metric.map.HeapTotal|bytes|Probe = Heap
bachi.metric.map.HeapUsedPct|percent|Probe = Heap
bachi.metric.map.Bytes\ Used|bytes|.NET = Heap
#pre-9.1#bachi.metric.map.CPU|percent|System = CPU
#pre-9.1#bachi.metric.map.ContextSwitchesPerSec|count|System = CPU
#pre-9.1#bachi.metric.map.MemoryUsage|percent|System = Memory
#pre-9.1#bachi.metric.map.VirtualMemoryUsage|percent|System = Memory
```

```
# To align with infrastructure content pack
bachi.metric.map.CPU|percent|System = CPULoad
bachi.metric.map.MemoryUsage|percent|System = MemoryUsageLevel
bachi.metric.map.VirtualMemoryUsage|percent|System = MemoryLoad
bachi.metric.map.ContextSwitchesPerSec|count|System = CPULoad
bachi.metric.map.DiskBytesPerSec|bytes|Disk = HostDiskUtilization
bachi.metric.map.DiskIOPerSec|count|Disk = HostDiskUtilization
bachi.metric.map.NetworkBytesPerSec|bytes|Network = InterfaceUtilization
bachi.metric.map.NetworkIOPerSec|count|Network = InterfaceUtilization
bachi.metric.map.PageInsPerSec|count|System = MemoryLoad
bachi.metric.map.PageOutsPerSec|count|System = MemoryLoad
```

AFTER

```
## Provides mapping support to allow renaming metric category names based specific metrics.
## ** Note this takes precedence over "bachi.cat.map" mapping below.
bachi.metric.map.HeapUsed|bytes|Probe = Heap
bachi.metric.map.HeapFree|bytes|Probe = Heap
bachi.metric.map.HeapTotal|bytes|Probe = Heap
bachi.metric.map.HeapUsedPct|percent|Probe = Heap
bachi.metric.map.Bytes\ Used|bytes|.NET = Heap
bachi.metric.map.CPU|percent|System = CPU
bachi.metric.map.ContextSwitchesPerSec|count|System = CPU
bachi.metric.map.MemoryUsage|percent|System = Memory
bachi.metric.map.VirtualMemoryUsage|percent|System = Memory
# To align with infrastructure content pack
#bachi.metric.map.CPU|percent|System = CPULoad
#bachi.metric.map.MemoryUsage|percent|System = MemoryUsageLevel
#bachi.metric.map.VirtualMemoryUsage|percent|System = MemoryLoad
#bachi.metric.map.ContextSwitchesPerSec|count|System = CPULoad
#bachi.metric.map.DiskBytesPerSec|bytes|Disk = HostDiskUtilization
#bachi.metric.map.DiskIOPerSec|count|Disk = HostDiskUtilization
#bachi.metric.map.NetworkBytesPerSec|bytes|Network = InterfaceUtilization
#bachi.metric.map.NetworkIOPerSec|count|Network = InterfaceUtilization
#bachi.metric.map.PageInsPerSec|count|System = MemoryLoad
#bachi.metric.map.PageOutsPerSec|count|System = MemoryLoad
```

- If a Diagnostics 9.01 installation is upgraded to a later version and that installation is integrated with BSM, then if you see issues with re-integrating with BSM you may have to do the following:
  - On the BSM server, find the HPOvSecCS*.msi file that was used to originally install HPOvSecCS (HPOvSecCS is the Certificate Server).
  - Execute ovc -stop.
  - Use the HPOvSecCS*.msi file to uninstall the Certificate Server.
  - Use the HPOvSecCS*.msi file to reinstall the Certificate Server.
  - Execute ovc -start.
  - Go to all systems that were granted certificates and perform the certificate request and grant steps again (see the HP Diagnostics Installation and Configuration Guide chapter on BSMIntegration).

New Features in 9.10

- Enhanced ability to create custom dashboards for use within BSM.

  Background: An effort is being made in every release to make Diagnostics easier to use. Past versions have had good Dashboard type capabilities. 9.10 brings new ways to create dashboards for use within BSM.

  Description: With 9.10, creating custom dashboards is made easier than in the past with more features. A Dashboard view exposes only the graphs and hides all controls to maximize the amount of information which can be placed on the screens. The new 9.10 Dashboard features include:

  - With any view in the UI, there is an action to save the view as a dashboard. If the user selects specific entities (vs. top N) then the dashboard is locked, otherwise it can be used to open on a selected entity (MyBSM and Open in Custom view).

- New MyBSM Diagnostics Dashboard available in BSM.

  Description: Users can associate select CI's with a saved dashboard and display the dashboard in "MyBSM" by selecting the CI under the "Diagnostics Dashboard" tab.

  Benefits: Usability, Functionality

- Improved licensing support.

  Background: Prior versions of Diagnostics supported a proprietary licensing scheme. Also, licenses did not distinguish between a probe running in "AD" mode (Application Deployment in a LoadRunner/Performance Center environment) and a probe running in "AM" mode (Application Management or Enterprise mode for production environments). This was an issue because in "AD" mode, it is common to have the probe installed on more nodes that will be active at any one time. For example, you might have 100 instances with the probe installed, but only 20 are active at any one time during a test run. In this situation, it makes sense to only pay for 20 instances.

  Description: Starting with 9.10, Diagnostics uses the HP standard "Autopass" licensing systems. Customers familiar with that system can obtain HP Diagnostics licenses now in the same fashion as all other Autopass products. The previous Diagnostics proprietary licenses will continue to work as well for backwards compatibility reasons.

  In addition, customers can now choose between "AD" and "AM" licenses or any combination of the two. As explained above, the advantage of an AD license is that the Diagnostics server only checks the count of licenses against probes that are active during a test run. However, these AD licenses cannot be used in production and will not count towards a probe that is configured to run in AM mode in a production environment.

  Benefits: Cost of Ownership, Usability

- Added support for .NET REST style web services over http.

  Description: In the .NET Agent WCF REST services are supported with a new keyword:

  keyword = REST

This keyword is added to the default out-of-the-box WCF.points file and so enabled by default in new installations.

REST services will follow the Web Service Model in Diagnostics and will show up as SOA service with similar properties (Namespace, Servicename, Operations) as Web Services. The Binding or WsPort is not relevant for REST services and hence omitted.

Benefits: Functionality

- Can configure the processing of data from mediators in batches.

  Background: The commander broadcasts a database export query to all mediators and collates the results. This collation can use up a lot of memory on the commander. Consider 60 mediators and 3000 probes: If the database export task runs a query like /probegroup/probe, the commander will end up with a result set from all 60 mediators for all 3000 probes before the results are exported to the database.

  Description: This enhancement will allow the user to set the processing of mediators in batches (e.g. get results for 10 mediators, export these results and process the next 10 mediators and so on). This configuration can be found in the property servers-per-query in the etc/data-export-config.xml configuration file.

  Benefits: Performance

- Can upgrade from BSM 8.x integrated with Diagnostics to BSM 9.10 integrated with Diagnostics.

  When integrating Diagnostics with BSM 9.10 you will notice the following changes in the integration. See the BSM Upgrade Guide for information on upgrading with a Diagnostics integration.

  - The diagnostics_probe CI is no longer used and is replaced with more granular application infrastructure CIs representing J2EE and .NET application servers, databases, MQ messaging systems, SAP instances and so on.
  - With Diagnostics 9.10 a new MyBSM Diagnostics Dashboard view is provided in BSM.
  - The Application KPI is no longer used for Diagnostics and is replaced by System and Software KPIs and specific Health Indicators based on Diagnostics metrics categories.
  - The Diagnostics view in Business Service Management is replaced by Diagnostics Probe Group and Infrastructure view.

  The Business Service Management upgrade will affect old Diagnostics data and configurations as follows:

  - The old diagnostics_probe CI that is no longer used will be deleted from the Run-time Service Model after 40 days.
  - The Application KPI that was associated with the diagnostics_probe CI will be deleted at upgrade.
  - Manual modifications on the Application KPI will be lost.

- LDAP configuration needs a simple authentication type that requires a binding username and password instead of anonymous access, before normal LDAP user login. 59925

  Background: Prior versions of Diagnostics have limited LDAP features.

Description:

Users can now specify credentials to use for searching the directory

User can now specify any FRC 2254 compliant search filter.

LDAPLoginModule uses the DN returned from the search as the authentication DN.

Users can now specify that the LDAP server should refer search requests to other servers, if it cannot fulfill the search request.

Benefits: Functionality, Security

- Add support for specific JMX "generational" heap metrics which are a complex JMX metric type. 61634

  Background: In prior versions of Diagnostics, only “simple” JMX metric data types could be extracted and displayed (integer, string, etc.).

  Description: Composite Type mbeans are now supported. See the comments at the top metrics.config for the updated grammar to select the fields from the composite type. Default Memory Pool metrics were added to support Hotspot, IBM, Jrockit, and Azul JVMs. If additional metrics are needed, see the comments in metrics.config under "Java Platform Metrics" OR uncomment the line "#default.dump.available.metrics = true" to get all supported metrics output to a file in the probe's log folder.

  Benefits: Functionality

- Diagnostics agents should not trim BPM/LR/PC requests based on latency. 61491

  Background: In prior versions of Diagnostics, the Java agents would trim all server requests based on the property minimum.fragment.latency in dispatcher.properties which is located under the agent's etc directory. This included server requests from Business Process Management (BPM) and LoadRunner/Performance Center (LR/PC). In most cases, by default, it is better not to trim any BPM or LR/PC server requests since customers typically would want to see all synthetic transactions.

  Description: A new property was added (minimum.synthetic.transaction.fragment.latency) with a default value of 0ms to dispatcher.properties. The agent was modified to use this value when the fragment is synthetic (from BPM or LR/PC) instead of the original minimum.fragment.latency.

  Benefits: Functionality, Performance, Usability

Defect Fixes in 9.10

Diagnostics 9.10 has a number of defect fixes shown below. Note that the defect tracking number shown (for example 35266) is generally prefixed with QCCR1I.

Enterprise User Interface

- 58687 - Only one metric is graphed instead of N metrics when Open in Custom View.

  Problem: When opening a view from a selected entity and choosing Open in Custom View, only a single metric is graphed instead of the N metrics in the Custom View.

  Resolution: A coding error prevented all the metrics from being properly graphed. They will now all be graphed correctly.

Java Agent

- 58984 - Instrumentation issues with Glassfish App Server.

  Problem: After installing a java agent on Solaris for monitoring a J2EE application running on Glassfish, installation did not return any errors. JREInstrumenter was run and the following lines were generated:

  ```
  "-Xbootclasspath/p:/opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent/classes/Sun/1.6.0_20/instr.jre"

  "-javaagent:/opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent/lib/probeagent.jar"
  ```

  After instrumenting the Glassfish App Server it caused the application server to crash. This was the error seen in the logs:

  ```
  INFO [com.mercury.opal.common.loader]: Loading modules.properties from:
  /opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent/lib/../lib/modules.properties

  HP Diagnostics J2EE Probe 'davilab_co_davivienda_com', version: 9.00.77.1123

  Could not load Logmanager "com.sun.enterprise.server.logging.ServerLogManager"

  java.lang.ClassNotFoundException: com.sun.enterprise.server.logging.ServerLogManager
  ```

  Cause: Glassfish uses -Djava.util.logging.manager=...ServerLogManager. This class is loadable only via the Glassfish application class loader. If the probe initializes quickly, it soon hits a standard Java class that loads java.util.logging.LogManager. This class tries to load ServerLogManager, as specified, upon its initialization. If Glassfish has not started yet, ServerLogManager will not be found, and Glassfish logging will be broken as a result.

  Resolution: The agent delays some of the agent activities hoping that Glassfish will have enough time to initialize itself and java.util.logging.LogManager.

- 61736 - ORA-00020: maximum number of processes (80) exceeded after Diagnostics Java instrumentation.

  Problem: After a certain amount of time, the application server that is being instrumented by the Diagnostics agent will abort with the following error condition:

  ```
  ORA-00020: maximum number of processes 80 exceeded

  Died during process startup with error 20 (seq=225)

  OPIRIP: Uncaught error 20. Error stack:
  ```

Cause: The problem occurs because the agent attempts to determine the type and version of the database. With this environment, the query fails. This normally would not cause an issue, but in this situation it is causing an issue with Oracle and process resources are not released.

Resolution: Because this is a corner case and very rarely occurs, a workaround is needed. Uncomment the following line from the capture.properties file:

db.collection.class.name = com.mercury.opal.capture.FallbackDBCollection

Collector

- 60990 - Interference between SQL Server Collector and VMware Collector.

  Problem: Running the SQL Server collector in the same JRE as the VMware collector may prevent the VMware's collector's data from displaying in the Diagnostic client.

  Cause: When the VMware metrics tree was placed in the default group and the SQL Server collector already created a synthetic group, the VMware metrics would be put into the synthetic group, which would cause them to be invisible.

  Resolution: Use the VMware hosts index as the root for VMware metrics, which prevents the mediator from getting confused and putting them into the SQL Server collector's synthetic group by mistake.

- 61420 - Extend the VMware Collector to support ESX directly vs. just VirtualCenter connections.

  Problem: The VMware collector currently works only at the VirtualCenter level. This means that metrics for entire "data centers" of servers are consumed by creating just one connection with the Collector. The collector needs to allow individual ESX Servers to be contacted (they too support the "/sdk" tools) which would allow customers to only pull data from a subset of hosts and guests.

  Cause: When the collector was changed to use more efficient CSV return values rather than XML, a side effect was that the collector no longer continued to work properly with ESX servers. It failed because ESX Servers encode their date-times using a different format than vCenters use:

  ESX Server Date-Time: 2011-07-07T17:16:20-06:00

  vCenter Date-Time: 2011-07-07T23:16:20

  Resolution: The code was modified to handle either date format

.NET Agent

- 62517- HP.MetricsAgent reports large negative network bytes/sec.

  Problem: The .NET metrics agent occasionally will report extremely large negative values for Network Bytes/Sec.

Cause: The rate counters on a 32 bit system (and OS) were wrapping unexpectedly at the UInt32.MaxValue = 4294967295. This caused the calculations for the difference between raw values after a counter wrap to go negative.

Resolution: Added code to check if the counter values are wrapping at different boundaries (64bit and 32bit) and adjusted the calculations to account for the wrapped values.

What's New in 9.02

Diagnostics 9.02 has a number of defect fixes and new features, click the following links to see details:

- 9.02 New Features
- 9.02 Defect Fixes

Platform Support Changes

- As of Diagnostics 9.00 the .NET Agent requires .NET Framework 2.0 or later. If you must support .NET Framework 1.1, use an earlier version of the .NET Agent (8.x) which will continue to be supported via patches and is compatible with later Diagnostics Servers 9.x.

Notes on Integration with BSM

- Diagnostics 9.02 can be integrated with Business Availability Center 8.0x and Business Service Management 9.00/9.01.
- Diagnostics 9.02 integration with BSM 9.01/9.00 requires installation of OMI_00005 patch on the BSM 9.01 Gateway Server(s) and Data Processing Server(s). You can get this patch from the SSO patch site by selecting the Operations Manager i product and downloading the OMI_00005 patch.

New Features in 9.02

- BSM's Run-time Service Model population of data has been enhanced to link database CI instances populated by Diagnostics with business transactions so you can see the complete transaction picture.

  Background: In versions prior to Diagnostics 9.02, only database CI instances are populated (SQL Server & Oracle) for databases discovered by the collector. There is no link from a business transaction to a database CI which results in an incomplete picture of the infrastructure the transaction runs on.

  Benefits: Functionality

- It is now possible to enable call profile collection for a specific amount of time and/or when certain conditions are met.

Background: In prior version of Diagnostics, call profiles are collected to show the minimum time of a server request, maximum time and average time. There was no method for enabling addition call profile collection.

Description: On demand call profile collection rules can be based on various conditions and combinations of conditions. For example:

- When the latency value meets a condition (such as exceeds or falls below) based on a given value
- When CPU usage meets a condition (such as exceeds or falls below) based on a given value
- When the HTTP status meets a condition (such as equals or not equals) a given value
- When the consumer ID equals, contains or matches a given value

By creating On Demand Call Profile rules, it is now possible to narrow in on the problem area and then use the call profiles that were collected to further debug the performance or availability issue in your application.

Benefits: Functionality

- Enhanced ability to create custom dashboards.

  Background: An effort is being made in every release to make Diagnostics easier to use. Past versions have had good Dashboard type capabilities. 9.02 brings new ways to create a Dashboard. The new dashboard can be used within Diagnostics.

  Description: With 9.02, creating custom dashboards is made easier than in the past with more features. A Dashboard view exposes only the graphs and hides all controls to maximize the amount of information which can be placed on the screens. The new Dashboard features include:

  - Dashboards can be created from a snapshot, which creates a dashboard with fixed entity metric pairs (cannot be applied to other entities).
  - With any view in the UI, there is an action to save the view as a dashboard. If the user selects specific entities (vs. top N) then the dashboard is locked, otherwise it can be used to open on a selected entity (Open in Custom view).

  Benefits: Usability, Functionality

- TIBCO BusinessWorks 5.9 and Active Matrix Service Bus (AMSB) 2.3 user’s can see SOA Service Topology, Server Request Topology and end-to-end latencies (with call profiler) for SOAP/HTTP and SOAP/JMS web services in Diagnostics.

  Background: TIBCO BusinessWorks and AMSB use multiple threads to handle individual server requests. Previous versions of Diagnostics did not correlate these threads into a single server request, treating them each as a separate, uncorrelated, server request.

  Description: Diagnostics now correlates these threads into a single server request, providing the user with the end-to-end latency, Call Profile and Server Request Topologies. In addition, the SOA Service Topology is now correctly populated.

  Benefits: Functionality

- Can distinguish two or more .NET Application Domains (AppDomains) with the same name on the same IIS server.

Background: It is common to have multiple virtual directories (AppDomains) with the same name (for example “CallChain”). In previous versions of the .NET Agent the probe_config.xml extract to enable the CallChain AppDomain looked like this:

```
<appdomain name=”CallChain”>

    <points file=”CallChain.points” />

<\appdomain>
```

The configuration format does not provide a way to distinguish the distinctly configured CallChain applications in IIS. The result of this was that that you had to enable all (CallChain AppDomains) or disable all. This is not a desirable level of control to enable/disable Web Applications. More so, the probes instantiated by all the CallChain AppDomains were named exactly the same and so the server requests from these applications got attributed to the single consolidated probe on the server. The probe metrics were also averaged over completely independent instances which would be incorrect data presented to the user.

Description: The primary purpose of this feature is to be able to distinguish two or more AppDomains on the same IIS server which have the same name. This feature changes the format for ASP.NET (IIS) AppDomain configuration format in the probe_config.xml file to enable distinguishing the AppDomains by qualifying each with the IIS configuration path. The new configuration for three CallChain applications would be as follows:

```
<appdomain enabled="false" name="1/ROOT/CallChain/CallChain" website="Default Web
Site">
    <points file="Default Web Site-CallChain-CallChain.points" />
</appdomain>


<appdomain enabled="false" name="1/ROOT/CallChain" website="Default Web Site">
    <points file="Default Web Site-CallChain.points" />
</appdomain>


<appdomain enabled="false" name="2/ROOT/CallChain" website="WebSite2">
    <points file="WebSite2-CallChain.points" />
</appdomain>
```

The resultant probes are now distinguished using the IIS path and show up in the Enterprise UI uniquely.

Benefits: Functionality, Usability

- A new collector has been added that connects to and gathers metrics from VMware environments.

  Description: Collects data from vSphere and vCenter servers and displays data in two new views: VMware Hosts view and VMware Guests view. Sideways drills exist that allow you to jump between the two views in context.

  Benefits: Functionality

- Added support for .NET REST style web services over http.

  Description: In the .NET Agent WCF REST services are supported with a new keyword:

  keyword = REST

This keyword is added to the default out-of-the-box WCF.points file and so enabled by default in new installations.

REST services will follow the Web Service Model in Diagnostics and will show up as SOA service with similar properties (Namespace, Servicename, Operations) as Web Services. The Binding or WsPort is not relevant for REST services and hence omitted.

Benefits: Functionality

- Can configure the processing of data from mediators in batches.

  Background: The commander broadcasts a database export query to all mediators and collates the results. This collation can use up a lot of memory on the commander. Consider 60 mediators and 3000 probes: If the database export task runs a query like /probegroup/probe, the commander will end up with a result set from all 60 mediators for all 3000 probes before the results are exported to the database.

  Description: This enhancement will allow the user to set the processing of mediators in batches (e.g. get results for 10 mediators, export these results and process the next 10 mediators and so on). This configuration can be found in the property servers-per-query in the etc/data-export-config.xml configuration file.

  Benefits: Performance

Defect Fixes in 9.02

Diagnostics 9.02 has a number of defect fixes and a number of new features shown below. Note that the defect tracking number shown (for example 35266) is generally prefixed with QCCR1I.

Server

- 59785 – Discovery 7d NodeDataQueries taking minutes to run and for discover BizTxn, the query is run for each application.

  Problem: Some discovery queries were taking too long – Discovery of business transactions was using too many queries to the server.

  Resolution: Discovery queries were optimized to run a lot faster.

- 60468 – SMTP events – alert stays open even when an all clear alert is received.

  Problem: When an alert is triggered a notification is received, however, when the condition cleared out there is no clear alert event, or an OK notification.

  Resolution: A new alert option was introduced that allows the user to enable/disable sending of SMTP/SNMP alerts when the state changes from red/yellow to green.

  This option is enabled by default for all new alerts.

- 51729 - Server Install fails - Permission denied.

  Problem: Server Install fails - Permission denied errors:

  sh: /opt/HPDiagnostics/bin/runscript.sh: cannot open [Permission denied] sh: /opt/HPDiagnostics/bin/runscript.sh: cannot open [Permission denied]

sh: /opt/HPDiagnostics/bin/runscript.sh: cannot open [Permission denied]

Resolution: This release does not support installing as non-root. With this fix a non-root install will work as long as the event agent (a.k.a. opc agent or OMi agent) is not selected. When installing Diagnostics and selecting integration with BSM, a non-root installer must not select the agent installation. The OMi agent can be installed later, as root.

- 40372 - Server - can't find 'Throughput' metrics after the metrics data export to external database.

  Problem: After the metrics data was exported to an Oracle database, the 'Throughput' metrics are not present in the export.

  Resolution: The "latency" record has additional information in it which includes CPU total and count, timeouts, exceptions and throughput (calculated). None of these additional metrics were exported.

  Diagnostics 9.02 has added the following metrics in the category "latency":

  - total_cpu
  - exception_count
  - timeout_count
  - throughput

  These metrics will be exported as if they were separate metrics/records. Since the min/max is not tracked for these, min/max will be set to 0.

- 51315 - Data Export not exporting accurate 'min' & 'max' values for JDBC metrics.

  Problem: Data export of probe metrics to Oracle for metrics:

  > WebSphere6/EXPAND_PMI[*]/connectionPoolModule.UseTime = JDBC Use Time|milliseconds|JDBC Connection Pools

  results in the Min and Max values being identical and do not appear to reflect correct data. Data Exporter should export: Average (for time Period), Min Value and Max value.

  Resolution: The minimum and maximum data exported does not currently allow for decimal places. The server code has been enhanced to use double to preserve the data precision. The new schema will use database columns of type float to preserve the precision.

  If the customer upgrades to 9.02 Diagnostics and continues to use the existing database from a prior version, then the columns can be modified using the following DDL. This may affect any of the customer’s data consumers.

  Oracle:

  ```
  ALTER TABLE RECORD MODIFY (
    REC_COUNT NUMBER(38),
    TOTAL FLOAT,
    MINIMUM FLOAT,
    MAXIMUM FLOAT)
  ```

SQL Server:

ALTER TABLE RECORD ALTER COLUMN REC_COUNT DECIMAL(19)

ALTER TABLE RECORD ALTER COLUMN TOTAL FLOAT

ALTER TABLE RECORD ALTER COLUMN MINIMUM FLOAT

ALTER TABLE RECORD ALTER COLUMN MAXIMUM FLOAT

- 52765 - The priority that the data retention policy has over purging is not clearly documented and discussed in the Installation Guide.

  Problem: Despite having specified a particular purging threshold e.g.
  persistence.purging.threshold = 5G

  the actual disk storage used is exceeding this amount. In these circumstances, messages of the following form may be found in the server.log:

  yyyy-mm-dd hh:mm:ss,ms: WARNING archive : The archive size threshold (5368709120) has been exceeded by 35455152128 bytes to honor minimum purging policies and data preservation. Try increasing the persistence.purging.threshold property, deleting some Incidents, or adding additional servers. (logged only once)

  In some case the available disk storage space is exhausted.

  The priority that the data retention policy has over purging is not clearly documented and discussed in the Installation Guide.

  Resolution: Updated the Diagnostics Install Guide, Appendix E Diagnostics Data Management to discuss the affect of data retention settings on the purging mechanism including adding the following:

  If you find that disk space is being exhausted this does not mean that purging isn't working it may mean that one of the following factors has affected the purging mechanism. For example if you have allocated 10GB of disk space on the server for Diagnostics but you see the archive at 20GB in danger of exhausting disk space on the system, this could be possible for any of the following reasons:

  - Purging interval has not been reached yet. You can adjust to a shorter interval.
  - There are a large number of snapshots on the system that by design do not get purged.
  - The data retention settings may be requiring too much data be retained. You may need to adjust data retention in order to save disk space.

- 56717 - Diagnostics fail to publish topology to ODB.

  Problem: Diagnostics fail to publish topology to ODB. The metrics for uCMDB Relay failed in Diagnostics monitor page.

  Resolution: The HTTPS communication link from Diagnostics to the uCMDB has been fixed.

Integrations

- 52206 - LoadRunner 11 Controller crashes after J2EE Diagnostics addin is configured on non-English OS

Problem: When installing the Diagnostics LoadRunner Addin the following message box with error appears and the Controller crashes:

"Runtime error. Program wlrun.exe. Abnormal program termination."

Resolution: An internal error occurred and has been corrected. The crash will no longer occur.

- 52144 - Admin->Diagnostics shows fields for Gateway and DPS when it used to show 2 Gateways-- add descriptive type to registrar components page.

  Problem: The Diagnostics 9 server refers to BAC core and center servers, but the BAC UI refers to Gateway and DPS servers.

  Resolution: The registrar pages use internal terms. To help clarify, the type column now shows "BSM GW/Core" and "BSM DPS/Center".

Java Agent

- 59397 – SAP RFC instrumentation not working.

  Problem: Instrumenting a WebLogic 10.3.1 that uses a (customized) version 2.1.10 of the SAP JCO libraries to talk to SAP R3. Capturing JCO$Client.execute methods works (after changing the signature wildcard in the point) and extracting the RFC call name from the arguments works. However, the need is to trend the individual RFC calls, i.e. show them in the outbound calls view, report & alert on them, etc. The default [RFC] point never gets hit.

  Resolution: Added 2 instrumentation points subclassing the original main instrumentation point. These 2 points apply for the SAP JCO 2.1.10 (or later) and JCO 2.1.9 (or earlier).

- 58553 – Enable “class.name.replace=s s#\\$Proxy[0-9]+$#$Proxy#" by default to avoid symbol table explosion.

  Problem: Server generates hundreds of java_pidxx.hprof files taking up GBs of disk space from symbol table explosion due to issue with Java probe.

  Resolution: Enabled the class name replacement algorithm by default to avoid symbol table explosion.

- 59787 – Missing SQL Statements in Diagnostics 9.01 for Class OraclePreparedStatement ExecuteQuery on Oracle WebLogic 11g ADF.

  Problem: A regression occurred starting with Diagnostics version 9.01. Some oracle.jdbc.* do not get instrumented anymore.

  The regression was a side effect of optimizing (i.e. reducing the overhead of) our instrumentation where some duplicate class parsing was eliminated after noticing that the agent visited some classes more than once. Unfortunately, this caused the probe to miss some classes completely (WARN: No method signatures found ...). This happens for classes in oracle.jdbc.* packages, because they were excluded from "preprocessing".

  Resolution: Removed oracle.jdbc.* classes from the prevent.class.preloading.list property.

- 59794 – SOAP/HTTP Nested outbound calls not recognized.

Problem: For SOAP/HTTP Nested outbound calls, the outbound calls view showed the HTTP method instead of the SOAP Service operation. This is because, by default, the last outbound call is the one reported.

Resolution: Code was updated to prioritize the SOAP Service operation as the outbound call when followed by an HTTP outbound call.

- 51725 - Massive SEVERE errors in probe.log with IBM Java 1.4.2 64-bit.

  Problem: Multiple errors in probe.log (examples follow) and so the probe does not report any data:

  ```
  2010-10-14 13:26:37,523 SEVERE capture.metrics [Metrics Collection] Unknown error
  collecting metrics for Mercury System, initialized: true [Mercury System] ->
  com/mercury/diagnostics/capture/jni/VmwareJNI.getHostTimestampUsecs()J
  java.lang.UnsatisfiedLinkError:
  com/mercury/diagnostics/capture/jni/VmwareJNI.getHostTimestampUsecs()J
  at com.mercury.opal.util.VmwareTimer.getTimestamp(VmwareTimer.java:66)
  at com.mercury.opal.util.Timestamp.timestamp(Timestamp.java:345)
  at com.mercury.opal.util.TimerDriftMetric.doubleValue(TimerDriftMetric.java:57)
  at com.mercury.opal.util.TimerDriftMetric.getValue(TimerDriftMetric.java:46)
  at com.mercury.diagnostics.common.metrics.MetricCollector.collect(MetricCollector.java:156)
  at com.mercury.diagnostics.capture.metrics.CollectorAgent.run(CollectorAgent.java:749)
  at java.lang.Thread.run(Thread.java:770)
  2010-10-14 13:26:43,065 SEVERE
  com.mercury.diagnostics.common.util.InfrequentEventScheduler
  [shared InfrequentEventScheduler] While running rare event:
  com.mercury.opal.util.Timestamp$1@404e404e
  java.lang.UnsatisfiedLinkError:
  com/mercury/diagnostics/capture/jni/VmwareJNI.getHostTimestampUsecs()J
  at com.mercury.opal.util.VmwareTimer.getTimestamp(VmwareTimer.java:66)
  at com.mercury.opal.util.Timestamp.timestamp(Timestamp.java:345)
  at com.mercury.opal.util.Timestamp$1.timeDriftEstimate(Timestamp.java:1214)
  at com.mercury.opal.util.Timestamp$1.run(Timestamp.java:1201)
  at
  com.mercury.diagnostics.common.util.InfrequentEventScheduler$Event.run(InfrequentEventScheduler.java:319)
  at
  com.mercury.diagnostics.common.util.InfrequentEventScheduler.runThisEvent(InfrequentEventScheduler.java:620
  at
  com.mercury.diagnostics.common.util.InfrequentEventScheduler.runEvents(InfrequentEventScheduler.java:598)
  at
  com.mercury.diagnostics.common.util.InfrequentEventScheduler.access$5(InfrequentEventScheduler.java)
  at
  com.mercury.diagnostics.common.util.InfrequentEventScheduler$BackgroundThread.run(InfrequentEventSchedule
  ```

  Resolution: A coding error prevented the probe from recognizing that the platform was 64-bit. The product will now recognize it as 64 bit.

- 53383 - detail=args:0 in the point definition turns into an error message.

  Problem: Every time the value is set to 0 (detail=args:0), the following message appears:

  ```
  2010-11-08 16:21:20,214 WARN capture.inst [main] Malformed detail '0' argument number out
  of range.
  (Must be integer >= 1)
  ```

```
2010-11-08 16:21:20,214 WARN class
com.mercury.opal.capture.inst.compile.CodeSnippetCompiler
[main] Ignoring code snippet '#arg-1.toString()'
com.mercury.opal.capture.inst.compile.ParseException: Encountered "." at line 1, column 7.
```

Resolution: Works as expected.

- 56225 - Probe Metrics not available for JBoss Application Server.

  Problem: JBoss JMX metrics are missing in the Profiler and the end user UI. Errors in the probe log indicate an error of some sort when connecting to the JBoss JMX Metric Server MBean.

  Resolution: Changed the jar files delivered with the probe to eliminate class duplicates.

- 56287 - Java probe attempts to open incorrect directory on Turkish system - system metrics are not generated.

  Problem: Java probe attempts to open incorrect directory on Turkish system - system metrics are not generated.

  Resolution: The probe was attempting to open the wrong directory (ppc-aýx instead of ppc-aix). A change to the code which opens the directory resulted in the correct directory (ppc-aix) being opened.

- 57326 - Sun 1.5 JVM crashes with Assertion `(*env)->GetDirectBufferCapacity(env, data) >= (offset + length)' failed.

  Problem: Sun 1.5 JVM crashes with Assertion `(*env)->GetDirectBufferCapacity(env, data) >= (offset + length)' failed.

  Resolution: Modified code to avoid the JVM crash.

- 58226 - Instrumentation issues with Glassfish App Server.

  Problem: There are some issues with metrics collection with Glassfish App Server. After instrumenting the Glassfish App Server it caused the APP server to crash. This was the error seen in the logs:

  ```
  INFO [com.mercury.opal.common.loader]: Loading modules.properties from:
  /opt/MercuryDiagnostics/JavaAgent/DiagnosticsAgent/lib/../lib/modules.properties

  HP Diagnostics J2EE Probe 'xxxxxxxx', version: 9.00.77.1123

  Could not load Logmanager "com.sun.enterprise.server.logging.ServerLogManager"

  java.lang.ClassNotFoundException: com.sun.enterprise.server.logging.ServerLogManager
  ```

  Resolution: Changed the probe code to delay its initialization, giving time for Glassfish and Java login to initialize without interference.

.NET Agent

- 59321 - A feature in Diagnostics .NET agent to limit log folder size does not work.

  Problem: .NET generates huge logfiles - even after adding the following to the probe_config.xml file:

  ```
  <logdirmgr enabled="true" maxdirsize="200 MB" scaninterval="20m"/>
  ```

Some servers have log directories over 1 GB and some are approaching 2 GB.

Resolution: The code that attempted to limit the logfile size not work and failed silently. Logic problems with the code were fixed. There is yet an issue with the delete permissions to the log folder. The customer will need to modify the permissions on the log folder so that the .NET process Account (IIS App pool Account) has permissions to delete files in this folder.

- 58912 - .NET agent causes java.lang.NumberFormatException on Diagnostics server when sending Probe Metrics in Danish locale.

  Problem: HP Diagnostics .Net Agent 9.00.77.34620 on Windows 2003 Server with locale set to Danish regional settings.

  Server log reports:

  ```
  2011-04-07 12:44:12,771: SEVERE athDispatcherHandler: unhandled exception processing
  HTTP request for POST
  /metricdata/?sleep=false&probe=<servername>.1302170136733284&probehost=<servername>.dk
  HTTP/1.1: java.lang.NumberFormatException: For input string: "0,002377253"
  java.lang.NumberFormatException: For input string: "0,002377253"
  at sun.misc.FloatingDecimal.readJavaFormatString(Unknown Source)
  ```

  Probe log reports:

  ```
  2011.04.07.12.44.12.140 [0012] SEVERE ProbeMetrics ProbeMetricsAgent(5896758) caught
  exception when sending metrics to
  uri(<http://10.172.4.112:2006/metricdata/?sleep=false&probe=<servername>.1302167756125077&probeh
  ost=<servername>>)
  System.Net.WebException: The remote server returned an error: (500) Internal Server Error.
  at System.Net.HttpWebRequest.GetResponse()
  at Mercury.Capture.ProbeMetricsAgent.Request()
  ```

  Resolution: Changed the .NET agent to report number with CultureInvariant format.

- 56526 - Diagnostics Commander taking very long time to startup OR not starting at all in .NET environment with a large number of unique SQL statements.

  Problem: Diagnostics Commander taking very long time to startup OR not starting at all in .NET environment with a large number of unique SQL statements.

  Resolution: Cause - the cause of the problem is that the .NET agent and server do not merge SQL statements that are unique. For example, the Java agent will take the following SQL statements:

  1. select max(toAccount) from Relations where Relationship = 'R' and FromAccount = 23489234

  2. select max(toAccount) from Relations where Relationship = 'A' and FromAccount = 23492348

  3. select max(toAccount) from Relations where Relationship = 'B' and FromAccount = 12985712

  And merge them into a single statement:

- select max(toAccount) from Relations where Relationship = ? and FromAccount = ?

  The solution is to enhance the .NET agent and server to work the same way as the Java agent.

Enterprise User Interface

- 59831 - Profiler UI download creates multiple copies of the same jar for the same version under %TEMP%.

  Problem: When downloading the UI jars files (client or probe profiler UI) only the jar file names will be persisted to the client cache in %TEMP% instead of the complete path of the URL from where it was downloaded. Before this fix, the jar name contained the probe name which increases the number of downloads and wastes storage under %TEMP%.

  Resolution: The code no longer downloads the same version multiple times.

- 59783 - Network bytes in the range of GB per second on AIX.

  Problem: In the Diagnostics UI -> Standard View - Hosts, sometimes the information for the Network bytes utilization shows GB per second, which does not seem a feasible metric value.

  Resolution: Network I/O KBytes/Second shows the total network bytes of all network interfaces as well as the loopback interface, which cannot be measured externally. The benchmark for the Diagnostic UI is that it needs to match what the OS reports as the total network I/O bytes (some OSes include loopback and others don't).

- 59173 - Cannot reset Availability Threshold after setting to "No Threshold".

  Problem: After setting the Probe Availability threshold to "No Threshold", you cannot go back and reset the Availability threshold for that probe.

  Resolution: Corrected dialog creation so no NPE will occur.

- 59046 - Even though `Availability' metric is showing in the inspector (details pane), it does not turn yellow/red.

  Problem: - Even though `Availability' metric is showing in the inspector (details pane), it does not turn yellow in the inspector window correctly.

  Resolution: The metric will properly show yellow in the inspector window.

- 58688 - Go to any view and select multiple metrics, then leave the view and come back. The graphs are sometimes messed up.

  Problem: - Go to any view and select multiple metrics, then leave the view and come back. The graphs are sometimes messed up.

  Resolution: Internal logic error was corrected.

- 53875 - Want to share a view between applications.

  Problem: Want to share a view between applications. When a view is created inside application A, it should be easy to share it with all other applications or specific applications so when connecting to other applications, it's easy to run the same view.

Resolution: Regardless of the application in which the application is created, as long as specific entities are not selected (topN is not my Selections) this change will allow a view to be copied and used between applications.

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

  Resolution: 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.

  Workaround: 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 9.02.

System Metrics

- 59122 - Network metric values are too high on SunOS on VMware.

  Problem: Network metric values are consistently too high when SunOS is running under VMware.

  Resolution: Modified algorithm to take into account the virtual environment to return correct metric values.

What's New in 9.01

Diagnostics 9.01 has a number of defect fixes and new features, click the following links for details:

- 9.01 New Features
- 9.01 Defect Fixes

Platform Support Changes

As of Diagnostics 9.00 the .NET Agent requires .NET Framework 2.0 or later. If you must support .NET Framework 1.1, use an earlier version of the .NET Agent (8.x) which will continue to be supported via patches and is compatible with later Diagnostics Servers 9.xx.

New Features in 9.01

- .NET Agent - Support multiple application domains with the same name and distinguish them so that the data from them does not get aggregated together. (42733)

  Background: Change the format for the ASP.NET AppDomains which were previously not qualified with the IIS Path.

  Description:

  Old Probe_config.xml Configuration

  ```
  <appdomain name="CallChain">
  <points file="CallChain.points" />
  </appdomain>
  ```

  The configuration format does not provide a way to distinguish the 3 distinctly configured CallChain applications in IIS. The result of this was that you had to enable all (CallChain AppDomains) or disable all. This is not a desirable level of control to enable\disable Web Applications. More so, the probes instantiated by all the CallChain AppDomains were named exactly the same and so the server requests from these applications got attributed to the single consolidated probe on the server. The probe metrics were also averaged over 3 completely independent instances which would be incorrect data presented to the user.

  This feature changes the format for ASP.NET(IIS) AppDomain configuration format in the probe_config.xml file to enable distinguishing the AppDomains by qualifying it with the IIS configuration path.

  The new configuration for the 3 CallChain applications would be as follows:

  NEW Probe_config.xml Configuration

```
<appdomain enabled="false" name="1/ROOT/CallChain/CallChain" website="Default Web
Site">
<points file="Default Web Site-CallChain-CallChain.points" />
</appdomain>
<appdomain enabled="false" name="1/ROOT/CallChain" website="Default Web Site">
<points file="Default Web Site-CallChain.points" />
</appdomain>
<appdomain enabled="false" name="2/ROOT/CallChain" website="WebSite2">
<points file="WebSite2-CallChain.points
```

Benefits: Functionality

- Integrations and Enterprise UI - Enhance Diagnostics from within BSM to include the Help links. (48575)

  Background: The help menu including "Help on the View" is disabled when bringing up Diagnostics UI from within BSM. This means users can only access Diagnostics help from the BSM menu and they lose the context sensitive help links from the Diagnostics UI.

  Description: Enabled the help menu links in the Diagnostics UI when bringing up Diagnostics from within BSM.

  Benefits: Usability.

Defect Fixes in 9.01

Diagnostics 9.01 has a number of defect fixes and a number of new features shown below. Note that the defect tracking number shown (for example 35266) is generally prefixed with QCCR1I.

Server

- 47433 - IllegalStateException during OMi registration (BSM/Diagnostics Registration) in log file.

  Problem: This error gets logged when doing registration of BSM/Diagnostics but it does not indicate a real problem. A bad code path was doing a send with an event channel integration error (along with a successful BAC registration), but then continuing to try to ALSO "sendSuccess". Doing two responses to one request was the problem.

  Resolution: The unnecessary 2nd send was the one that was failing. Now it is avoided.

- 48066 - The discovery that adds probes/server requests/hosts to an Application when a Business Transaction is added to the Application only works for probes (SRs/hosts) that are not directly connected to the commander.

  Problem: The discovery that adds probes/server requests/hosts to an Application when a Business Transaction is added to the Application only works for probes (SRs/hosts) that are not directly connected to the commander.

  Resolution: The new feature of automatically adding entities to apps will now work on probes directly connected to the commander.

- 47650 - Newly created Business Transactions briefly show a latency of zero with big numbers for min/max.

Problem: Immediately after a Business Transaction is created via the Diagnostics UI or BSM it shows up in Diagnostics with latency of 0. The latency looks strange because it shows zero values and very high/low numbers for min/max. After a few seconds the latency numbers go away (until real numbers come in).

Resolution: The Business Transaction will only show latency now when an actual value is present and ready to show.

Java Agent

- 48656 - Getting Mismatch Serialization UID.

  Problem: When adding an instrumentation point with regular expression including all classes and methods, client application was failing with mismatch serialization error. Specifying a specific class and method avoided the problem..

  Resolution: To avoid application crashes we now do not add method wrappers for serializable classes. At the same time the user gets precise control over where the wrappers will be added.

- 48962 - Probe must not override good information with "Unknown".

  Problem: Customers sometimes see "Unknown" instead of the actual web service fields (for example they may see the web service operation as "Unknown").

  Resolution: The probe now reports the best information it can. If non-Unknown information was captured at some point in the chain, then we will use it.

- 49374 - For a single outbound call, the Outbound calls screen can show multiple entries.

  Problem:For a single outbound call, the Outbound calls screen can show multiple entries. This was in the case of nested outbound calls, i.e. when an outbound instrumentation point was hit multiple times consecutively.

  Resolution: Fixed the code so that only one entry is shown.

- 30602 - Wrong root method name is shown for the call profile of a cross-VM callee side server request.

  Problem: Wrong root method name "Outbound Call" is shown for the call profile of a cross-VM callee side server request.

  Resolution: The right root method name will now be shown.

- 49460 - Service topology does not connect the SOAP over JMS web services.

  Problem: SOAP over JMS Web Services were not connected in the Service topology screen.

  Resolution: The code problem has been fixed and now they are connected.

.NET Agent

- 33682 - CLR Issue with instrumentation (fixed for CLR 4.0 only) - Hp.Profiler.dll is loaded into applications not configured for monitoring.

Problem: The .NET agent inspects all .NET applications which come up for instrumentation by verifying scope defined in the probe_config.xml. This is how the .NET profiler works; it invokes the agent for every .NET application instantiated. Some applications (read executables) are fleeting, for example the Visual Basic compiler and C# compiler which get triggered in certain deployment scenarios in IIS. We are unable to handle the fleeting nature of such applications as they exit even before we have a chance to verify scope. Also there are applications like the Visual Basic compiler (mainly Microsoft binaries) which do not make sense to try and instrument or verify whether they are in scope. Verifying scope adds considerable start-up overhead. The .NET agent should maintain a list of such applications (preferably configurable) so that we recognize the futility of trying to instrument these processes right at the start and ignore them.

Resolution: Changed the return value for Initialize for CLR\.NET V4 to CORPROF_E_PROFILER_CANCEL_ACTIVATION for Applications which are NOT configured to be monitored. This will release the HP.Profiler.dll from the process memory. This is not available with earlier version of the profiler API hence cannot be solved for CLR\.NET V2.

Collector

- 47933 - Diagnostics Collector - creates/updates node CI with wrong (unqualified) primary DNS name attribute.

  Problem: The DB Collectors were sometimes reporting the unqualified DB host names to the Diagnostics Server (and hence the node CI was created with the unqualified name).

  Resolution: Added comments to the configuration files of both DB Collectors and the HP Diagnostics Installation and Configuration Guide to instruct the user to use the fully qualified host name when configuring the DB Collectors.

Enterprise UI

- 49580 - Service Health 360 View Drill to Probe Group Summary opens with incorrect probe group context.

  Problem: Diagnostics Probe Group Summary opens in context of probe group that was not selected in the 360 view.

  Resolution: Diagnostics Probe Group Summary opens in context of the probe group selected in the 360 view.

- 48539 - Graph is blank on the new "Server - Basic Metrics" view when first navigated to as a top level view.

  Problem: Graph is blank on the new "Server - Basic Metrics" view when first navigated to and when navigated to as a top level view.

  Resolution: The default graph is not blank and the graph shows the throughput data.

- 30577 - Call profile wedding cake for JMS onMessage() does not show the time for the onMessage() itself.

Problem: The call profile is displayed for queue://queue://FX. However, the top level bar/layer in the wedding cake does not show the time for the onMessage(). Instead, it shows the times for the methods called by the onMessage().

Resolution: Re-architecting the probe made this change possible.

- 49424 - Business Transaction view should not show CPU column.

  Problem: Don't show CPU column because it isn't used for business transactions.

  Resolution: Removed column from screen definition.

Integrations

- 48259 - HIs missing "sec" units.

  Problem: Health Indicators from Diagnostics missing "sec" units.

  Resolution: Added "sec" to Synthetic Business Transaction HIs.

- 48318 - There are 2 indicators in HPBSM\AppServer\webapps\site.war\bam\conf\repositories\external\diagnostics\indicators\def.xml that have same name on the same CI type.

  Problem: Duplicate Health Indicator definition.

  Resolution: Removed the duplicate TransactionManager definition on diagnosticsprobe.

- 49972 - Diagnostics Server Linux 64-bit Agent Registration Fails with BSM 9.0x.

  Problem: The Agent Registration with BSM 9.0x fails with the Diagnostics Commanding Server Linux 64-bit version.

  Resolution: The <install dir>/bin/swith_ovo_agent.sh script could not find the path of the Agent 64-bit libraries. The script was using /opt/OV/lib when it should have been using /opt/OV/lib64. Set lib64 directory.

Documentation

- 49236 - Setting up SiteMinder module.

  Problem - Problem with the configuration guidelines found in the Diagnostics Installation guide for setting up SiteMinder:

  Set up Apache Server 2.2 with reverse proxy (reverse proxy redirection works fine) to Diagnostics

  Enable the SiteMinder Agent in the Apache Server

  Configured the SiteMinder Module in the jaas.configuration

  When entering into the SiteMinder site and provide credentials, the Browser shows "Internal Server Error".

Resolution: Updated the 9.01 Installation Guide, Appendix B, section on "Using Reverse Proxy with SiteMinder JAAS LoginModule" in the example of Apache reverse proxy setup on HP-UX, updated the procedure as follows:

Edit the Apache configuration file httpd.conf and add the following properties:

ProxyPass /siteminderagent !

ProxyPass / **Error! Hyperlink reference not valid.** of Diagnostics Server>:2006/

(2006 is the default Diagnostics Server port, use the port configured for your Diagnostics Server)

ProxyPassReverse **/ Error! Hyperlink reference not valid.** of Diagnostics Server>:2006/
(2006 is the default Diagnostics Server port, use the port configured for your Diagnostics Server)

What's New in 9.00

Diagnostics 9.00 has a number of defect fixes and a number of new features shown below.

New Features

- Business Transactions
  - Diagnostics can now automatically detect and report on "Business Transactions", identifying cross VM dependencies and reporting the time taken along the different paths of the transaction and showing the topology in the UI.
  - The configuration of these transactions can be done in either the Diagnostics UI or the BSM UI.
  - The Business Transactions can be viewed in either the BSM UI or the Diagnostics UI as a topology or table view.
  - In addition, users will be able to see server request topologies in Diagnostics (not previously available).
- Business Service Management integration
  - We have greatly improved integration with the BSM framework including the population and synchronization of many Configuration Items (CIs) within the ODB. In addition, we are populating Key Performance Indicators (KPIs) and Health Indicators (HIs) inside of the BSM Dashboard.
  - Business Transactions can be viewed in the BSM Transaction Management UI and can be used to track service levels with SLM.
- UI Enhancements
  - Two new View Groups
    - Aggregate View Group - entities aggregated by probe group (useful in cluster environments)
    - Transactions View Group - Business Transactions (All Users) and Synthetic Transactions (BPM)
  - New Java Probes and .NET Probes views under Standard views
  - SQL Statements by Probe view added to Standard views
  - Outlook Style tabbed pane now organizes the view groups
  - Virtual Hosts - VM Guest value is displayed in Host Topology view
  - Open in Custom View - User can create a custom view that can be opened in context for a selected entity. (For example > Save view for a probe and open it in context of any probe, this

is similar to opening a probe in the Probe Summary view.)

    - Chart Groups - In any Standard entity view, the user can group the chart trends by metric or by entity.
    - View context sensitive help for each UI view through use of new Help menu item "Help on View".
    - User can export data from a Diagnostics view into a CSV file
        - Export Selected chart data to CSV
        - Export the data zoomed to in a chart in Standard views
        - Exports the trend for all charts in summary views
    - The Mercury System user can use Diagnostics in a large deployment to manage System Health
        - Show number of probes per probe group and number of probes per mediator (Capacity View)
        - User can see only the probes of interest in the Diagnostics UI screens (System Health)
    - UI configurations (ui.properties) for large deployments
        - Set the Top N probe entities query parameter
        - Throttle server query frequencies by query type
    - CAM Discovery Policy informs UI to Include probe's entities in top level views
        - Default setting for all applications
        - Informs the UI to not filter by application below the probe level
- Diagnostics will automatically discover Business Transactions CIs based on server requests-Rule based definition of Business Transactions
- .NET agent enhancements
    - Scalability improvements - The .NET agent deployments can scale more with the Diagnostics server with the use of the .NET Probe Aggregator Service on the agent system. This reduces significantly network I/O on both the agent and server systems.
    - Monitoring of asynchronous .NET applications; Sending and receiving messages using MSMQ.
    - WCF features match those of other protocols.
    - Diagnostics .NET agent will provide collection data to the server (LWMD).
    - .NET agent collects more metrics OTB and the user can easily add new metrics on demand.
    - CLR 4.0 .NET environment support.
    - .NET agent will support parameter capture.
    - .NET agent captures parameters of the first method that sees an exception thrown.

  - .NET agent has enhanced correlation capability to outbound http calls made to Java components.

- Java Agent enhancements
  - New Collection Leak Pinpointing feature automatically detects memory leaks. This feature is incredibly cool!
  - Java agent correlation across multiple threads - In complex Enterprise Service Bus solutions like Tibco Active Matrix, WebLogic's AquaLogic and others - a single "server request" can span multiple threads within the ESB. Until now, Diagnostics could not correlate the server request across the threads, now it can show the total latency across the threads. Initially, we support WebLogic AquaLogic and will add others (like Tibco) as customer demands dictate.
  - Cross VM correlation can be configured through instrumentation to support any protocol, including home grown generic TCP communication.
- User can see Server Request topology and Business Transaction flow from a .NET probe to a Java probe based on an HTTP call (and back).
- Other misc enhancements:
  - CORBA Cross MV Tracing
  - User can select probe metrics to be included in the LR, PC offline analysis file.

System Requirements

The HP Diagnostics Installation and Configuration Guide contains information about system requirements. This guide is available in a PDF version (Diagnostics_Install_Guide.pdf) located on the installation disk or in the Diagnostics Server download package. Examples of typical system requirements are provided below for the Diagnostics Server.

For the most recent information on supported environments refer to the Diagnostics Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp.

Examples of Requirements for the Diagnostics Server

Diagnostics uses the Java 1.7 JVM for the Server (and the Collector).

Java 1.7 JVM is supported on the following operating systems and requires the patches listed below.

The following table shows an example of typical system requirements for the host of a Diagnostics Server with different numbers of Java Probes.

Platform	Item	Up to 50 Java probes	Up to 100 Java probes	Up to 200 Java probes
Windows	CPU	2x 2.4 GHz	2x 2.8 GHz	2x 3.4 GHz
Windows	Memory	4 GB	4 GB	4 GB
Solaris	CPU	2x Ultrasparc 3	2x Ultra Sparc 4	2x Ultra Sparc 4
Linux	CPU	2x 2.0 GHz	2x 2.4 GHx	2x 2.8 GHz
Linux	Memory	2 GB	4 GB	4 GB
All	Java Heap	512 M	750 M	1280 M
All	Disk	4 GB per probe		
Notes regarding the test environment:				

Platform	Item	Up to 50 Java probes	Up to 100 Java probes	Up to 200 Java probes
Call profile (depth of method calls) for each Server Request: 5 Number of unique Server Requests per probe: 23				

The following table shows an example of typical system requirements for the host of a Diagnostics Server with different numbers of .NET Probes.

Platform	Item	Up to 10 .NET probes	Up to 20 .NET probes	Up to 50 .NET probes
Windows	CPU	1x 1.0 GHz	1x 2.0 GHz	2x 2.4 GHz
Windows	Memory	768 GB	1 GB	3 GB
Solaris	CPU	1x Ultrasparc 2	2x Ultra Sparc 2	2x Ultra Sparc 3
Solaris	Memory	1 GB	1.5 GB	3 GB
Linux	CPU	1x 1.0 GHz	1x 2.0 GHx	2x 2.4 GHz
Linux	Memory	768 GB	1 GB	3 GB
All	Heap Size	350 M	700 M	1400 M
All	Disk	3 GB per probe		

Information on System Impacts When Probed (.NET Probe)

Item	Minimum
Latency	+3% while CPU < 50%-once the CPU starts saturating, latency overhead increases
Memory	60 MB additional RAM
CPU	+5%
Disk Space	200MB free disk space is required for the probe install
Network I/O	+3% to +5% (assuming a normal ASP.NET application with a remote database Backend)
Disk I/O	Impact should be statistically insignificant
Throughput	< 1% reduction while CPU < 50% - should not exceed 15% reduction even under peak load Be careful with other metrics once throughput is impacted – as the application is doing less work, results are not directly comparable
RAM utilization	Baseline footprint increase of up to 20MB per worker process Note: This is quite difficult to measure accurately.

Information on System Impacts When Probed (Java Probe)

Item	Minimum
-Xmx	Add 40 MB
Disk Space	200MB free disk space is required for the initial probe install. Note, more space may be required during runtime due to the creation of logfiles and classmap. For large applications, it is recommended to have an additional 200MB available per probe for logfiles and classmap data.
Latency	+3% while CPU < 50%—once the CPU starts saturating latency overhead increases
CPU	+5% to +10%
Network I/O	+3% to +5% (assuming a normal Java EE application with a JDBC, RMI, or Web-services Backend)
Disk I/O	Impact should be statistically insignificant
Throughput	< 1% reduction while CPU < 50% - should not exceed 15% reduction even under peak load Be careful with other metrics once throughput is impacted – as the application is doing less work, results are not directly comparable
RAM utilization	Baseline footprint increase of up to 40MB

The requirements and general impact on the system depends on the monitored application and on the probe configuration. The values listed above are examples for a default probe configuration only.

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

- 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" erros after deploying probe on an IBM Java 6 64-bit JVM (60289)

      A bug in some versions of IBM's Java implementation may 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.

  JVMDUMP032I JVM requested System dump using 'XXXX.JVM.XXXX3.D120523.T133140.X&DS' in response to an event

  IEATDUMP in progress with options SDATA=(LPA,GRSQ,LSQA,NUC,PSA,RGN,SQA,SUM,SWA,TRT)

  IEATDUMP success for DSN='XXXX.JVM.XXXX3.D120523.T133140.X&DS'

```
JVMDUMP010I System dump written to XXXX.JVM.XXXX3.D120523.T133140.X&DS

JVMDUMP032I JVM requested Java dump using
'/tmp2/JavaAgent/DiagnosticsAgent/javacore.20120523.133140.50397214.0002.txt' in
response to an event

JVMDUMP010I Java dump written to
/tmp2/JavaAgent/DiagnosticsAgent/javacore.20120523.133140.50397214.0002.txt

JVMDUMP032I JVM requested Snap dump using
'/tmp2/JavaAgent/DiagnosticsAgent/Snap.20120523.133140.50397214.0003.trc' in response to
an event

JVMDUMP010I Snap dump written to
/tmp2/JavaAgent/DiagnosticsAgent/Snap.20120523.133140.50397214.0003.trc

JVMDUMP013I Processed dump event "traceassert", detail "".
```

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 Collector 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. Therefore, these calls do not show up in the Outbound Calls view. You can enable it as follows:
  - For version 7.0 and higher of the Java Probe, set create.database.fragmentArcs=true in the probe's dispatcher.properties file.

    Once capturing of Outbound database calls has been enabled, these calls will be displayed in the Outbound Calls view. (40722)
- 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

- 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/uibridge/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)

Collectors (Oracle, SAP, MQ, SQL Server)

- 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

- 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?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.

.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)

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.

User Interface

- The ability to delete more than one alert rule at a time in the UI is currently not supported. (70380)

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

  For example, if a custom view 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 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 standalone in IE8 (not inside of BSM).

- 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

- When the Diagnostics Server is installed on 64-bit Linux systems (only) the docs directory is not created and so the manuals are not accessible from the <diagnostics_server_install_dir>/docs directory. Copies of the pdf files can be found on the installation media and in the <diagnostics_server_install_dir>/html/opal/docs/eng/help/pdfs directory. The manuals are also accessible from the Diagnostics Online help.
- 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.

Environment

- For an IPV6 environment, the following restrictions exist:
  - All Diagnostic Server, Collector, and Java Probe configuration must be based on host names (not IP addresses).
  - .NET Probes are not supported.
  - Collectors must be running on IPV4 tunneled networks.
  - Linux and Solaris operating systems must be explicitly configured to use the IPV6 network host/IP resolution. On dual hosts, update /etcnsswitch.conf. On Solaris, update ipnodes.

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 might 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 and documentation in these languages:

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