

HP Diagnostics

for the Windows® , Unix and Linux operating systems

Software Version: 9.01

User's Guide

Document Release Date: September 2010

Software Release Date: September 2010



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© 2004 - 2010 Hewlett-Packard Development Company, L.P.

Trademark Notices

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

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 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

Additional troubleshooting resources are available on the HP Software Support web site. Choose Help > Troubleshooting & Knowledge Base to access the Troubleshooting page where you can search the Self-solve knowledge base. The URL for this Web site is <http://h20230.www2.hp.com/troubleshooting.jsp>.

Table of Contents

Welcome to This Guide	15
How This Guide Is Organized	15
HP Diagnostics Documentation	17
How Do I Find the Information That I Need?	18
Additional Online Resources	20
Documentation Updates	21

PART I: DIAGNOSTICS INTRODUCTION

Chapter 1: Introducing Diagnostics	25
Diagnostics Overview	26
Diagnostics Solutions Overview	28
Diagnostics Components	31
Interpreting Diagnostics Data	35
How to Install and Configure Diagnostics	37
How to Access the Diagnostics UI	39

PART II: USING DIAGNOSTICS

Chapter 2: Working with Applications	45
Applications Overview	46
Using the Applications Window	46
Entities and Parent Entities Added to an Application	49
Working in the Application Context	51
How to Create Application Groups	53
How to Create/Edit Applications	54
How to Delete an Application	58
How to Add/Remove Entities from an Application	58
How to Select an Application	61
Application Permissions	63
Applications User Interface	64

Chapter 3: Common Controls in the Diagnostics UI	71
Common Diagnostics UI Controls.....	73
View Context Drop-down Menu	75
Monitor and Analyze Tabs	76
Panning, Pausing and Zoom Controls	77
Viewing Time Filter	78
Diagnostics Toolbar.....	81
View Toolbar	81
Breadcrumb Trail.....	82
View Bar.....	83
About Different View Layouts	86
Resizing and Docking Windows	92
About the Different Display Panes in a View.....	95
About Graphs (Chart).....	97
About View Filters	100
Multiple Graph Layout Options.....	101
Entities and Metrics Charted in the Graph.....	102
The Data Behind a Charted Metric on the Graph	104
Zoom Features in the Graph	107
About the Graph Entity Table.....	108
About Status Indicators	110
Additional Information about the Entities in the Table.....	112
Table Header Controls.....	114
UI Properties	116
About Common Tasks and Navigations	117
Common Tasks.....	119
Common Navigations	125
How to Save or Export Data	128
How to Work with View Filters.....	131
How to Zoom In and Out on a Graph	137
How to Customize Table Columns	141
How to Customize Table Sort Order	144
How to Find an Entity in the Table	146
How to Select Multiple Rows in the Table	147

Chapter 4: Working with Thresholds and Metrics in the Details Pane	149
About the Details Pane	150
Custom Attributes	154
Chart a Metric in the Graph.....	156
Include a Metric in a Snapshot	157
Set an Alert Rule for a Metric	157
About Metric Thresholds.....	158
Threshold Definitions	158
View Metric History.....	162
Query Page to View All Thresholds Set	162
Audit Log of Threshold Changes	163
Using Thresholds in Monitoring Business Applications.....	163
Investigating Threshold Violations.....	165
Probe CPU Utilization Metrics	167
Configure Metrics.....	168
Set Threshold Dialog Box Description	173
Chapter 5: Working with Alerts	179
About Alerts.....	180
Alert Properties	181
About Alert Notification Rules	183
Alert Event and Alert Notification Logs.....	184
How to Access the Alert Properties Page	186
How to Enable and Disable Alert Notifications	187
How to Configure SNMP or Email Alert Notifications	188
How to Create or Maintain Alert Notification Rules for an Entity... ..	189
How to Create or Maintain Alert Notification Rule on a Metric	190
Alert Notifications User Interface.....	194
Chapter 6: Performing Snapshot Analysis	201
About Snapshot Analysis.....	202
Monitoring Performance Versus Analyzing Snapshots.....	203
How to Create and Manage Snapshots	205
How to Add and Delete Metrics or Entities from a Snapshot	207
Analyze Snapshots User Interface	210

Chapter 7: Working with Custom Views	215
About Custom Views.....	216
Retention of Customization.....	216
Share Custom Views.....	217
Upgrade Custom Views from Previous Diagnostics Version	219
Custom Views for Use with Different Instances (Open in Custom View)	219
How to Create a Custom View Group.....	221
How to Hide or Open a Custom View Group.....	222
How to Create a Custom View	223
How to Save a Custom View	227
How to Rename a Custom View	229
How to Delete a Custom View	230
My Views User Interface.....	231
Chapter 8: Working with Topologies	243
About Diagnostics Topologies.....	244
Description of the Topology Layout	245
The Topology Diagram.....	247
The Topology Toolbar	249
Save the Topology to an HTML Web Page.....	256
Chapter 9: Working with Instance Trees	257
About Instance Trees	258
Solve Performance Problems Using Instance Tree Data	261
Cross VM Trees	265
Exception Data	266
SOAP Fault Data	269
Types of SOAP Fault Markers	271
When Instance Trees Are Not Sufficient.....	271
Aggregate Trees.....	272

PART III: DIAGNOSTICS STANDARD VIEWS

Chapter 10: Server Summary View	281
Server Summary Overview	282
How to Use the Server Summary View	285
Server Summary User Interface	289
Chapter 11: Application Explorer View	315
Application Explorer Overview	316
How to Use the Application Explorer	319
Application Explorer Available Metrics by Entity Type.....	324
Application Explorer User Interface	326

Chapter 12: Application Metrics View	337
Application Metrics Overview	338
How to Use the Application Metrics View	339
Application Metrics User Interface	341
Chapter 13: Topology View	347
Topology View of the Probes	348
Selecting an Entity from Topology Diagram or Table	349
Topology Diagram Overview	351
How to Use the Topology Diagram Probe Grouping	356
Probe Topology User Interface	358
Chapter 14: Hosts View	369
Hosts Overview	370
How to Use the Hosts View	372
Hosts User Interface	373
Chapter 15: Server Requests View	379
Server Requests Overview	380
Display CPU Time for Server Requests	381
Instance Trees and Call Profiles for Server Requests	382
About Instance Tree Markers	383
How to Use the Server Requests View	385
How to Delete a Server Request	388
Server Requests User Interface	389
Chapter 16: Call Profile View	405
Call Profile Overview	406
Configure Collection and Display of Exception Data	409
Configure Collection of SOAP Faults and Payloads	409
Asynchronous Thread Stack Trace Sampling	411
How to Use the Call Profile	414
How to do Call Profile Analysis	422
Call Profile User Interface	426
Chapter 17: Probes View	433
Probes Overview	434
How to Use the Probes View	436
How to Configure Instrumentation Points for Method Trending	437
Probes User Interface	439

Chapter 18: Collections and Resources Views	459
Collections and Resources Overview	460
How to Enable LWMD for a .NET Probe	461
How to Enable LWMD for a Java Probe	461
How to Use the Collections and Resources Views	463
Collections and Resources User Interface	464
Chapter 19: Outbound Calls View	475
Outbound Calls Overview	476
How to Use the Outbound Calls View	478
Outbound Calls User Interface	479
Chapter 20: SQL Statements View	485
SQL Statements by Probe Overview	486
How to Set SQL Trending Thresholds	487
SQL Statements User Interface	489
Chapter 21: Layers View	495
Layers Overview	496
Custom Layers	496
How to Use the Layers View	497
Layers User Interface	498
Layers Defined by Default	502
Chapter 22: Load View	507
Load Overview	508
How Load is Calculated in Diagnostics	508
How to Use the Load View	510
Load User Interface	511
Chapter 23: Status View	515
Status Overview	516
Status Propagation	517
Status Indicators	517
Status User Interface	520

PART IV: UNDERSTANDING DIAGNOSTICS SPECIALIZED VIEWS

Chapter 24: Aggregate Views	529
Aggregate Views Overview	530
Aggregate Request Overview	530
Dependent Services Overview	531
SQL Statements Overview	532
Trended Methods Overview	532
How to Set SQL Trending Thresholds	534
How to Configure Instrumentation Points for Method Trending....	536
Aggregate View Group User Interface	538
Chapter 25: System Views.....	563
System Views Overview.....	564
System View Group User Interface.....	565
Chapter 26: Transactions Views.....	573
Transactions View Group Overview.....	574
Types of Transactions in Diagnostics.....	575
Synthetic Transactions and Backend Data from Diagnostics	577
Business Transaction Monitoring with Diagnostics	578
Business Transaction Paths in Diagnostics	579
How to Create or Edit Business Transactions in Diagnostics	583
How to Configure Business Transaction Discovery	587
Transactions View Group User Interface.....	591
Chapter 27: Portals Views	613
Portals Overview.....	614
Configure Collection of CPU Time Metrics for Portlets	616
Instance Tree Call Profiles for Portlets	618
How to Use the Portals View Group	621
Portals View Group User Interface	626

Chapter 28: SOA Services Views	651
About SOA Services Monitoring.....	653
Supported Web Service Platforms	654
SOA Services Overview	655
Instance Tree Call Profiles for SOA Services.....	656
Cross VM and SOAP Faults for Services	659
About Consumer IDs.....	663
Monitoring REST Style Services.....	665
About the Enterprise Service Bus Environment.....	666
How to Specify a Name for the Application Server Instance.....	667
How to Change the Display Name of a Web Service.....	668
How to Monitor a BEA AquaLogic Service Bus Environment.....	669
How to Monitor the SOA Policy Enforcer Broker	675
How to Monitor an IBM WebSphere Enterprise Service Bus Environment.....	680
How to Monitor TIBCO ActiveMatrix Environment	684
How to View BEA WLI Business Process Tracing Data	690
SOA Services View Group User Interface	700
Chapter 29: SAP Views	743
SAP Overview	744
SAP View Group User Interface.....	745
Chapter 30: Oracle Database Views.....	777
Oracle Database Overview.....	778
Oracle Database View Group User Interface	780
Chapter 31: SQL Server Database Views	799
SQL Server Database Overview.....	800
SQL Server Database View Group User Interface	802
Chapter 32: MQ Views	821
MQ Overview	822
MQ View Group User Interface.....	823
Chapter 33: BEA WebLogic Views.....	839
BEA WebLogic Overview.....	840
BEA WebLogic View Group User Interface	841
Chapter 34: IBM WebSphere Views.....	869
IBM WebSphere Overview	870
IBM WebSphere View Group User Interface.....	871
Chapter 35: CICS Views.....	915
CICS Overview	916
CICS View Group User Interface.....	917

Chapter 36: External Monitors Views	947
External Monitors Overview	948
How to Display a Graph and See Status in the Monitors View	952
External Monitors View Group User Interface.....	953

PART V: USING THE DIAGNOSTICS PROFILERS FOR JAVA AND .NET

Chapter 37: Diagnostics Profiler for Java.....	963
About the Java Diagnostics Profiler.....	965
How the Java Agent Provides Data for the Java Profiler	966
Java Diagnostics Profiler UI Navigation and Display Controls	968
Analyzing Performance Using the Call Profile Window.....	969
Thread Call Stack Trace Sampling.....	976
Comparison of Collection Leak Pinpointing and LWMD	978
Object Lifecycle Monitoring	980
Heap Walker Memory Analysis Execution Steps	984
Heap Walker Performance Characteristics.....	989
How to Access the Java Diagnostics Profiler	991
How to Enable LWMD for Memory Analysis and Collections Displays	992
How to Enable Allocation Capture	993
How to Enable Object Lifecycle Monitoring.....	994
How to Analyze Object Allocation.....	995
How to Enable Heap Breakdown.....	996
How to Enable Memory Analysis	997
Java Diagnostics Profiler User Interface	999
Chapter 38: Diagnostics Profiler for .NET.....	1049
About the .NET Diagnostics Profiler	1050
How the .NET Agent Provides Data for the .NET Profiler.....	1051
.NET Diagnostics Profiler UI Navigation and Display Controls	1052
.NET Diagnostics Profiler Inactivity Timeout	1054
How to Access the .NET Diagnostics Profiler	1055
How to Enable and Disable the .NET Diagnostics Profiler	1056
.NET Diagnostics Profiler User Interface	1058

PART VI: INTEGRATIONS

Chapter 39: Diagnostics Integration with Other HP Software Products1083

- Diagnostics Integrates with other HP Software Products1085
- Diagnostics and Business Service Management.....1086
- Overview of Business Service Management Concepts for Diagnostics Users1088
- Overview of Diagnostics Data Provided to Business Service Management1090
- CI Population and Models1091
- Run-time Service Model1093
- Event Based and Metric Based Coloring/HIs.....1097
- Data Samples Sent to Business Service Management.....1097
- Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics1098
- Application Infrastructure HI Status Set Based on Events from Diagnostics.....1100
- Open the Diagnostics UI and Online Help from Business Service Management1106
- The Diagnostics View in Business Service Management Service Health.....1108
- Business Transactions View in Business Service Management Service Health1110
- General Integration between Diagnostics and Business Service Management1112
- Integration between Diagnostics and End User Management1114
- Integration between Diagnostics and Transaction Management...1115
- Integration between Diagnostics and Application Management for SOA1117
- Diagnostics Configuration that can Affect Integration with Business Service Management1118
- Integration Between Diagnostics and SiteScope1121
- Integration Between Diagnostics and LoadRunner1122
- Analyze Offline Data Provided by Diagnostics1122
- Integration Between Diagnostics and Performance Center.....1124
- Index.....1125**

Welcome to This Guide

Welcome to the *HP Diagnostics User's Guide*. This guide describes how to use HP Diagnostics to analyze the performance of your enterprise applications.

How This Guide Is Organized

The guide contains the following parts:

Part I Diagnostics Introduction

Provides an overview of Diagnostics concepts, data collection, and a general workflow for installing and configuring Diagnostics.

Part II Using Diagnostics

Describes how to work with the HP Diagnostics views once the components are installed and configured.

Part III Diagnostics Standard Views

Describes the HP Diagnostics standard views.

Part IV Understanding Diagnostics Specialized Views

Describes the HP Diagnostics specialized views.

Part V Using the Diagnostics Profilers for Java and .NET

Describes the Diagnostics Profiler for Java and the Diagnostics Profiler for .NET.

Welcome to This Guide

Part VI Integrations

Describes Diagnostics integration with other HP Software products.

HP Diagnostics Documentation

Your HP Diagnostics application comes with the following documentation:

- ▶ **Diagnostics User's Guide and Online Help.** Explains how to use HP Diagnostics to analyze the performance of your enterprise applications. You access the online help for Using HP Diagnostics from the **Help** button in Diagnostics or from the help menu in the integrated HP Software product. You access the PDF version of this guide from help, or from the Windows Start menu (**Start > Programs > HP Diagnostics Server > User Guide**), or from the **Documentation** directory on the HP Diagnostics installation disk, or from the Diagnostics Server installation directory.
- ▶ **Diagnostics Installation and Configuration Guide.** Explains how to install and configure the Diagnostics components and how to configure Diagnostics for integration with other HP Software products. You access this guide from the help, or from the **Documentation** directory on the Diagnostics installation disk, or from the Diagnostics Server installation directory, or from the Windows Start menu (**Start > Programs > HP Diagnostics Server > Install Guide**),
- ▶ **Readme.** Provides last-minute technical and troubleshooting information about HP Diagnostics. The file is located in the HP Diagnostics installation disk root directory. There is also a Patch Install Instructions document with details for installing a patch release.
- ▶ **Diagnostics Java Agent Guide.** Describes how to install, configure, and use the Diagnostics Java Agent and the Diagnostics Profiler for Java. You access this guide on the agent system, or from the Profiler UI Help link or in the Documentation directory on the HP Diagnostics installation disk.
- ▶ **Diagnostics .NET Agent Guide.** Describes how to install, configure, and use the Diagnostics .NET Agent and Diagnostics Profiler for .NET. You access this guide on the agent system, or from the Profiler UI Help link or in the Documentation directory on the HP Diagnostics installation disk.

Note: The information in the Diagnostics Agent guides is based on information in the **Diagnostics Installation and Configuration Guide** and the **Diagnostics User's Guide**.

How Do I Find the Information That I Need?


This guide is part of the HP Diagnostics online help, provided in pdf format if you want to print or search the document as a whole.


Topic Types



Within this guide, each subject area is organized into topics. A topic contains a distinct module of information for a subject. The topics are generally classified according to the type of information they contain.

This structure is designed to create easier access to specific information by dividing the documentation into the different types of information you may need at different times.

Three main topic types are in use: **Concepts**, **Tasks**, and **Reference**. The topic types are differentiated visually using icons.

Topic Type	Description	Usage
Concepts 	Background, descriptive, or conceptual information.	Learn general information about what a feature does.

Topic Type	Description	Usage
Tasks 	<p>Instructional Tasks. Step-by-step guidance to help you work with the application and accomplish your goals. Some task steps include examples, using sample data. Task steps can be with or without numbering:</p> <ul style="list-style-type: none"> ▶ Numbered steps. Tasks that are performed by following each step in consecutive order. ▶ Non-numbered steps. A list of self-contained operations that you can perform in any order. 	<ul style="list-style-type: none"> ▶ Learn about the overall workflow of a task. ▶ Follow the steps listed in a numbered task to complete a task. ▶ Perform independent operations by completing steps in a non-numbered task.
	<p>Use-case Scenario Tasks. Examples of how to perform a task for a specific situation.</p>	Learn how a task could be performed in a realistic scenario.

Topic Type	Description	Usage
 Reference	General Reference. Detailed lists and explanations of reference-oriented material.	Look up a specific piece of reference information relevant to a particular context.
	User Interface Reference. Specialized reference topics that describe a particular user interface in detail. Selecting Help on this page from the Help menu in the product generally open the user interface topics.	Look up specific information about what to enter or how to use one or more specific user interface elements, such as a window, dialog box, or wizard.
 Troubleshooting and Limitations	Troubleshooting and Limitations. Specialized reference topics that describe commonly encountered problems and their solutions, and list limitations of a feature or product area.	Increase your awareness of important issues before working with a feature, or if you encounter usability problems in the software.

Additional Online Resources

Troubleshooting & Knowledge Base accesses the Troubleshooting page on the HP Software Support Web site where you can search the Self-solve knowledge base. Choose **Help > Troubleshooting & Knowledge Base**. The URL for this Web site is <http://h20230.www2.hp.com/troubleshooting.jsp>.

HP Software Support accesses the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > HP Software Support**. The URL for this Web site is www.hp.com/go/hpsupport.

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

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

To register for an HP Passport user ID, go to:

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

HP Software Support accesses the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > HP Software Support**. The URL for this Web site is www.hp.com/go/hpsoftwaresupport.

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

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

To register for an HP Passport user ID, go to:

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

HP Software Web site accesses the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help > HP Software Web site**. The URL for this Web site is www.hp.com/go/software.

Documentation Updates

HP Software is continually updating its product documentation with new information.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to the HP Software Product Manuals Web site (<http://h20230.www2.hp.com/selfsolve/manuals>).

Welcome to This Guide

Part I

Diagnostics Introduction

1

Introducing Diagnostics

This chapter introduces Diagnostics and discusses general topics related to installation, configuration and system administration.

This chapter includes:

Concepts

- ▶ Diagnostics Overview on page 26
- ▶ Diagnostics Solutions Overview on page 28
- ▶ Diagnostics Components on page 31
- ▶ Interpreting Diagnostics Data on page 35

Tasks

- ▶ How to Install and Configure Diagnostics on page 37
- ▶ How to Access the Diagnostics UI on page 39

Concepts

Diagnostics Overview

HP Diagnostics is a composite application monitoring, triage and diagnostics solution that is designed to help you improve the performance of your Java, .NET, and other enterprise applications throughout the application lifecycle. HP Diagnostics enables you to:

- ▶ detect slow performing components and code in pre-production or production.
- ▶ gain end-to-end visibility of composite components through transaction tracing across multiple tiers.
- ▶ isolate the performance of incoming requests and correlate them with outbound requests.
- ▶ measure latency at service-consumer level and service-provider level.
- ▶ discover "rogue" code/components real-time as they are invoked.
- ▶ reduce Mean Time To Repair (MTTR) by shortening the Composite Application triage time.

HP Diagnostics provides solutions for:

- ▶ **Composite Applications** which are a combination of legacy, packaged and new application logic developed on Java/.NET and middleware/legacy platforms. Diagnostics collects performance metrics from a wide range of composite application components including SAP, WebSphere MQ, Oracle, SQL Server Database, and portal implementations. Diagnostics also collects data from the transport layer supporting HTTP, RMI, WS, and JMS protocols. Diagnostics correlates the performance metrics to provide graphical topology views and cross component transaction views.

- ▶ **Java Applications** that run on most of the Java EE compliant application servers. Diagnostics collects performance metrics on the servlets, JSPs, EJBs, JNDI, JDBC, JMS, Portlets and Struts method calls that are performed by your application. You can customize the instrumentation that Diagnostics uses to monitor your applications so that it will capture specific business logic methods.
- ▶ **.NET Applications** that run on the Microsoft .NET Framework. Diagnostics uses runtime instrumentation to capture method latency information from specified applications. By default, Diagnostics captures server requests for ASP.NET, WCF and outbound ADO.NET calls. You can customize the instrumentation that Diagnostics uses to monitor your applications so that it will capture specific business logic methods. Instrumentations files (point files) are provided for .NET applications using MSMQ and .NET Remoting and these points files will need to be included in the appdomain scope to enable monitoring. Monitoring memory allocations of collections in .NET can be enabled by turning on the Light Weight Memory Diagnostics (LWMD) feature.
- ▶ **Database Applications** including Oracle Database and SQL Server Database.
- ▶ **SAP.** HP Diagnostics provides end-to-end analysis of SAP Enterprise Portal transactions starting from portal pages and iViews, while maintaining the business context of services provided through the portal. Diagnostics also supports tracing and diagnosis of Java WebDynPro applications hosted on the Portal.
- ▶ **SOA Services.** Diagnostics detects SOA services, operations and consumers and monitors performance and availability of these services. Diagnostics auto discovers and provides a services topology.

HP Diagnostics has been integrated with HP's Business Availability Center and Performance Center solutions to provide you with the insight and information that you need to build, develop, test, and monitor applications that perform efficiently and effectively.

Diagnostics Solutions Overview

You can configure HP Diagnostics to work with one of HP Software's Application Delivery or Application Monitoring products, or you can use it as a stand alone diagnostics tool.

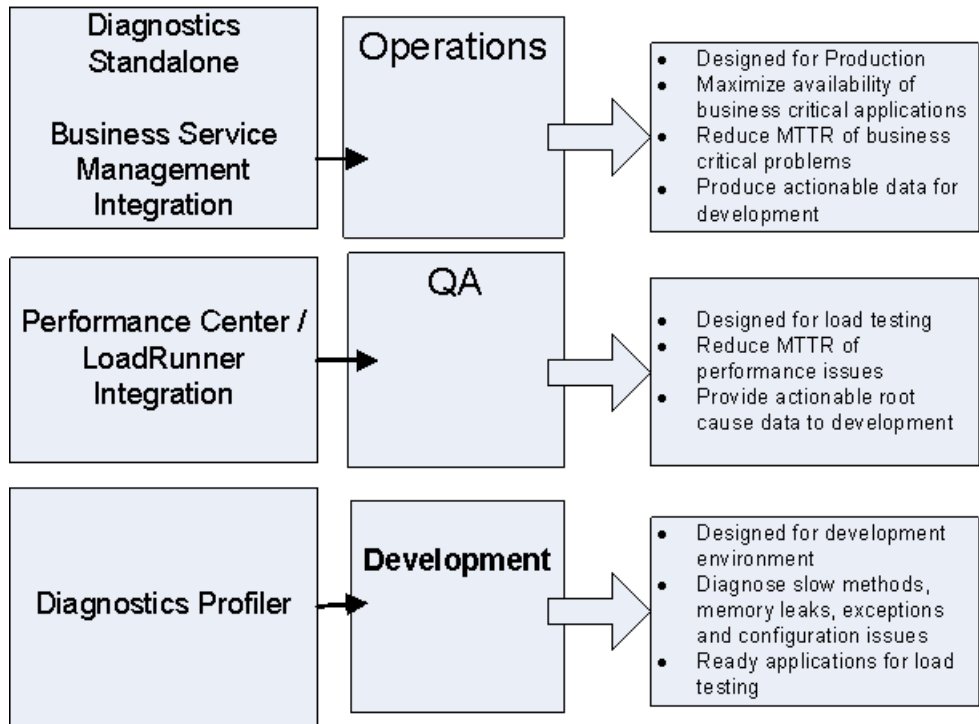
When you install the Java or .NET Agents that gather your application's performance metrics, the Diagnostics Profiler is automatically installed as well. The Diagnostics Profiler is an independent diagnostics application that has its own UI. It is accessed either directly on the local machine or through the main HP Diagnostics UI.

This section includes:

- "Diagnostics Solutions by Role" on page 29
- "Integration with Business Service Management" on page 29
- "Integration with LoadRunner and Performance Center" on page 30
- "Diagnostics Standalone" on page 29
- "Diagnostics Profilers" on page 30

Diagnostics Solutions by Role

The following diagram illustrates how the Diagnostics solution can be used across different parts of the organization.



Diagnostics Standalone

You can also work with Diagnostics as a standalone product. When you work in Diagnostics Standalone, you access the Diagnostics views directly from the Diagnostics Server.

Integration with Business Service Management

HP Diagnostics is integrated with Business Service Management, allowing you to monitor the availability and performance of your production enterprise application. This integration enables you to significantly reduce the Mean Time To Resolution of problems and thus increase the availability and value of the business applications.

From within Business Service Management, you can track the performance status of your applications that are being monitored by HP Diagnostics.

The Diagnostics integration with Business Service Management allows you to drill down to Diagnostics data from specific Business Service Management configuration items and reports. You can also generate high level reports in Business Service Management about the performance of applications and business transactions that are monitored by Diagnostics.

For more information about the Diagnostics integration with Business Service Management, see the Chapter 39, "Diagnostics Integration with Other HP Software Products."

Integration with LoadRunner and Performance Center

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

During a load test, you can drill down to HP Diagnostics data for the whole scenario or for a particular transaction. After you have run your scenario, you can use LoadRunner Analysis to analyze offline Diagnostics data generated during the scenario.

For more information about the Diagnostics integration with LoadRunner and Performance Center, see the Chapter 39, "Diagnostics Integration with Other HP Software Products."

Diagnostics Profilers

Diagnostics Java and .NET Profilers use the Diagnostics Agent to provide a development ready profiling capability that can be integrated into the product/JVM as part of the application lifecycle. Among other things, the Diagnostics Profiler helps you identify:

- ▶ where time is spent in an application; either processing data or waiting for a response from another part of the application.
- ▶ the slowest layers.
- ▶ the slowest server requests which are the application entry points.
- ▶ outliers to help diagnose intermittent problems.

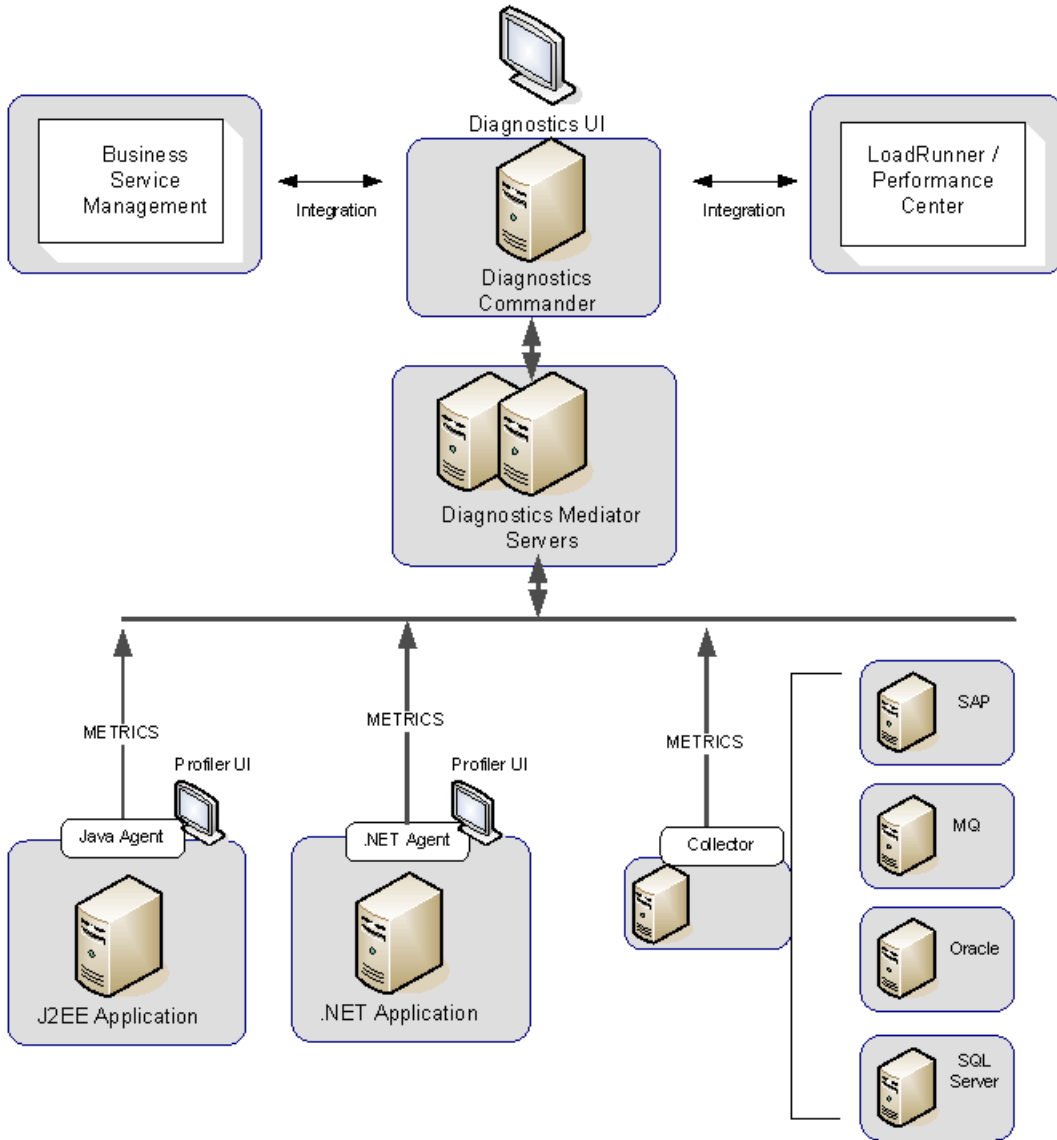
- threads that may be contributing to performance issues.
- memory problems and garbage collection issues.
- the fastest growing and largest size collections.
- leaking objects, object growth trends, object instance counts, and the byte size for objects.
- the slowest SQL query and report query information.
- exception counts and trace information which often go undetected.

For more information about using the Diagnostics Profiler for Java, see Chapter 37, "Diagnostics Profiler for Java." For more information about using the Diagnostics Profiler for .NET, see Chapter 38, "Diagnostics Profiler for .NET."

Diagnostics Components

The following diagram illustrates the Diagnostics components and integrations with other HP Software products.

Diagnostics Components



HP Diagnostics consists of the following components:

- ▶ **Diagnostics Agents.** Responsible for capturing events from your application, aggregating the metrics, and sending the performance metrics to a Diagnostics Server. The agents are configured to run probes to capture events such as method invocations, the beginning and end of business transactions, and server transactions. Multiple probes can be configured with each agent.

Note: The functionality of the Java Agent is provided by the HP Diagnostics/ TransactionVision Java Agent. The functionality of the .NET Agent is provided by the HP Diagnostics/TransactionVision .NET Agent.

- ▶ **Diagnostics Collectors.** Responsible for collecting data from external ERP/ CRM environments including Oracle 10g Database, SQL Server, IBM WebSphere MQ or SAP NetWeaver - ABAP system. You install the Diagnostics Collector and define specific instances of these systems to be monitored. Each instance of a collector is represented as a probe in the Diagnostics UI.
- ▶ **Diagnostics Servers.** Responsible for working with the agents, collectors and with other HP Software products to capture, process, and present the performance metrics for your application.

The Diagnostics Server processes and further aggregates the data that it receives from each of the probes that report to it, and formats the information so that it can be displayed in the views of the user interface.

The Diagnostics Server in Commander mode is responsible for the command and control functions between the various Diagnostics components and the components of the other products with which Diagnostics is working.

The Diagnostics Server in Commander mode keeps track of the location and status of the other Diagnostics components, and is the communication hub between the other components.

The Diagnostics Server in Commander mode is also responsible for displaying the performance information for the monitored applications in the charts and graphs of the Diagnostics views. If you are using Diagnostics with other HP Software products you can also access the Diagnostics views from the user interface of the other products.

A Diagnostics deployment may consist of one or many Diagnostics Servers. If there is only one Diagnostics server in your deployment, it is configured in Commander mode and must perform both the Commander and Mediator roles. If there is more than one Diagnostics Server in a deployment, one must be configured in Commander mode, and all the rest in Mediator mode.

In a typical deployment there is a Diagnostics Server in Commander mode connected to a one or more Diagnostics Servers in Mediator mode. Each Diagnostics Server in Mediator mode is connected to a number of agents or collectors that run probes to capture events from the monitored applications.

The probes send these captured performance metrics to the Diagnostics Server in Mediator mode, which filters and aggregates the events. This information is sent to the Diagnostics Server in Commander mode, which displays the processed metrics in customizable views.

- ▶ **User Interfaces.** When you are monitoring your application in real time, you access the Diagnostics UI from Business Service Management or directly from the Diagnostics Server. During a load test, you access the Diagnostics UI from LoadRunner or Performance Center.

You can also access the Diagnostics Profiler for Java and the Diagnostics Profiler for .NET directly from the agent system or through the Diagnostics UI.

Diagnostics has integrations with the following other HP Software products:

- ▶ Business Service Management
- ▶ LoadRunner
- ▶ Performance Center

Interpreting Diagnostics Data

The following sections explain the manner in which Diagnostics presents the performance information in its views.

About Layers and Sublayers

HP Diagnostics groups the performance metrics for the classes and methods invoked by your applications into layers and sub-layers based on the resources that the application invokes to perform the processing. These layers help you to isolate and identify the areas of the system that may be contributing to performance issues.

For Java applications, the methods and classes that are associated with each layer are defined in the capture points file `<probe_install_dir>auto_detect.points` where the instrumentation for the probe is specified.

For .NET applications, capture points files for ASP.NET applications are located in `<probe_install_dir>\etc\`.

For more information about Diagnostics layers and how classes and methods are assigned to layers in the Capture Points file, see the *HP Diagnostics Installation and Configuration Guide*.

For more information on viewing the performance metrics by layer, see "Layers View" on page 495.

About Time Measurements in Diagnostics

It is important to understand how Diagnostics calculates and presents the time measurements in the Diagnostics views. When analyzing the Diagnostics metrics and comparing them with the metrics reported by other HP Software products, you should consider the following:

- Response time measurements are not the same in Diagnostics and LoadRunner/Performance Center. Response time on the Diagnostics views refers to the server response time as measured on the server side. Response time on the LoadRunner/Performance Center screens refers to the server response time as measured on the client side. This can cause the response times in a Diagnostics view and the response times on a LoadRunner/Performance Center screen to differ.

- ▶ The server side may have multiple threads executing simultaneously. In Diagnostics the time spent on each thread is counted as opposed to the total overall time, this can cause the aggregate server side time to be much larger than the time as measured by the client.

Tasks

How to Install and Configure Diagnostics

This task describes the recommended working order for installing and configuring your Diagnostics server.

This task includes the following steps:

- "Prerequisites" on page 37
- "Install the Diagnostics Servers" on page 37
- "Install Diagnostics Agents and Collectors" on page 38
- "Additional Configuration for the Agents and Collectors" on page 38
- "Configure Diagnostics Integration with other HP Software Products - Optional" on page 39

1 Prerequisites

Carefully plan and prepare your installation:

- Check system requirements and other installation considerations
- Plan what application servers and applications you want to monitor.

See the *HP Diagnostics Installation and Configuration Guide* for information on requirements.

2 Install the Diagnostics Servers

Install the Diagnostics Server software on the system you plan to use as the Commanding server. And install the software on systems you plan to use as Mediating servers. Setup the Diagnostics Server components to work in your environment.

See the *HP Diagnostics Installation and Configuration Guide* for information on installation and configuration of Diagnostics Server components.

3 Install Diagnostics Agents and Collectors

Install and setup Diagnostics Java Agent software and Diagnostics .NET Agent software on the application servers you plan to monitor.

Install and setup Diagnostics Collectors for monitoring SAP, Oracle, SQL Server or MQ systems. The collectors do remote monitoring.

See the *HP Diagnostics Installation and Configuration Guide* for information on installation and setup of Diagnostics Agents and Diagnostics Collectors.

4 Additional Configuration for the Agents and Collectors

Configure the Java Agents to work in your environment:

- ▶ JRE instrumenter instruments the ClassLoader class for the JVM each application is using and provides you with the JVM parameter to use when the application server is started so the server will use the instrumented class loader. The JRE instrumenter runs automatically during the Java Agent installation and setup but it can optionally be run later.
- ▶ The Java Agent runs one or more probe data collections depending on how many JVMs/applications you are monitoring on the application server where the agent is installed.
- ▶ Configure the application server startup scripts so that the Java Agent that is to monitor the application gets started along with the application.
- ▶ System metrics and JMX/PMI metrics are collected by default. You can configure this, as needed, to control which metrics are collected.
- ▶ Do any custom instrumentation or advanced configuration, as needed.

Configure the .NET Agents to work in your environment:

- ▶ Change the default .NET agent configuration as needed using the `probe_config.xml` file
- ▶ Create or enable additional instrumentation points specific for your applications.
- ▶ Do any advanced configuration as needed.

Configure the Collectors to work in your environments:

- For each collector, define the SAP, Oracle, SQL Server or MQ systems you want to monitor. The instances you want to monitor are defined in xml files provided with the collector (for example, oracle-config.xml).

See the *HP Diagnostics Installation and Configuration Guide* for information on configuring the Java Agent, the .NET Agent and the Collector.

5 Configure Diagnostics Integration with other HP Software Products - Optional

Configure integration of Diagnostics with Business Service Management, Performance Center or LoadRunner. See Chapter 39, "Diagnostics Integration with Other HP Software Products" for an overview.

Also see the *HP Diagnostics Installation and Configuration Guide* for information on configuring integration with other HP Software products.

How to Access the Diagnostics UI

You can access HP Diagnostics views directly from the or from the user interface of other HP Software applications that have been integrated with Diagnostics.

This task includes the following steps:

- "Prerequisites" on page 37
- "Open the Diagnostics UI" on page 41
- "Open the Diagnostics UI from Business Service Management - Optional" on page 42
- "Open the Diagnostics UI from Performance Center - Optional" on page 42
- "Open the Diagnostics UI from LoadRunner - Optional" on page 42



Prerequisites

For optimal display of the Diagnostics views, ensure that your screen resolution is at least 1024x768.

Before you can access the Diagnostics UI from Business Service Management, LoadRunner or Performance Center you must configure these other HP Software products with the details for the Diagnostics Server. For more information, see the *HP Diagnostics Installation and Configuration Guide*.

Open the Diagnostics UI

Use the following procedure to open the Diagnostics UI directly.

- 1** In your browser, navigate to http://<diagnostics_server_host>:2006, or select **Start > Programs > HP Diagnostics Server > Administration**. Port number 2006 is the default port for the Diagnostics Server in Commander mode. If the Diagnostics Server was installed and configured to use an alternate port, specify that port number in the URL.
- 2** The Diagnostics Server administration page opens.
- 3** Select **Open Diagnostics**.
- 4** If you have not already signed into the Diagnostics Server, you are prompted to provide a user name and password.
- 5** Enter the user name and password for a user that has been granted the permissions necessary to open Diagnostics.
- 6** A user that has been granted **view** privileges is able to see the Diagnostics views. One of the default user names that you can use is **admin**; the default password for this user is **admin**. For more information about user names and user permissions, see the *HP Diagnostics Installation and Configuration Guide*.
- 7** Click **OK**.

Notes:

Diagnostics continues to prompt for a user name and password until you enter valid values.

If you click **Cancel**, Diagnostics displays the following error message in your browser: **Access denied. You must specify a valid username and password.**

If the user name and password that you entered are for a valid Diagnostics user that does not have permission to see the Diagnostics views, Diagnostics displays the following error message in your browser: **Access denied. You do not have the required permission to view this screen.**

If you are prompted for your user name and password a second time, enter the same information again, and click **Yes**.

Open the Diagnostics UI from Business Service Management - Optional

To access the Diagnostics UI from Business Service Management select any of the following:

- ▶ Click **Applications > Diagnostics** from the Business Service Management UI.
- ▶ Or on the Site Map menu, click Diagnostics.
- ▶ Or drill down to Diagnostics data from relevant Business Service Management configuration items and reports.

For more information, see the Chapter 39, "Diagnostics Integration with Other HP Software Products."

Open the Diagnostics UI from Performance Center - Optional

To access the Diagnostics UI from Performance Center select any of the following:

- ▶ Click **View Diagnostics** on the right side of the Performance Center window.
- ▶ Drill down to HP Diagnostics data from a particular listed transaction.

For more information, see the Chapter 39, "Diagnostics Integration with Other HP Software Products."

Open the Diagnostics UI from LoadRunner - Optional

To access the Diagnostics UI from LoadRunner select any of the following:

- ▶ Click the **Diagnostics for J2EE/.NET** tab at the bottom of the LoadRunner Controller window.
- ▶ Drill down to HP Diagnostics data from a particular listed transaction.

For more information, see the Chapter 39, "Diagnostics Integration with Other HP Software Products."

Part II

Using Diagnostics

2

Working with Applications

This chapter describes the main concepts, tasks, and reference information for working with applications.

This chapter includes:

Concepts

- Applications Overview on page 46
- Using the Applications Window on page 46
- Entities and Parent Entities Added to an Application on page 49
- Working in the Application Context on page 51

Tasks

- How to Create Application Groups on page 53
- How to Create/Edit Applications on page 54
- How to Delete an Application on page 58
- How to Add/Remove Entities from an Application on page 58
- How to Select an Application on page 61

Reference

- Application Permissions on page 63
- Applications User Interface on page 64

Concepts

Applications Overview

An **Application** in Diagnostics is a composite application made up of a logical grouping of other Diagnostics entities, such as probes, hosts and server requests.

You can define applications in Diagnostics which allows you to view Diagnostics data within the context of that application.

An **application group** allows you to organize applications together into groups of related applications. An application group can contain applications, and it can contain other application groups.

By default, there is a predefined application (**Entire Enterprise**) that includes the entire managed environment context. In this application, all Diagnostics views are displayed showing all the relevant entities.

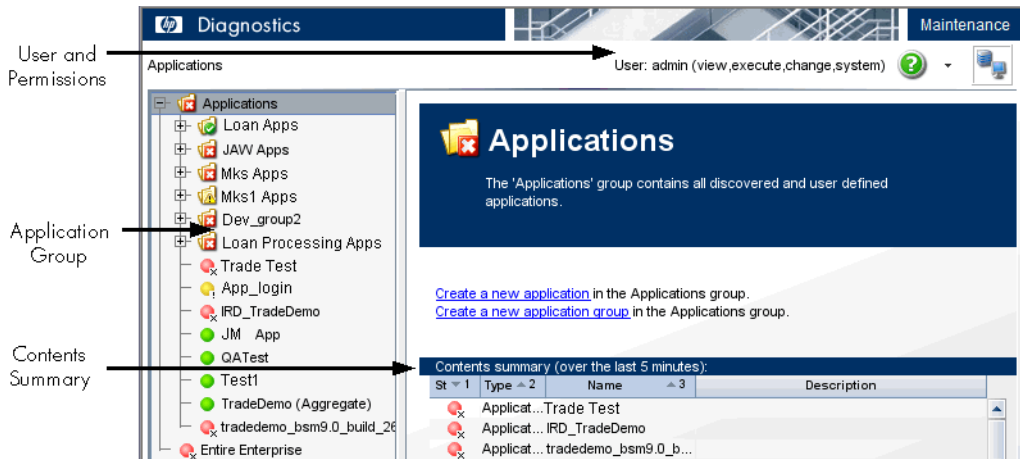
When you have Business Process Monitors running in your environment, Diagnostics creates an application for each Business Process Monitor application (in Business Process Monitor 8.x the BPM Profile application is created).

Using the Applications Window

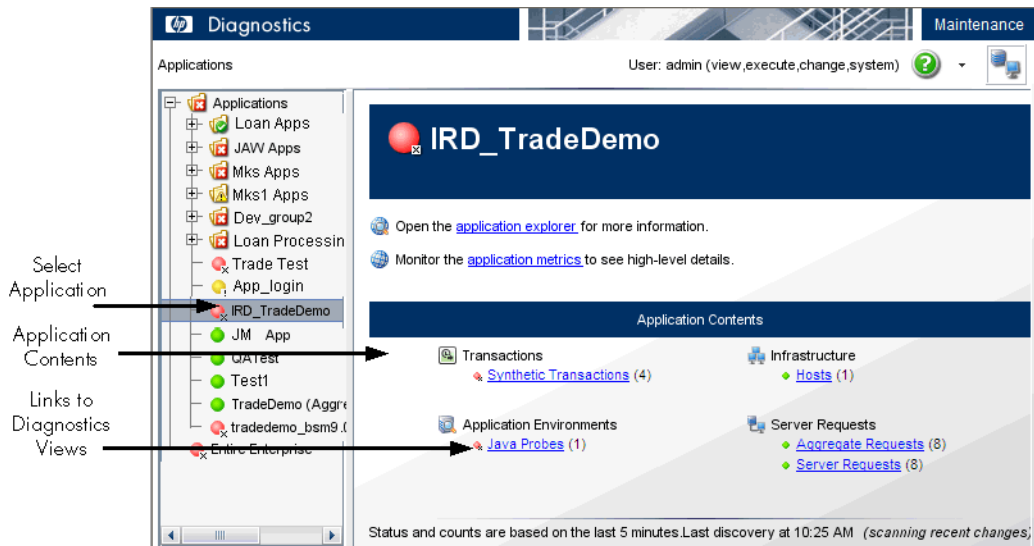
From the Applications window, you can select an existing application and go to a Diagnostics view in the context of the selected application. Or you can also select the Entire Enterprise application and view all data displayed in the Diagnostics views.

You can also create, update, and delete applications, provided you have the required permissions.

When you first enter the Applications window or you select an **application group** in the selection pane, the **Contents Summary** shows information about the application group and applications in the group.



When you select an **application** (or select Entire Enterprise), the **Application Contents** shows the entities that make up the application. You can click any of the entity types to go directly to that Diagnostics view (for example, click the Java Probes link to display the Java Probes view).



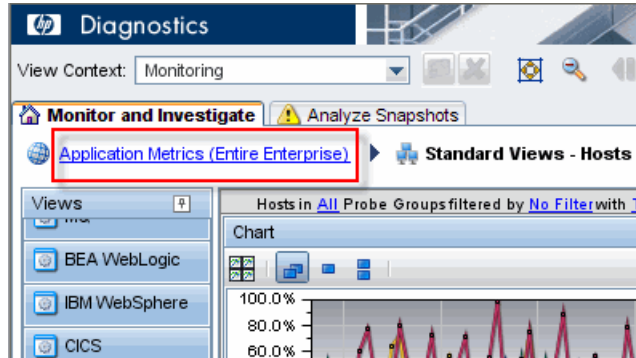
For how to create applications, see "How to Create Application Groups" on page 53.

A typical process for working with applications in Diagnostics includes;

- ▶ Define application groups and applications (you need Execute permissions).
- ▶ Select **Entire Enterprise** in the Applications window, and enter the Diagnostics views.
- ▶ Add entities to the applications you created. Many entities are automatically discovered and added to the application based upon what you add. For example, if you add a particular WebSphere JVM to an application, the Host it is running on as well as the database it communicates with are also added.
- ▶ Click the **Applications** link in the breadcrumb to go back to the Applications window.
- ▶ Double-click the application you want to monitor, and enter the Diagnostics views.
- ▶ You may want to check that the correct view groups are shown/hidden for each application and make any changes necessary. You can right-click in the View bar and select **Open a View Group** to see hidden view groups.
- ▶ At this point, when users select an application they will be viewing application-specific data in the UI.

Returning to the Applications Window

Once you are in the Diagnostics views, the breadcrumb trail shown at the top shows you the context for the information that is presented in the view. You can return to the **Applications** window from the Diagnostics views by clicking the **Applications** level in the breadcrumb trail.



The first level displayed in the breadcrumb shows the application context of the Diagnostics views. If you select to view the **Entire Enterprise**, you see All Applications in the Applications window. If you select a specific application, you see the name of the application. Selecting the first-level link in the breadcrumb takes you back to the Applications window.

The last level displayed in the breadcrumb is not a link because it represents the current Diagnostics view.

Entities and Parent Entities Added to an Application

You can associate the following entities with an application:

- Hosts
- Probes (Java, .NET, Oracle, SAP, SQL Server, MQ)
- SQL Statements
- Server Requests
- Web Services
- Aggregate Requests

- ▶ Outbound Calls
- ▶ Portlets
- ▶ Trended Methods
- ▶ Probe Connections
- ▶ MQ Queues
- ▶ MQ Channels
- ▶ Resources
- ▶ Collections

When you add an entity to an application, the 'parent' entities are also added. Some examples of this rule are shown below:

- ▶ If you add a probe to an application, the associated host and connected probes (and the connections) will also be added.

If you selected the **Add probe contents to application** checkbox when creating the application, the UI shows probe contents, the entities do not need to be added to application to be displayed in the UI. See "How to Create/Edit Applications" on page 54 for descriptions of the discovery policy checkboxes. Connected probes are not added when the **Add all Connected probes to application** checkbox is de-selected when creating the application.

- ▶ If you add a server request to an application, the probe is also added (which adds the host the probe runs on, and the connected probes, which adds their hosts and their connected probes, and so on).
- ▶ If you add a Web service to an application, the associated Web service operations are added and the connected services are also discovered (which includes probes and the host the probe data collection runs on, and so on).
- ▶ If you add an aggregate request to an application, any server requests that match it are also added. When the server requests are added, the probe they are associated with is added, which results in the hosts being added, and so on.

- If you add a trended method to an application, nothing else is added unless it is a trended method associated with a probe and then the probe (and host and so on) is also added.
- If you add a probe topology link (probe connection) to an application, the endpoint (probe) is automatically added (and the probes that probe points to).
- If you add an MQ queue or MQ channel to an application, the MQ queue manager is also added.
- If you add a resource or collection to an application, the probe associated with it is also added (which adds the hosts, and so on).
- If you add an Outbound call, the host and probe are also added.
- If you add a portlet, the host and probe are also added.

Note: When you have Business Process Monitors running in your environment, Diagnostics automatically generates an application for each Business Process Monitor application (in Business Process Monitor 8.x the BPM Profile application is created) and related synthetic transactions.

Working in the Application Context

If you select an application in the initial Applications window, the data displayed in the Diagnostics Views is specific to this application (and the entities associated with it).

Application Context in the View Sidebar

When you enter the Diagnostics views for a selected application, the View sidebar does not show any of the custom view groups (for example, BEA WebLogic or IBM WebSphere). By default, when you create an application, the only views that appear in the View sidebar are the Standard Views and shared views. You can right-click the View sidebar to select **Open a View Group**.

Any view (or view group) that is created in the context of a particular application is shared with all other users viewing that application. This shared viewing is different from viewing the Entire Enterprise, where any views or view groups created are private to the user who created them.

View Entities Under Probes in an Application Context

Only entities that were added to an application appear when you are viewing that application. When drilling down from a probe, however, you see everything running on that probe. For example, if you drill down from a probe to the server requests for that probe, you see all server requests. If any server requests have been added to the application explicitly, those server requests have a check mark in the **App?** column of the graph entity table.

View Layers

When viewing layers for an entity, no filtering by application is performed.

Custom Views

When a custom view is created within an application (excluding the Entire Enterprise default application), the custom view is available to all users with permissions to view the application.

Tasks

How to Create Application Groups

When you create an application in Diagnostics it functions as a container for other non-application entities. These containers enable you to group entities, and to associate them with an application. For example, you may want to separate the entities used to implement and support the Human Resources application from the entities used for the CRM application.

If the user you are logged in as, has the appropriate permissions, you can create a new application group in the top-level application group named **Applications**. You can also create a new application group within another application group.

This task describes how to create an application group.

To create an application group:

- 1** Select the default application group named Applications in the navigation pane, or select an existing application group from the tree in the navigation pane.
- 2** Then select **Create a new application group**. Alternately you can select the top-level Applications group or another application group in the navigation pane and right-click to select **Create New Group**.
- 3** Enter a name for the application group.

Example:

The Create New Group dialog box is shown below:



How to Create/Edit Applications

If the user you are logged in as has the appropriate permissions, you can create a new application or edit the properties of an existing application.

What actions the user or role is allowed to perform in the Diagnostics UI is determined by a combination of the following:

- ▶ User and role permissions set for applications in the Create/Edit Application dialog box. For details on setting view, edit or edit screens permissions for applications, see "Application Permissions" on page 63.
- ▶ User and role permissions set for the enterprise server and probes in the Security dialog box, accessed from the Diagnostics welcome screen by selecting Configure Diagnostics and then selecting Security. See the *HP Diagnostics Installation and Configuration Guide* for details about setting user permissions at the enterprise or probe level.

To create an application:

- 1** First select an application group, either the default application group named Applications in the Applications Window's navigation pane, or select an existing application group from the tree in the navigation pane.
- 2** Then, in the Application Group view, select **Create a new application**. Alternately you can select the top-level Applications group or another application group in the navigation pane and right-click to select **Create New Application**.

- 3** The Create New Application dialog box is displayed as shown in the example below.

Example:

The Create (Edit) New Application dialog box is shown below.

Create New Application

Name

Description

Path

Permissions

User or Role

User or Role	Edit Screens	Modify	View
(any_diagnostics_user)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Discovery Policies

Add all connected probes to application

Add probe contents to application

Name. Enter a name for the application you want to create.

Description. Enter a description for the application.

Path. The path associates the application with an application group. The path contains the application group name you selected when creating the application. If you want to move (not copy) an application from one group to another, you can change the path to a different application group name.

Permissions. Define permissions for the application.

User or Role. Select or add users and roles to associate with this application, and associate permissions to each user and role. You can select from a drop-down list of existing users and roles or you can add a new user or role. The users and roles you select are allowed access to this application's data in the Diagnostics views. For each user or role, select the permissions allowed:

Edit Screens. Make edits using the Application Overview screen.

Modify. Delete applications, rename applications, modify applications, and add or remove an entity from an application.

View. View applications, and edit application properties (also requires Enterprise permission set to change).

Discovery Policies. Select special discovery policies to be applied to the application.

Add all Connected probes to application. When checked (which is the default) the discovery process will add additional probes to the application based on connections to the probes that are currently members of the application.

If you add a probe to an application, the associated host and child entities such as connected probes and their connections will also be added. Note: if you add a host, related probes are not added because this discovery policy is for when a probe is added not when a host is added.

The discovery process runs every five minutes to see which entities are present in the application and to determine the status and count. See "Entities and Parent Entities Added to an Application" on page 44 for details on what is automatically added to the application during discovery.

You can disable discovery of probes within this application's topology by de-selecting this checkbox. When de-selected, discovery will not add probes to the application based on the connections to currently defined probe members. You can manually add the probes you want to be included in this application.

Alternately you could allow discovery to add all the connected probes and then use the Remove/Exclude feature, but this won't prevent new probes from being automatically added to the application.

Add probe contents to application. When checked (which is the default), the UI shows probes and everything below the probe when in the context of the application. For example, probe contents includes server requests, Web services, outbound calls and portlets.

You can disable display of probe contents by de-selecting this checkbox. When de-selected, the UI will not display entities below the probe. So in the application context, for example, you would not see any server requests in the Server Request view unless you added them directly to the application. You may want to de-select this checkbox if your probe monitors two different applications and you do not want to see all the server requests for both applications when in the context of one of the applications.

After you add a new application, you are able to add entities (for example, probes and server requests) to this new application from within the Diagnostics view. For details on adding and removing entities from an application, see "How to Add/Remove Entities from an Application" on page 58.

To edit an application:

- 1** If the user you are logged in as, has the appropriate permissions, you can edit an application's properties by selecting the application in the left pane, and right-clicking **Edit Application**.
- 2** In the (Create) Edit Application dialog box, you can change the application's name, description, path, user, and role. To change permissions for a user or role for this application, check or uncheck the boxes for Edit, Modify, or View. Or check or un-check discovery policies.

How to Delete an Application

This task describes how to delete an application.

To delete an application:

- 1** In the navigation pane of the Applications window, right-click an application and select **Delete Application**.
- 2** After all the applications in an applications group have been deleted, the application group is deleted the next time you exit the Diagnostics UI.

How to Add/Remove Entities from an Application

In various Diagnostics views, you can associate entities with defined applications by selecting the entity, and then right-clicking **Add to application**. The user you are logged in as, must have the appropriate permissions for you to be able to add or remove entities from an application.

Note: Whenever you add or remove entities from an application it may take several minutes for the changes to take effect and for you to see the change.

This task includes the following topics:

- "Entities you can Associate with an Application" on page 58
- "Add to Application" on page 59
- "Remove or Exclude from Application" on page 60

Entities you can Associate with an Application

You can associate certain entities with an application. When you add an entity to an application, the 'parent' entities are also added. For a list of entities you can add to applications, see "Entities and Parent Entities Added to an Application" on page 49.

Add to Application

In a Diagnostics view, select an entity in the entity table and right-click to select **Add to Application** and the Select Application dialog box is displayed. See the example below. You select an application from the list and the entity will be added to that application.

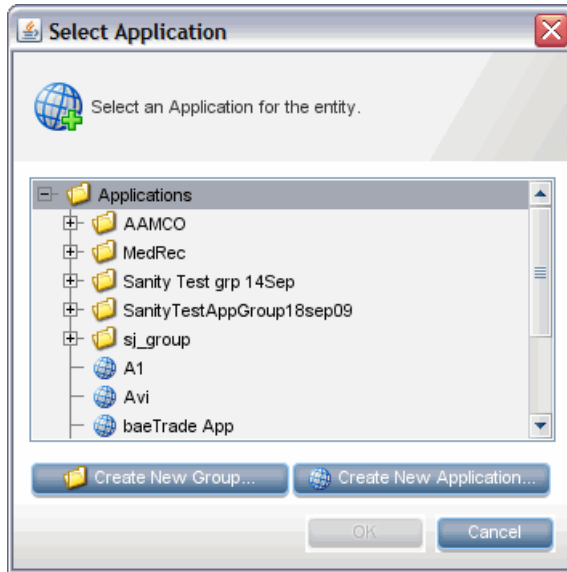
An entity can be added to any number of applications. When adding an entity to another application, right-click to select **Add to Another Application**.

You can select multiple entities in the graph entity table to add or remove them from an application. Typically, when selecting multiple entities you should select groups of entities you want to remove or add to the same application.

If the user you are logged in as, has the appropriate permissions, you can also select **Create New Group** and **Create New Application** from this dialog box. When you create a new group or application, that group or application will automatically be selected in the dialog box.

Note: If you create a new application (or multiple applications) from this dialog box, but then cancel out of the dialog box, the created applications will still exist.

Example:



Remove or Exclude from Application

Selecting **Remove from Application** from the entity right-click menu or from the common Tasks menu displays the Remove from Application dialog box. You can select an application to remove or exclude the entity from. The user you are logged in as, must have the appropriate permissions for you to be able to remove or exclude entities from an application.

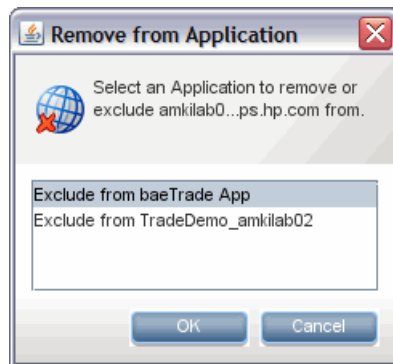
The easiest way to remove entities you do not want is from the Topology view. Select the entity in the Topology view and right-click to select **Remove from Application**. When you multiselect many entities then select **Remove from Application**, the message displayed (in hover text) is for a single entity. However what will actually be removed is - all selected entities from the application. To avoid confusion you would most likely use the multi-line remove when you want to select everything and remove from a specific application.

You cannot remove or exclude an entity if it is a parent of an entity that was manually added. For example, if you add the server request **/login.do**, you will not be able to remove or exclude the probe where this server request is running since this is the parent of the server request entity. This limitation exists to ensure that you will always be able to navigate to anything you have added to your application.

In the Remove from Application dialog box you may see Remove from or Exclude from actions. Remove will remove the entity and also any entities that were added to the application because of their relationship (parent/child) to the removed entity. Exclude will just remove the entity. In essence, the action is the same, but the result depends on the way the entity was added to the application initially.

You can remove entities from applications in either the Entire Enterprise or a selected application context.

Example:



How to Select an Application

You can select an application in the Application window as follows:

To select the pre-defined application Entire Enterprise:

- 1 Click Entire Enterprise in the navigation pane.

- 2 Double-click Entire Enterprise to open the application summary view or select an entity from the **Application Contents** list to go directly to the related Diagnostics view. Diagnostics views shows data for all components known to Diagnostics, allowing you to monitor the entire enterprise.

To select an application from an application group:

- 1 In the navigation pane, open the application group named **Applications** and navigate to select the application you want to monitor.
- 2 Double-click the application to open the Application Explorer view or select an entity from the Applications Contents list to go directly to the related Diagnostics view. Diagnostics views are filtered to show only the entities that are part of the selected application.

Reference

Application Permissions

By default, any user with execute permissions can create an application (or application group) in Diagnostics.

After an application has been created, permissions for the application can be assigned to users or roles.

See the table below for details about the application permissions (view, edit, or edit screens) required to perform various actions, such as creating or renaming an application.

Application permissions are set in the Applications window. See *HP Diagnostics Installation and Configuration Guide* (Appendix B) for details on setting user permissions at the enterprise or application level.

Area and Action	Enterprise Permissions			Application Permissions		
	view	execute	change	view	execute	change
Create application		x				
Delete application		x			x	
Rename application		x			x	
Change application path		x			x	
Modify application permissions		x			x	
Edit custom applications screen	x					x
Add entity to application	x				x	
Remove entity from application	x				x	

Area and Action	Enterprise Permissions			Application Permissions		
	view	execute	change	view	execute	change
Edit entity properties (thresholds, comments, etc.)		x		x		
View application	x			x		

Applications User Interface

This section describes:

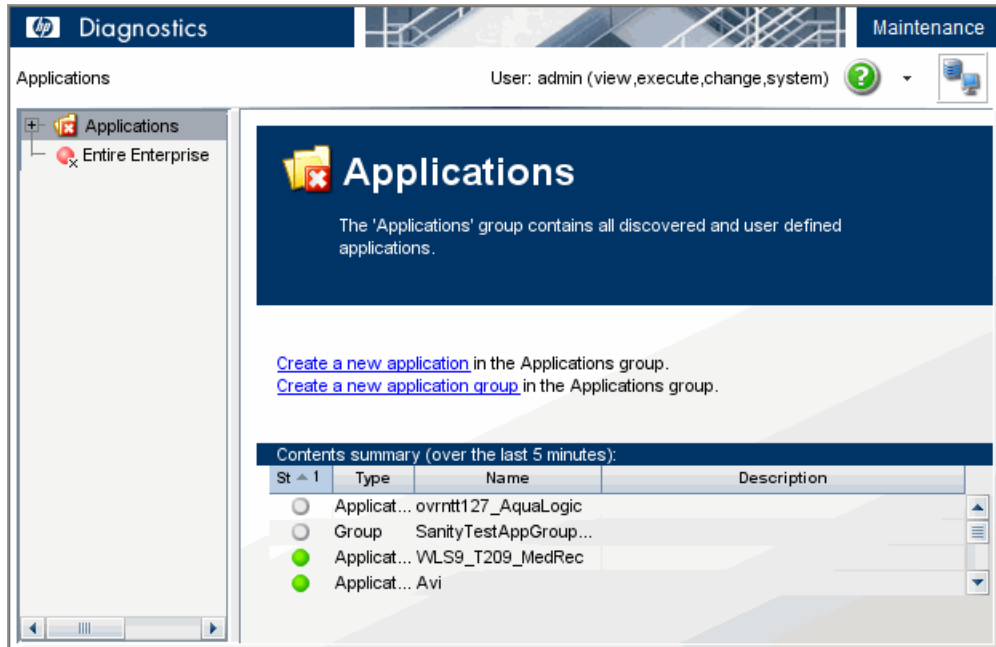
- ▶ Applications Window Description on page 64

Applications Window Description

The Diagnostics UI displays performance data for monitored applications. In the initial Diagnostics Applications window you define and then select the application context for viewing data in the Diagnostics UI.

From the Applications window, you can select an existing application and go to a Diagnostics view in the context of the selected application. You can also select the Entire Enterprise application and view all data displayed in the Diagnostics views. And from the Applications windows, you can create, update, and delete applications.

The following is an example of the Applications window.



<p>To access</p>	<p>The Applications window is the first screen you see when you select Open Diagnostics in the main Diagnostics Server UI window.</p>
<p>Important information</p>	<p>The Applications window is only available when running Diagnostics by itself or when integrated with Business Availability Center, and not when viewing Diagnostics from LoadRunner or Performance Center.</p> <p>You will only see the application groups that contain applications that the user you are logged in as, has permissions to view.</p>

Relevant tasks	<ul style="list-style-type: none"> ➤ See "How to Create Application Groups" on page 53 for information on how to create or edit applications ➤ See "How to Create/Edit Applications" on page 54 ➤ See "How to Delete an Application" on page 58 ➤ See "How to Add/Remove Entities from an Application" on page 58 ➤ See "How to Select an Application" on page 61 for how to select an application in the Applications window
See also	<ul style="list-style-type: none"> ➤ See "Application Permissions" on page 63 ➤ See "Using the Applications Window" on page 46 ➤ See "Status Indicators" on page 517

Navigation pane

The navigation pane lists the application groups and applications defined for your enterprise. You use the navigation pane to select the application context for viewing data in the Diagnostics UI.

To view Diagnostics data in the context of an application, select the application in the navigation pane. As a result, the Diagnostics views are filtered to show only the entities that are part of the selected application.

The following user interface elements are included:

UI Elements	Description
<Status>	A status icon is displayed next to the application group or application.
Applications	The default application group, named Applications, contains user-defined application groups and applications that are detected automatically.
Entire Enterprise	The pre-defined application, named Entire Enterprise, contains all entities known to Diagnostics. Select the Entire Enterprise application to monitor the entire environment. As a result, data in the Diagnostics views is not filtered by application.

Application Group Details

By default, application group details for all groups is displayed when you enter the Applications window.

Selecting an application group in the navigation pane displays application group details for that group in the Contents Summary table. You can also double-click an application group in the navigation pane to display application group details for that group.

The contents summary is for the last five minutes.

The following user interface elements are included:

UI Elements	Description
<p>Create a new application group</p> <p>Create a new application</p>	<p>If the user you are logged in as has the appropriate permissions, you can Create a new application group and Create a new application from the Application Group view.</p>
<Status>	<p>The status is displayed next to the application group in the Contents summary table.</p>
<p>Contents Summary table</p>	<p>Contents Summary lists the applications in the application group selected in the navigation pane.</p> <p>The following information is displayed in the Contents Summary:</p> <p>Status. Indicates how the application is performing. The status of an application is based on the status of the entities that comprise the application.</p> <p>Type. The type can be either application or group (a group contains other groups or applications).</p> <p>Name. User-specified name of the application.</p> <p>Description. User-specified description of the application.</p> <p>Double-clicking on an application in the Contents Summary table drills down to display details for the Application Contents.</p>

Application Details

Selecting a application (or the Entire Enterprise application) in the navigation pane displays details for that application in the Application Contents.

The following user interface elements are included:

UI Elements	Description
<p>application explorer</p> <p>Application summary</p>	<p>If you select an application, you will see a link to the application explorer, which (by default) takes you to the Application Explorer view in Diagnostics. Double-clicking on an application also opens the Application Explorer.</p> <p>If you select Entire Enterprise, you will see a link to the Application Summary (same as Server Summary) view. Double-clicking Entire Enterprise also opens the Application Summary view.</p>
<p>application metrics</p>	<p>A link to application metrics, which takes you to the Application Metrics view in Diagnostics for high level details about an application.</p>

UI Elements	Description
<Status>	<p>The status is displayed next to the application entity type (for example, Hosts, Java Probes, Outbound Calls) in the Application Contents list. Status is based on the last five minutes.</p>
<p>Application Contents list</p>	<p>The Application Contents list shows the entities that are part of the selected application. It shows which entities were active in the last five-minute interval. Server requests or transactions may not be included if they were not active during this time period.</p> <p>A discovery process runs every five minutes to see which entities are present in the application and to determine the status and count, which is displayed in the Application Contents as a red, yellow or green status indicator. If new entities appear, they are added to the list. If entities are not found, they are removed from the list.</p> <p>To open a specific Diagnostic view, select the link for an entity displayed in the details pane. For example, selecting the Java Probes link opens the Probes view. If the application is configured to not show the view group which contains the view for this entity type, the link will not be selectable.</p> <p>If you've selected the Entire Enterprise application, all data is shown in the Diagnostics views. If you've selected a single application, the Diagnostics views show data in the context of the selected application.</p>

3

Common Controls in the Diagnostics UI

This chapter describes the common controls in the Diagnostics UI.

This chapter includes:

Concepts

- ▶ Common Diagnostics UI Controls on page 73
- ▶ View Context Drop-down Menu on page 75
- ▶ Monitor and Analyze Tabs on page 76
- ▶ Panning, Pausing and Zoom Controls on page 77
- ▶ Viewing Time Filter on page 78
- ▶ Diagnostics Toolbar on page 81
- ▶ View Toolbar on page 81
- ▶ Breadcrumb Trail on page 82
- ▶ View Bar on page 83
- ▶ About Different View Layouts on page 86
- ▶ Resizing and Docking Windows on page 92
- ▶ About the Different Display Panes in a View on page 95
- ▶ About Graphs (Chart) on page 97
- ▶ About View Filters on page 100
- ▶ Multiple Graph Layout Options on page 101
- ▶ Entities and Metrics Charted in the Graph on page 102
- ▶ The Data Behind a Charted Metric on the Graph on page 104

- ▶ Zoom Features in the Graph on page 107
- ▶ About the Graph Entity Table on page 108
- ▶ About Status Indicators on page 110
- ▶ Additional Information about the Entities in the Table on page 112
- ▶ Table Header Controls on page 114
- ▶ UI Properties on page 116
- ▶ About Common Tasks and Navigations on page 117
- ▶ Common Tasks on page 119
- ▶ Common Navigations on page 125

Tasks

- ▶ How to Save or Export Data on page 128
- ▶ How to Work with View Filters on page 131
- ▶ How to Zoom In and Out on a Graph on page 137
- ▶ How to Customize Table Columns on page 141
- ▶ How to Customize Table Sort Order on page 144
- ▶ How to Find an Entity in the Table on page 146
- ▶ How to Select Multiple Rows in the Table on page 147

Concepts

Common Diagnostics UI Controls

The features and controls that are common to most of the Diagnostics views are described below using the following annotated screen image:

This section includes:

- "View Context Drop-down Menu" on page 75
- "Monitor and Analyze Tabs" on page 76
- "Panning, Pausing and Zoom Controls" on page 77
- "Viewing Time Filter" on page 78
- "Diagnostics Toolbar" on page 81
- "View Toolbar" on page 81
- "Breadcrumb Trail" on page 82
- "View Bar" on page 83
- "About Different View Layouts" on page 86
- "Resizing and Docking Windows" on page 92

Other topics in this section describe the different panes commonly seen in the Diagnostics views and how to work with the controls in these panes.

- "About the Different Display Panes in a View" on page 95
- "About Graphs (Chart)" on page 97
- "About the Graph Entity Table" on page 108
- "About Common Tasks and Navigations" on page 117

Chapter 3 • Common Controls in the Diagnostics UI

The screenshot displays the Oracle Diagnostics UI with the following components labeled:

- View Context Drop-Down:** Located at the top left, showing 'Monitoring'.
- Monitor and Analyze Snapshots Tabs:** Located at the top, including 'Monitor and Investigate' and 'Analyze Snapshots'.
- Panning, Pausing and Zoom:** Located in the top toolbar area.
- Diagnostics Viewing Time Filter:** Located in the top toolbar, showing 'Viewing Last 5 minutes'.
- Diagnostics Toolbar:** Located in the top right area.
- View Toolbar:** Located in the top right corner.
- Breadcrumb Trail:** Located on the left side, showing 'Application Metrics (Entire Enterprise) > Standard Views - Server Requests'.
- View bar:** Located on the left side, showing 'Views' and 'Server Requests'.
- View Specific Information:** Located on the left side, showing a list of views like 'Standard Views', 'Transactions', 'Portals', etc.

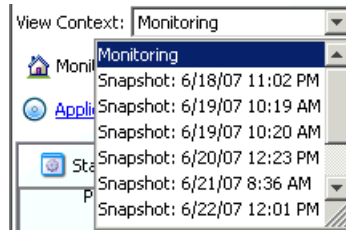
The main content area shows a chart titled 'All Server Requests in All Probe Groups filtered by No Filter with Top 6 by Latency Over Threshold charted for Last 5 minutes'. The chart displays latency (Avg) over time, with a threshold line at 2.3s. Below the chart is a table of server requests:

Status	Color	Chart?	Server Request	Pro.	Late.	Thror	CPU (A...	Info
Green	Green	✓	HelloPOA_invoke(so...	ovr...	2.3 s	8 / ...	1.1...	
Green	Red	✓	HelloPOA_invoke(ut...	ovr...	1.8 s	9 / ...	1.1...	
Green	Yellow	✓	_HelloStLib.someOpe...	ovr...	2.3 s	8 / ...	1.8...	
Green	Blue	✓	_HelloStLib.otherOpe()	ovr...	1.8 s	9 / ...	1.8...	
Green	Brown	✓	/CallChain/WebApp/C...	L81...	3.8 s	25 / ...	25...	
Green			CallChain2ImpiServic...	WL...	3.0 s	36 / ...	0.0...	
Green			CSessionBean.callM...	L81...	212...	25 / ...	0.0...	
Green			CallChain2ImpiServic...	WL...	106...	72 / ...	0.0...	
Green			CallChain2ImpiServic...	WL...	106...	24 / ...	0.0...	
Green			CallChain2ImpiServic...	WL...	191...	6 / ...	48...	
Green			CallChain2ImpiServic...	WL...	46...	12 / ...	0.0...	
Green			CallChain2ImpiServic...	WL...	112...	5 / ...	0.0...	
Green			TraderService::sell	L81...	992...	12 / ...	0.0...	

The right side of the interface includes a 'Common Tasks' panel with actions like 'Creates New Snapshot and Add', 'View Profiler for ovrux28-jre16-c...', 'Add to application...', 'Creates Alert Rule...', and 'Create Comments...'. Below that is a 'Navigations' panel with 'View Layers (1)' and 'View Probes (1)'. At the bottom right is a 'Details' panel showing information for 'HelloPOA_invoke(someOpe...' including arguments, default name, method, package, root method, server request, signature, and count.

View Context Drop-down Menu

The **View Context** drop-down allows you to control whether the views in the Monitor and Investigate tab are in Monitoring mode or in Snapshot Analysis mode. In Monitoring mode, the views contain performance metrics for the current processing of your applications. In Snapshot Analysis mode, the views contain performance metrics for a selected snapshot.



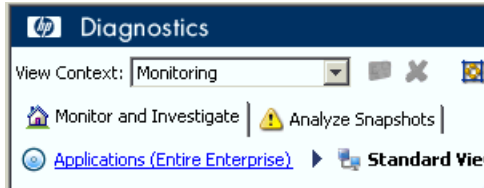
To view performance metrics for your application's current processing, select **Monitoring** from the **View Context** drop-down. Selecting **Monitoring** places Diagnostics in Monitoring mode. When in monitoring mode, the Monitor and Investigate tab presents the Diagnostics views with the application's performance metrics for the time period indicated in the Viewing drop-down.

To view the performance metrics for a particular snapshot, select the snapshot from the **View Context** drop-down. Selecting a snapshot from the **View Context** drop-down puts Diagnostics in Snapshot Analysis mode. When in Snapshot Analysis mode, the Monitor and Investigate tab presents the Diagnostics views with the performance metrics for the time of the selected snapshot.

For details on snapshot analysis, see Chapter 6, "Performing Snapshot Analysis".

Monitor and Analyze Tabs

Diagnostics displays its views on two tabs: the Monitor and Investigate tab and the Analyze Snapshots tab. Diagnostics displays different information in these tabs depending on whether you have selected Monitoring mode or Snapshot Analysis mode. The mode is determined by your selection from the **View Context** drop-down.



When in monitoring mode, the Monitor and Investigate tab presents the Diagnostics views with the application's performance metrics for the time period indicated in the Viewing drop-down. The Analyze Snapshots tab is not used when in Monitoring mode and instead contains instructions for creating a new snapshot.

When in Snapshot Analysis mode, the Monitor and Investigate tab presents the Diagnostics views with the performance metrics for the time of the selected snapshot. The Analyze Snapshots tab contains all of the entities and their metrics that you have included in the snapshot. For details on snapshot analysis, see Chapter 6, "Performing Snapshot Analysis".

Panning, Pausing and Zoom Controls

The **Pause** and **Pan** buttons at the top of the Diagnostics interface allow you to stop Diagnostics from updating the views so that you can analyze the data that is currently displayed before it is replaced by more current data. The **Zoom Out** and **Fit Data to Size** buttons allow you to modify the time filter and granularity of the data display.



- Click **Fit Data to Size** to reset any zooming in changes so that all points will be shown on the graph and the scroll bar is no longer displayed.



- Click **Zoom Out** to increment the time filter to the next granularity, for example, five minutes to 20 minutes.



- Click **Pause** to stop charting new data in the graph. The graph remains unchanged when this button is selected.



- Click **Pan Left** to smoothly scroll the graph from the current time period to the time period with the same duration that immediately preceded the one displayed.



- Click **Pan Right** to smoothly scroll the graph from the current time period to a later time period than the one that is currently charted.



- Click **Play** to activate the graph and begin charting the current performance metrics as they become available. The Time Range filter is reset to show the same granularity that you were viewing when you first clicked **Pause**.

Viewing Time Filter

Using the **Viewing Time** filter you can set the time period that Diagnostics uses to display the performance metrics in all of its views. You may select one of the default time ranges, or enter a custom time range.



The time range that you select from the **Viewing Time** filter is used in all of the Diagnostics views unless you have locked a view for a specific time period. See "Filter by Time" on page 134 for details on locking views.

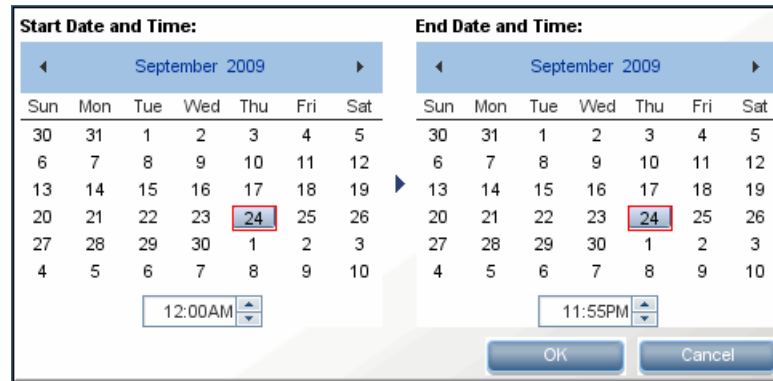
When you select a time period from the **Viewing Time** filter, the information in the graphs, the graph entity tables, and the Details pane of the Diagnostics Views are updated to correspond to the selected time range resolution. Diagnostics continues to update the views so that the most recent information for the selected time range resolution is always displayed.

This section also includes:

- ▶ "Setting a Custom Time Range" on page 79
- ▶ "Chart Update Frequency" on page 79
- ▶ "Understanding Time Range Resolution" on page 80
- ▶ "Using the Viewing Time Filter in Snapshot Analysis Mode" on page 81

Setting a Custom Time Range

When you select the Custom time range filter option two calendars are displayed to allow you to indicate the starting date and time and ending date and time for the metrics that you want Diagnostics to display in the view. The following image shows an example of the Custom time range filter:



After selecting the start and end dates, click **OK** to instruct Diagnostics to refresh the current view.

Chart Update Frequency

When you view the performance metrics, note that the charted metrics are updated at different rates, depending on the selected time range resolution. The following table provides the update frequency for each time range resolution:

Time Range Resolution	View Update Frequency
5 minutes	5 seconds
20 minutes	3 minutes
1 hour	4 minutes
6 hours	8 minutes

Time Range Resolution	View Update Frequency
1 day	11 minutes
1 week or greater	29 minutes

Note: If you are using Diagnostics with LoadRunner or Performance Center, and your selected time range is **Whole Scenario**, the update frequency decreases as the scenario progresses. For example, at the beginning of the scenario, the update frequency is approximately five seconds. However, after an hour into the scenario, the update frequency decreases to approximately 4 minutes.

Understanding Time Range Resolution

The actual time period included in the displayed metrics is not always limited to the time range resolution that you requested.

When you select **Last 5 minutes**, the time range resolution for the performance metrics displayed in the other parts of the view is changed to include only information from the previous five minutes.

However, for any other **Time Range** option, the actual metrics displayed include information for no less than the selected time range. The metrics normally include the information for up to twice the duration of the selected time range.

For example, when you select **Last 20 minutes**, the metrics displayed in the view will include data for the previous 20 to 40 minutes.

You can use the **Fit Data to Size** button to see the full time range in the graph. See "Panning, Pausing and Zoom Controls" on page 77.

Using the Viewing Time Filter in Snapshot Analysis Mode

When you are in Snapshot Analysis mode, Diagnostics sets the **Viewing Time** filter to the time of the selected snapshot. This time is used for the Diagnostics views displayed in the Monitor and Investigate tab and in the Analyze Snapshot tab. You may set the **Viewing Time** filter so that you can see the metrics included in the snapshot at a more current time period without leaving Snapshot Analysis mode.

Diagnostics Toolbar

The **Diagnostics Toolbar** contains the following buttons:



Turn On/Off Active Alert Notifications. A toggle button that allows you to control whether a message is to be displayed in the Diagnostics message box each time an alert notification is sent. For details on alerts, see Chapter 5, "Working with Alerts".



Turn On/Off Tooltips. A toggle button that allows you to control whether tooltips are displayed throughout the UI for your logon user.



Show Help. Gives you access to the user documentation, to information about the version of Diagnostics that you are using, and to the version of some of the third-party components that have been used in the product. Select the desired option from the menu that Diagnostics displays when you click **Show Help**. Select **Help on View** to see a help topic for the view displayed (context sensitive help).



Page Loading Indicator. When Diagnostics is retrieving and formatting the data that is going to display in a view, the **Page Loading** icon displays two blue arrows spinning in a loop.



When the view is refreshed with the formatted data, the arrows are no longer displayed in the icon.

View Toolbar

The **View Toolbar** contains the following buttons:



Reset Screen Layout. After rearranging the panes in the View you can select the Reset Screen Layout button to return the display to the default layout.



Save This View. If you have customized the Diagnostics view, you may want to save the custom view to use later. The **Save This View** button provides a convenient way for you to save your changes. For details on custom views, see Chapter 7, "Working with Custom Views".



Saves this view as an HTML Web Page or CSV file. This selection allows you to either:

- ▶ Save the currently displayed metrics to an HTML Web page that can be recalled for viewing or sharing with others.
- ▶ Save the currently displayed chart (graph) to a CSV file that can be viewed and formatted in Microsoft Excel.

For details on how to save view information as an HTML Web Page or CSV file, "How to Save or Export Data" on page 128.

Breadcrumb Trail

The breadcrumb trail at the top of most of the Diagnostics views provides a convenient way to determine the context for the information that is presented in the view. It also provides navigation to retrace your steps in your performance analysis. Clicking a level in the breadcrumb trail takes you back to the view for the selected level.

The first level displayed in the breadcrumb shows the application context of the Diagnostics views. You will see either Applications (Entire Enterprise) if you selected to view the Entire Enterprise or the name of the application you selected in the Applications window. Selecting the first level link in the breadcrumb takes you back to the Applications window.

The last level displayed in the breadcrumb is not a link because it represents the current Diagnostics view.

For additional help on drill-down navigation from one view to another, see "Common Navigations" on page 125.

 **View Bar**

The **View Bar** contains view groups which in turn contain the views that you use to analyze the performance of your applications. The views are organized into view groups to make it easier for you to locate views of interest.

Predefined view groups that are installed with Diagnostics contain views intended to assist you in diagnosing problems in particular situations or for specific environments. The **Standard Views** view group contains views that are useful in most situations. These views are described in Part III, "Diagnostics Standard Views."

The other view groups contain views that are useful in more specific situations. These views are described in Part IV, "Understanding Diagnostics Specialized Views." All of the view groups, except for **My Views** (or Shared Views when opened in context of an application), contain views that have been predefined to provide you with the ability to look at your application's performance information and begin analyzing the information the first time that you use Diagnostics.

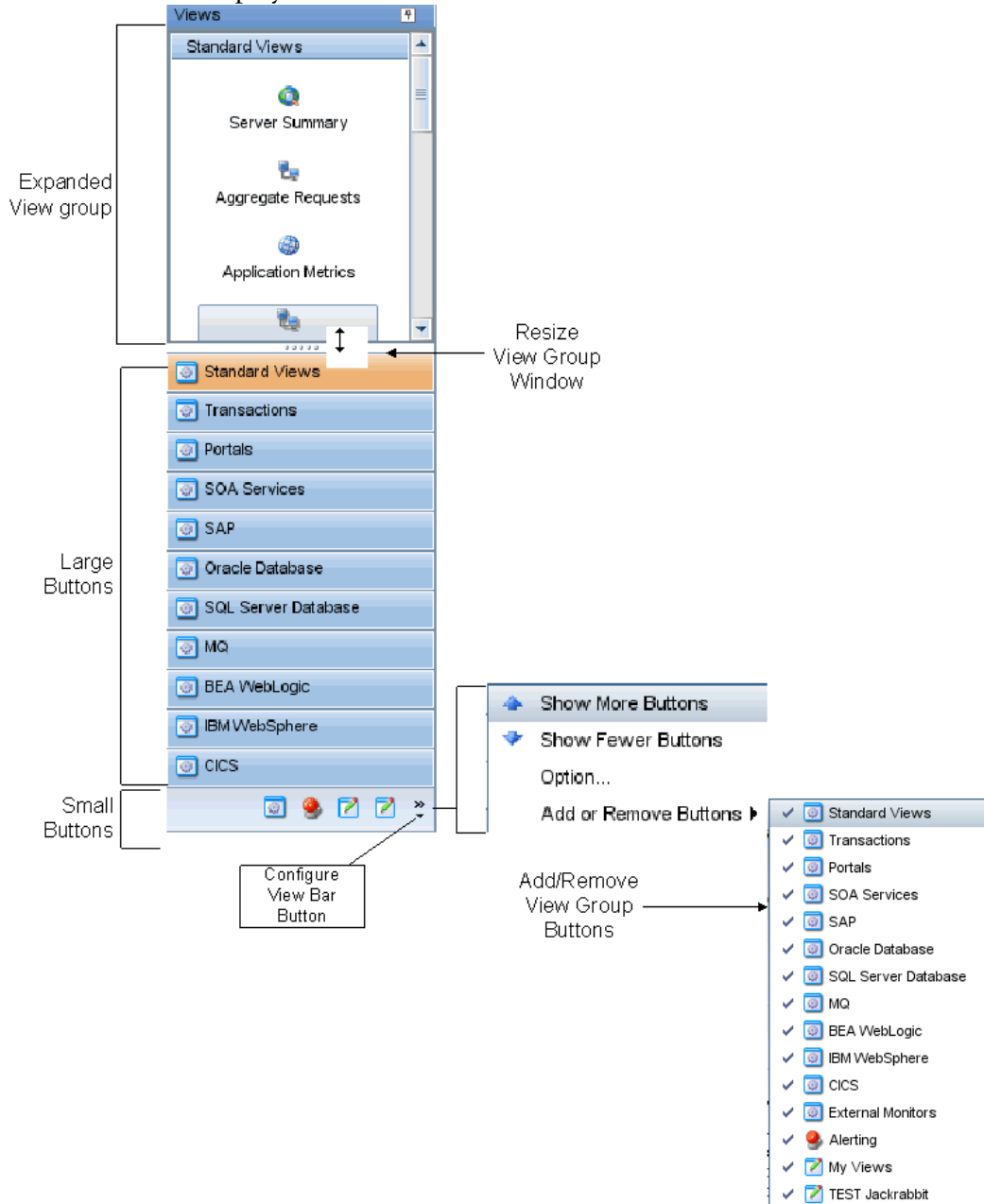
Note: You can right-click in the View Bar and select **Open a View Group** to unhide additional view groups. Or you can use the button at the bottom of the View Bar to show more or fewer view groups.

You may save customized versions of the predefined views or create original views using the functions of the View Bar. The My Views group (or Shared Views when opened in context of an application) contains your customized views. For information on working with customized views and view groups, see Chapter 7, "Working with Custom Views".

Selecting View Groups in the View Bar

Buttons for each view group are displayed in the View Bar. Selecting a view group (button) in the View Bar displays the view group at the top of the Views pane. The view group is expanded to show a list of related views. You can scroll down to see all the views listed in a view group. Selecting one of the views in the view group, displays that view.

In the View Bar, you can change the number and size of the view group buttons displayed.



To increase space in the View Bar, you can resize the expanded view group and change from large buttons that are displayed vertically to small buttons that are displayed horizontally across the bottom of the View Bar.



Point to the horizontal splitter bar in the expanded view group. When the pointer turns into a double-headed arrow, drag the splitter bar up or down. If you drag the horizontal splitter bar towards the bottom of the View Bar the large buttons are converted to small buttons and appear in a single row at the bottom of the pane.



Select the Configure Buttons button at the bottom of the View Bar to see a pop-up menu allowing you to show, order, add or remove view groups as large buttons displayed in the View Bar.

The menu items are:

- ▶ **Show More Buttons.** Displays an additional view group as a large button in the View Bar. Shortens the display of the view group at the top, as needed, to allow the display of an additional view group as a large button.
- ▶ **Show Fewer Buttons.** Displays one fewer view group as a large button in the View Bar. Takes advantage of more room in the View Bar to lengthen the display of the view group at the top.
- ▶ **Option...** Displays a dialog box to allow you to select the view groups to display as large buttons and the order in which to display them.
- ▶ **Add or Remove Buttons.** Gives you a pop-up list of view groups to check or uncheck for display as large buttons in the View bar.

(These options should not be confused with the Open and Close a View Group menu items available from the view bar. Those hide or unhide view groups, whereas these options move between small and large buttons.)

About Different View Layouts

Diagnostics comes with a set of predefined **Standard Views** where each view focuses on a specific aspect of your application's performance.

Diagnostics also provides **Specialized Views** focused on specific types of applications such as Oracle Databases or BEA WebLogic.

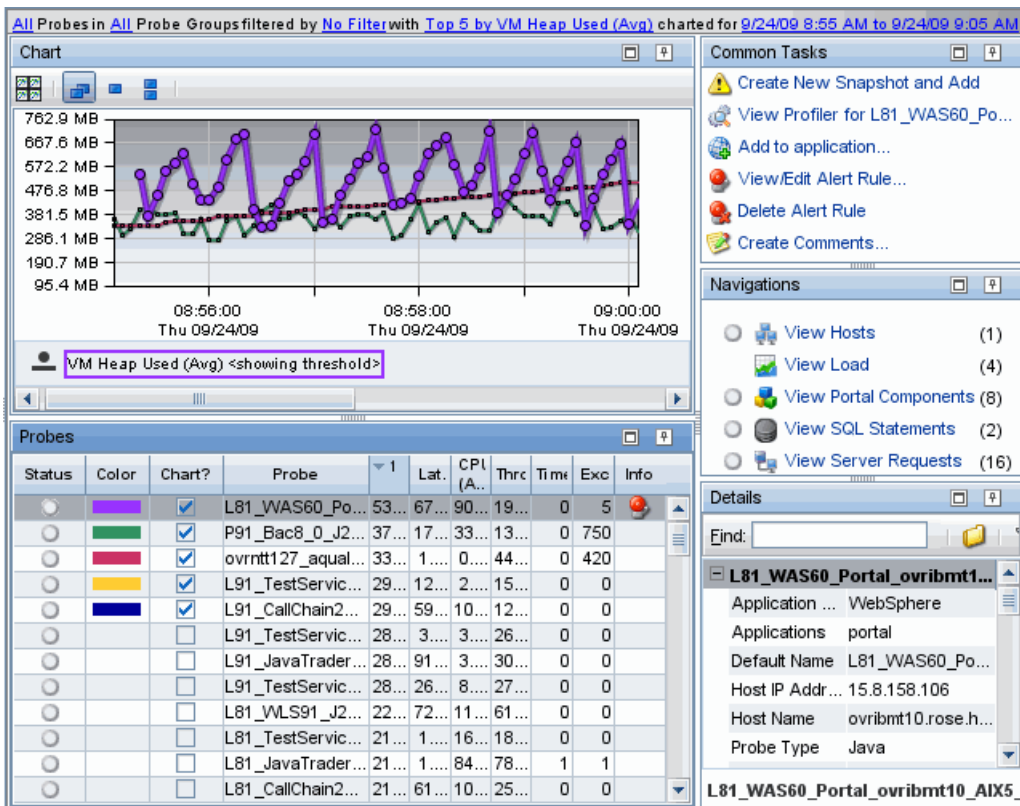
The views show the performance metrics in both tabular and graphical formats. Diagnostics presents the performance metrics for your applications in a variety of view layouts including: detail layouts, dashboard layout, status layout, call profile layout, and topology layout. Some examples of the different view layouts are shown on the following pages.

Detail Layout

The Detail layout has a standard graph, table and details.

Diagnostics views using the detail layout include the Load view, Probes view, and Server Requests view. From many of the views, you can drill down on one particular entity to other detail layout views for that entity.

The Probes view shown below is an example of a detail layout view:



For information on the features and controls in the detail layout, see "About the Different Display Panes in a View" on page 95.

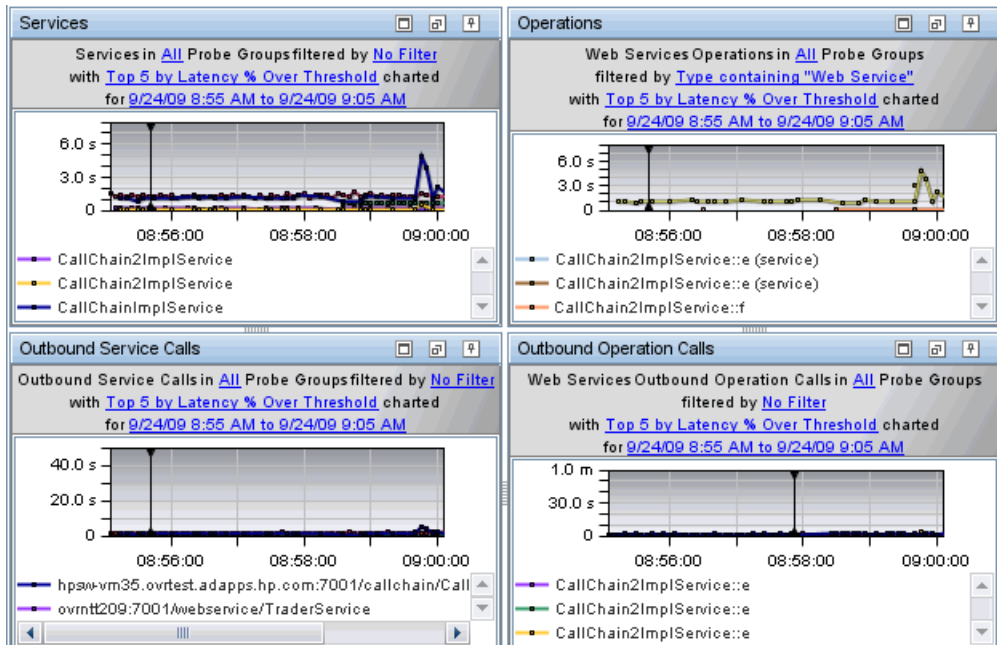
For information about using a specific standard detail layout view, see Part III, "Diagnostics Standard Views."

Dashboard Layout

The Dashboard layout views present key performance characteristics of your applications at a glance.

The dashboard layout views can be made up of two or more views. In a view with a dashboard layout, selected views are displayed in an abbreviated format known as a concise view. You can create customized Dashboard layout views, or use one of the predefined views installed with Diagnostics.

The SOA Services - Service Summary view shown below is an example of a predefined view with a dashboard layout:



A standard or custom view opens in a dashboard as a concise version of the view, where only the graph and the graph legend are included. You may manipulate the view filters that are included in the concise view title just as you can in the detail layout view. You can also see the tooltips for the nodes of the charted trend lines and the edges of the stacked areas by holding your mouse pointer over the appropriate location in the graph.

When you hold your mouse pointer over the items in the legend in the concise views, Diagnostics displays a tooltip with additional information.

To see the full-size version of a concise view, double-click the concise view. Diagnostics displays the full-size version of the detail layout view that you selected and sets the breadcrumb to indicate that you navigated from the Dashboard view.

Status Layout

The Status View is a table displaying performance information for entities such as the probe groups, their probes, and the hosts for those probes. For each level in the table, Diagnostics provides a status of each component compared to the thresholds that you set for the performance metrics. In addition, you can see the performance trends for each component. The trending indicator shows whether metrics performance trend is going up or down and at what rate.

The Status view shown below is an example of a predefined view with a status layout:

Status for Last 5 minutes baselined against Last 6 hours							
St. ▾ 1	Probe Group	Diagnostics Server			Info		
+	Sanity_CallChain_WebService_ovruxd10	server-ovruxd10.rose.hp.com					
+	Sanity_LR_8_1_ovrntt154	server-OVRNTT154					
-	Sanity_LR_9_1_hpswros018	server-HPSWROS018					
St. ▾ 1	Probe	VM Heap Used	Latency	Throughput	Timeouts	Exceptions	Info
⊗	L91_JavaTrader2.WebClie...	194.0...	4 m, ... ↑	132 / hr	11	0	
⊗	L91_JavaTrader2.WebClie...	277.8...	898.1 ... ▢	179 / ... ↑	0	0	
⊗	L91_CallChain2_0.NET_OV...	184.6...	1.0 s	12 / hr ↓	0	0	
⊗	L91_TestService2.WebClie...	186.1...					
⊗	L91_TestService2.WebSer...	190.2...					
⊗	L91_TestService2.WebSer...	187.4...					
⊗	L91_TestService2.WebSer...	274.0...	3.6 ms ▢	18 / min ↑	0	0	
⊗	L91_CallChain2_0.NET_OV...	277.1...	603.9 ... ▢	79 / min ↑	0	0	
⊗	L91_TestService2.WebSer...	277.1...	41.5 ms	18 / min ↑	0	0	
⊗	L91_MSPetShop_1_1_NET_...	43.6 ...	1.6 ms	12 / min ▢	0	0	
⊗	L91_WAS60_Plants_oxyc...	136.6...	7.8 ms	277 / ... ↑	0	0	
⊗	L91_TestService2.WebClie...	275.2...	16.5 ms	72 / min ↑	0	0	
St. ▾ 1	Host	CPU	Disk Bytes/S	Network Bytes/S	Page In/S	Memory	Info
⊗	ovresx1-vm10	8%	9.1 KB	107.4...	0	32%	
⊗	oxycontin.rose.hp.com	2%	4.8 KB	495.9...	51	96%	
⊗	ovresx2-vm3.ovrtest.adapp...	7%	10.3 KB	56.6 KB	0	32%	

For details on the features and controls of status views, see Chapter 23, "Status View".

Topology Layout

The Topology View displays a pictorial graph of the network topology for your monitored applications. The graph shows the links between the components that make up your deployment, along with the monitoring information that depicts the performance of the components and their business processes. For details on topology views, see Chapter 8, "Working with Topologies".

The following example shows the topology graph for a selected probe:

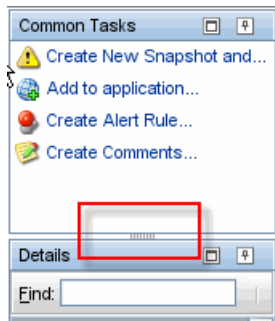
The screenshot displays the 'Diagnostics Topology' window. The main area shows a graph with a 'Consumers' icon on the left and a probe icon on the right, connected by an arrow. The probe is labeled 'L91_MSPetShop_1.1_NET_OVRNTT121_W2k3_probe8'. A legend below the graph shows a green circle next to a right-pointing arrow, labeled 'Consumer ...'. The right-hand side of the window contains several panels: 'Common Tasks' with options like 'Create New Snapshot and Add', 'View Profiler for L91_MSPetSh...', 'Add to application...', 'Create Alert Rule...', and 'Create Comments...'; 'Navigations' with 'View Layers (1)' and 'View Server Requests (2)'; and 'Details' for the selected probe 'L91_MSPetShop_1.1_NET_...', showing attributes like 'Default Name', 'Host IP Addr...', 'Host Name', 'Probe Type', 'app_server_...', 'Count', 'Exceptions', 'Throughput', and 'Timeouts'. At the bottom, a 'Probes' table lists various probes with their status, channel, name, VM Use, Latency, CPL, Throughput, Time, and Exceptions.

Status	Ch.	Probe	VM Use	Lat.	CPL (A...	Thro	Time	Exc	Info
●	✓	L91_MSPetShop_1.1...	43...	2.4...	46...	13...	0	0	
●		L81_WLS91_caller_...	85...	21...	0.0...	37...	0	0	
●		WAS61_Plants_ovri...							
●		P81_WLS81_MedRe...	10...	43...	11...	17...	0	0	
●		TOMCAT_hpswros0...	19...	3.4...	2.2...	20...	0	0	
●		L91_TestService2.W...	27...	3.4...	3.0...	18...	0	0	
●		WL81_SOA_Manage...	45...						
●		L91_JavaTrader2.W...	19...	8...	0.0...	60...	4	0	
●		L81_TestService2.W...	19...	1.2...	3.6...	65...	0	24	
●		L95_SimpleRemoting...	34...	10...	0.0...	19...	0	0	
●		L81_TestService2.W...	27...	16...	15...	13...	0	187	
●		L95_SimpleISClient	10	73	10	10	0	0	

Resizing and Docking Windows

You can customize the look of Diagnostics views by repositioning and rearranging (docking) the different display panes to suit your needs.

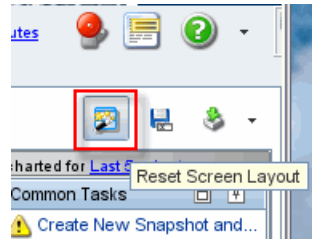
Most Diagnostics views are composed of several different panes. For example, the standard view for probes is made up of 4 panes: a graph pane, a table pane, a details pane and a common tasks and navigations pane. Many of the views have splitter controls that enable you to change the proportions of different areas of the view. They allow you to resize parts of the view so that areas of less interest are minimized to provide more area for the information that you want to see.



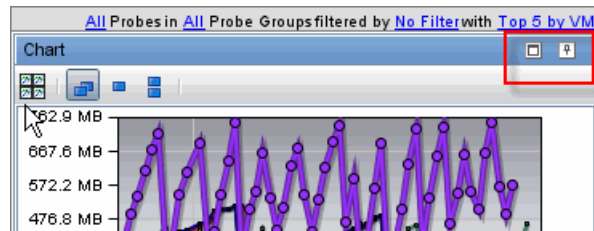
You can also select any of these panes and move and dock them independently. Select the pane by clicking at the top of the pane, the pane is then highlighted.

Click and drag the highlighted pane to move it wherever you choose in the window. You will see an outline of the pane displayed as you move it to different positions in the windows. Once you drop the pane in place, the other panes will resize to accommodate the new location and size of this window pane.



After rearranging the panes in the View you can select the **Reset Screen Layout** button to return the display to the default layout.



You can enlarge or hide one pane in a view by using either the Maximize/Restore icon or the Autohide icon.





Click the **Maximize/Restore** button to maximize the pane to fill the window or restore the pane to the previous size as illustrated in the following table.

Icon	Description
	Maximize. The pane is maximized to fill the window.
	Restore. The pane is restored to the previous size in the window.

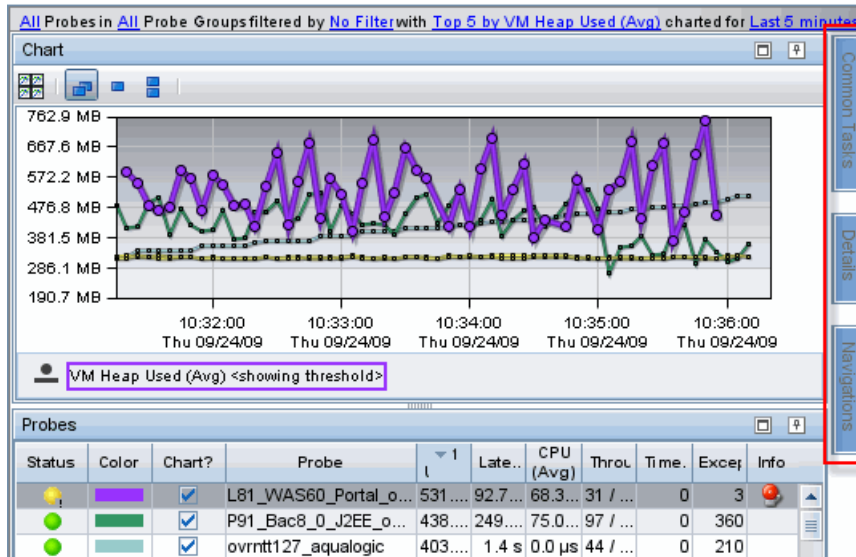
Maximize is a temporary condition - it is not considered a layout change so it is not persisted. To create a custom view with a maximized control - pin all controls that are not of interest.

You can use the Auto Hide feature to minimize panes that you don't want open. The window is minimized along the edges of the screen.

Click the **Auto Hide** button on the title bar of the window to enable or disable Auto Hide, as illustrated in the following table.

Icon	Description
	Auto Hide Enabled. Window is pinned (minimized) along the edges of the screen.
	Auto Hide Disabled. Window is displayed (open).

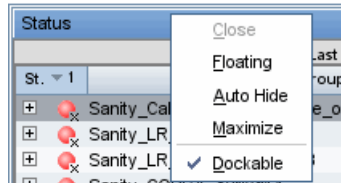
Minimized panes are shown in the example below:



You can even select auto-hide for the Views side-bar, if you need to have more room to display graphs and tables of data.

Mouse-over the hidden pane to view it and select the Auto Hide button to switch back to displaying the pane in the window.

You can also right-click in the title bar for a selected pane to Maximize/Restore or enable/disable Auto Hide.



In some views such as the Server Summary view you can set a pane to floating mode by right-clicking in the title bar to select Floating. The window pane undocks and floats above all the other panes. To dock the window pane back into position, double-click the title bar again.

About the Different Display Panes in a View

Many of the Diagnostics Views are divided into several panes as described below.

- **Chart (Graph).** The graph contains trend lines or stacked areas that represent the performance metrics.

For details, see "About Graphs (Chart)" on page 97.

- **Graph Entity Table.** The graph entity table lists the entities associated with a particular view. For example, the entities displayed in the Probes view are probes, and in the Load view they are layers.

For details about working with and customizing a graph entity table, see "About the Graph Entity Table" on page 108.

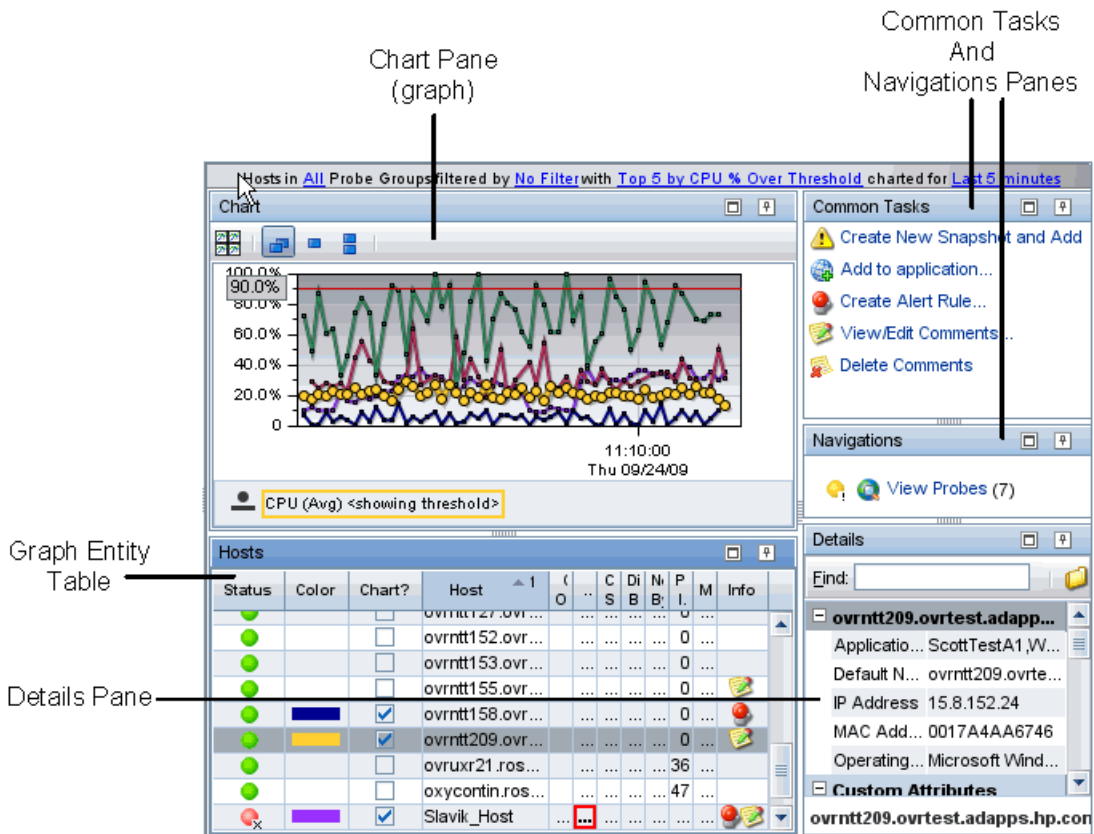
- **Common Tasks and Navigations.** The Common Tasks and Navigations panes show actions that are relevant to the entity selected in the graph entity table. They provide suggested actions you can take to further analyze a performance problem and diagnose the cause. You can also access these common tasks and navigation links by right-clicking on a row in the graph entity table and selecting from the menu displayed.

For details, see "About Common Tasks and Navigations" on page 117.

- **Details Pane.** The Details pane presents the metrics and custom attributes associated with the entity selected in the Graph Entity Table. From the Details pane, you control which metrics Diagnostics charts in the graph and which metrics are included in the active snapshot in Snapshot Analysis. You may also set thresholds that determine at what point you want to be warned that the value of a performance metric may be of concern.

For help on the Details Pane, see Chapter 4, "Working with Thresholds and Metrics in the Details Pane".

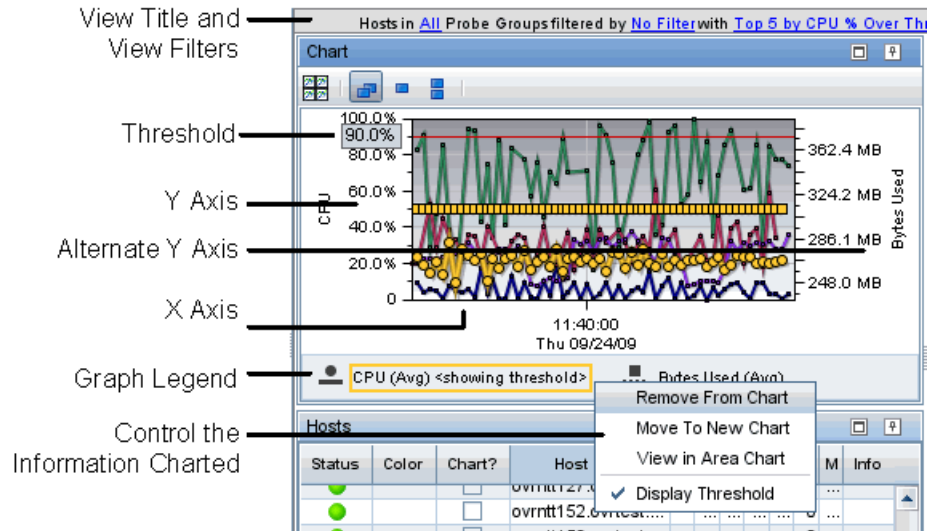
An example of these five panes displayed in a Diagnostics view is shown below:



About Graphs (Chart)

You use the view filters, graph entity table, and Details pane to control the information that appears in the graph.

The following image provides an overview of the common graph navigation and display controls:



- **View title and view filters.** The view title describes the view using the current settings for the view filters.

For details on how to work with view filters to change the information displayed in a view, see "About View Filters" on page 100.

- **Y-Axis.** The y-axis on the graph indicates the metric type, units, and scale for one or more of the related metrics that are charted on the graph. In the image above, the y-axis indicates that one or more of the metrics charted on the graph represents CPU utilization, and that CPU utilization is charted in percentage points.

- **Alternate Y-Axis.** An alternate y-axis is displayed when one or more metrics on the graph is measured with different units or scale than those plotted on the y-axis. The alternate y-axis works just like the y-axis. In the image above, the alternate y-axis indicates that one or more of the metrics charted on the graph represents Context Switches per second and that each tick mark on the axis represents 1000 context switches.
-

Note: Make sure to use the correct y-axis to determine the value for a particular graphed data point when more than one y-axis is displayed in the graph. In the image above, the two charted metrics each use a different y-axis.

- **X-Axis.** The x-axis on the graph indicates the time range and scale used to plot the metrics on the graph. In the image above, the x-axis indicates that **Last 5 Minutes** was selected from the **Time Range** filter, and that each interval across the axis represents a one minute time interval for a specific date and time.
-

Note: If you are using Diagnostics with LoadRunner or Performance Center as part of a load test, the x-axis represents the time that has elapsed since the beginning of the load test.

- **Graph Legend.** The graph legend displays the line patterns used to represent metrics on the graph. All of the lines on the graph that were charted with a particular pattern are associated with the metric indicated in the legend.
- **Menu Selections to Control the Information Charted.** By right-clicking on a graph legend item, you can access a menu with the following control options:

- ▶ **Remove From Chart.** Selecting this option removes all trend lines or stacked areas for the indicated metric from the graph. This is an alternate way of removing the metric from the graph instead of using the controls in the Details pane.

For details on selecting metrics to chart from the Details pane, see "Chart a Metric in the Graph" on page 156.

- ▶ **Move to New Chart.** When more than one metric has been charted in the graph, the menu contains an option to move the metric to a new graph. This can allow you to make a graph less cluttered so that you can see the information more clearly. Selecting this option causes Diagnostics to open a new graph for the metric. All trend lines, stacked areas, and thresholds for the indicated metric are moved from the original graph to the new graph.
- ▶ **Move to Chart with...** When more than one graph appears in the graph area, the menu contains an option to move a metric that appears on one graph to another graph. Selecting this option causes Diagnostics to move all trend lines, stacked areas, or threshold indicator for the selected metric from the original graph to the graph indicated in the selected menu option.

Note: You will not be able to move a charted metric to a graph that is already using the y-axis and the alternate y-axis when the metric to be moved is not compatible with the y-axis.

- ▶ **View in Area Chart.** Allows you to switch to a line view of the metric. All metrics that chart by default in a line graph can be viewed in an area chart. If the metric is displayed as an area graph, the menu item will be **View in Line Chart** to allow you to switch to a line view of the metric.
- ▶ **Display Threshold.** This option is only active when the metric can have a metric threshold assigned.

For details on setting thresholds, see Chapter 4, "Working with Thresholds and Metrics in the Details Pane".

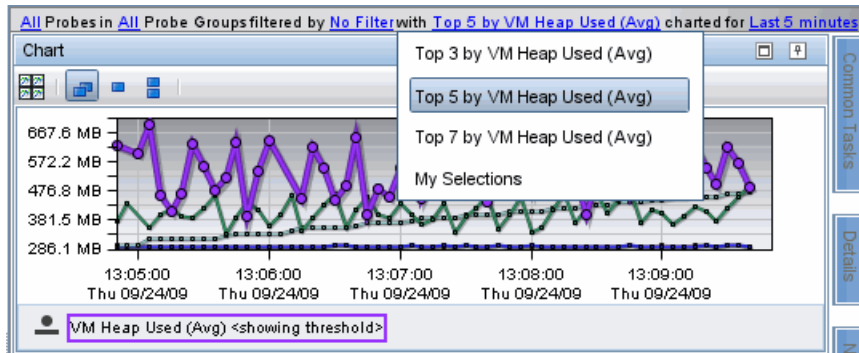
Note the following:

- ▶ Selecting this option when the threshold has not already been charted in the graph causes a line to be charted in the graph for the threshold that you have established for the metric. If the threshold for another metric had been previously charted, the threshold for that metric is removed from the graph because only one threshold can be charted on a graph.
- ▶ Selecting this option when the threshold is already charted causes the threshold line to be removed from the graph.

About View Filters

The view filters, a set of drop-down menus that are included in the view title of many of the Diagnostics views, are used to control the information displayed in the graph, graph entity table, and Details pane. The filters that appear in each view title vary according to the displayed view. To set a filter, click a link and select a value from the drop-down menu.

Below is an example of the view title and the view filters from the Probes view. The title includes the Probe Type **All**, Probe Group **All**, Custom Filter **No Filter**, Graph **Top 5 by VM Heap Used (Avg)**, and Time Range **Last 5 minutes** menus. The **Graph** filter menu is open displaying the available options.



The time range (locked/unlocked) and probe group filter settings apply as you drill to all other views. Filter settings at lower level views, after drill down, do not apply as you navigate back up to higher level views.

Multiple Graph Layout Options

In the Chart pane you can customize the layout of the graphs. This is useful when you have multiple graphs displayed in the Chart pane.

There are two default graph layouts to choose from:



Select to layout both vertically and horizontally, in two columns.



Select to layout horizontally, in one column.

Click the icon to toggle between the two default layouts. Graphs are added to the view based on the selected default layout.

There are also several options you can choose to determine how trend lines are grouped.



Charts trends with the least number of graphs. Adds each additional metric to an existing graph that is trending like metrics, as determined by units and available axis. Up to four metrics are assigned to a graph.



Charts trends with one metric per graph. Trend lines for the top entities (or selected entities) for a charted metric are shown on a single graph.



Charts trends with one entity per graph. The trend lines for each of the top entities (or selected entities) for the charted metrics are shown on separate graphs. The metrics are added to the chart as in the first case - least number of graphs.

Entities and Metrics Charted in the Graph

You can determine which trend lines or stacked areas in the graph are associated with a particular entity in the graph entity table by clicking the row in the table to select the entity. Diagnostics highlights the selected entity and each of the relevant charted metrics in the graph.

You choose entities or metrics to view in the graph in one of the following ways:

- ▶ Using the **Chart** column in the graph entity table to select which entities' metrics to chart in the graph (see below).
- ▶ Using the view filters in the graph to select, for example, the top five probes by latency to chart in the graph.
For details, see "How to Work with View Filters" on page 131.
- ▶ Using the **Details** pane chart metric icon to add a selected metric to the graph.
For details, see "Chart a Metric in the Graph" on page 156.

You can determine which metrics are charted on the graph by looking at the graph legend.

This section also includes:

- ▶ "Using the Chart Column in the Table" on page 102
- ▶ "Viewing Multiple Metrics on a Graph" on page 103

Using the Chart Column in the Table

The **Chart** column on the graph entity table lets you control which entities' metrics Diagnostics is to chart in the graph.

- ▶ To chart the metrics for an entity in the graph, select the **Chart** check box for that entity.
- ▶ To remove the charted metrics for an entity from the graph, clear the **Chart** check box for that entity.

- To remove the charted metrics for all of the entities selected in the **Chart** column, right-click the **Chart** column header and select **Remove All Series from Graph**. You may then repopulate the graph by clicking the **Chart** check box for the entities whose metrics you would like to have charted. You can also use the **Graph** filter in the **View Title** to let Diagnostics pick significant sets of entities to chart.

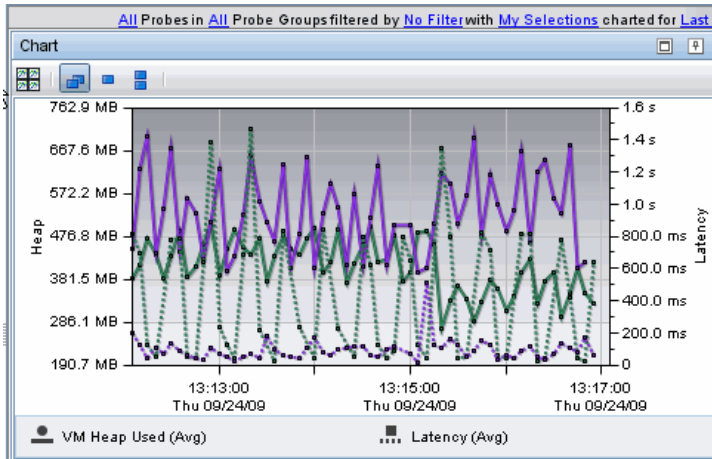
When you click the **Chart** check box for any entities in the graph entity table, Diagnostics changes the **Graph** filter in the **View Title** to **My Selections** to indicate that you have manually selected the entities. For more information about the **Graph** filter, see "How to Work with View Filters" on page 131.

Viewing Multiple Metrics on a Graph

When more than one metric is charted in the graph, Diagnostics helps you identify each metric by charting each with a different pattern. The patterns that are used for each metric are recorded in the **Graph Legend**.

Each entity is color coded, and can be identified in the graph by a line charted in the same color that is indicated for the entity in the **Color** column of the graph entity table. Each metric selected from the **Details** pane is charted for each entity selected from the graph entity table, so long as a valid metric value exists for the entity.

In the following image, both the Heap and Latency metrics are charted for the two probe entities selected from the graph entity table. The pattern used to chart metrics for each entity are shown in the graph legend.



The Data Behind a Charted Metric on the Graph

From the graph, there are several ways to determine the entities and metrics that are behind the trend lines and stacked areas charted in the graphs.

This section also includes:

- ▶ "Identify the Entity in the Graph Entity Table for a Charted Metric" on page 104
- ▶ "Identify the Metric Details for a Charted Point on the Graph" on page 105

Identify the Entity in the Graph Entity Table for a Charted Metric

From the graph, you can associate a trend line or stacked area with the entity in the graph entity table that it represents. You do this in one of the following ways:

- ▶ Hold the mouse pointer over a data point in the graph. A tooltip appears displaying the details for the entity and metric represented by that point in the trend line. This is discussed in more detail in the section below.
- ▶ Click anywhere on a trend line or stacked area in the graph. The trend line or stacked area is highlighted in the graph, and the entity is highlighted in the graph entity table. At the same time, the Details pane is updated to display all of the metrics for the selected entity.

When you select a trend line in the graph, all of the trend lines for the metrics charted for the entity are also highlighted.

When you select a stacked area in the graph, it is displayed as shaded rather than solid.

Identify the Metric Details for a Charted Point on the Graph

From the graph, you can associate a trend line or stacked area with the entity in the graph entity table that it represents. You do this in one of the following ways:

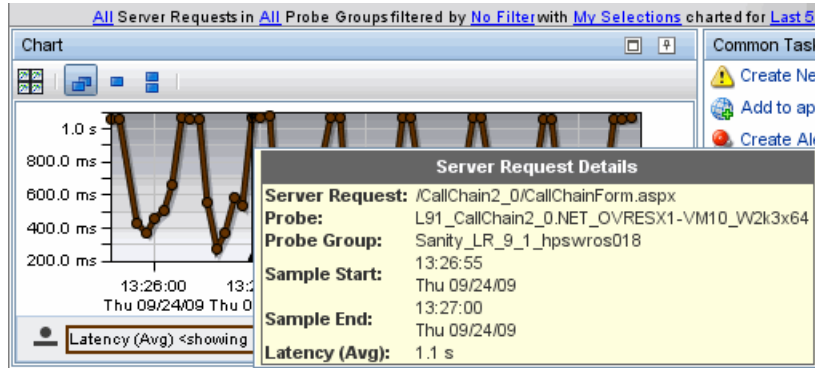
- ▶ Hold the mouse pointer over a data point in the graph. A tooltip appears displaying the details for the entity and metric represented by that point in the trend line.
- ▶ Click anywhere on a trend line or stacked area in the graph. The trend line or stacked area is highlighted in the graph, and the entity is highlighted in the graph entity table. At the same time, the Details pane is updated to display all of the metrics for the selected entity.

When you select a trend line in the graph, all of the trend lines for the metrics charted for the entity are also highlighted.

When you select a stacked area in the graph, it is displayed as shaded rather than solid.

The data points that were used to chart the trend lines on the graph are highlighted with a point marker on the trend line. On the stacked area charts, you can recognize these data points when the slope of the boundary of the stacked area changes.

When you hold the mouse pointer over a data point, a tooltip appears containing details for the underlying data point, as shown in the following example:



The information in the tooltip varies depending on the detail layout view that Diagnostics has displayed. The tooltips can include:

- ▶ **Title.** The title bar identifies the type of entity to which the data point belongs. In the example above, the title bar is **Server Request Details**, which indicates that the tooltip contains the details for a metric displayed in the Server Requests view.
- ▶ **Entity Type and Name.** The entity name identifies the type and the name of the displayed entity. In the example above, the type of entity is **Server Request**, and the name of the entity is **/CallChain2_0/CallChainForm.aspx**.
- ▶ **Aggregation Period Start Time.** The **Sample Start** entry contains the starting time of the aggregation of the metrics for the data point.
- ▶ **Aggregation Period End Time.** The **Sample End** entry contains the time that metrics for the data point were last gathered.
- ▶ **Metric Name.** The metric name identifies the name and the value for the charted metric. In the example above, the metric is **Latency (Avg)**.

Zoom Features in the Graph

Diagnostics enables you to control the appearance of the graphs by zooming in and out.

Diagnostics provides two different types of zooming: magnification zooming and resolution zooming. It automatically determines which type of zooming to use based on the selected time range and zoom range.

Magnification Zooming

Magnification zooming provides a magnified view of the data points. Diagnostics does not retrieve any additional information when portraying a magnified view of the performance metrics. There are two cases when Diagnostics uses a magnification zoom:

- ▶ when the graph has been plotted using data that had been aggregated into the highest resolution data points
- ▶ when you select a zoom range that spans a time period that requires too much data to be retrieved for an efficient resolution zoom

Resolution Zooming

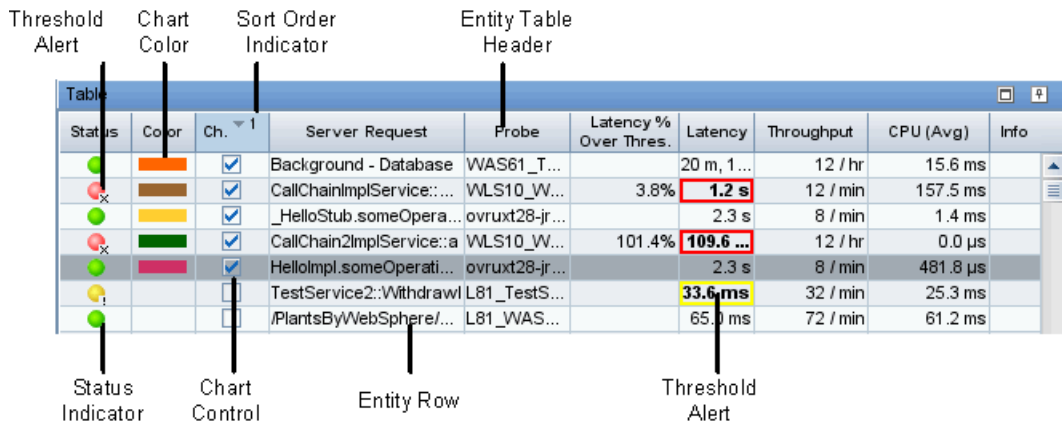
Resolution zooming provides a higher resolution view of the performance metrics. Diagnostics retrieves additional data to chart metrics at a higher resolution.

When the selected time range is larger than **Last 5 Minutes**, Diagnostics retrieves data that has been aggregated into lower resolution data points. That is, the points charted on the graph represent larger time periods. For example, when you select a time range of **Last Hour**, the data is aggregated into one-minute aggregations. These points are initially charted so that all of the data for the last hour is visible in the graph. If the data points are too close together to analyze the performance for a particular area on the graph, you can zoom into that area. In this situation, where the graph displays data that has been aggregated at a lower resolution, Diagnostics retrieves data that has been aggregated at a higher resolution to zoom into the metrics charted.

Note: If the selected zoom range is large and requires Diagnostics to retrieve more high-resolution data than can be done efficiently, the zoom is performed using the magnification zoom instead.

About the Graph Entity Table

From the graph entity table you can select the entities that are to be charted on the graph. The graph entity table is highly customizable allowing you to control the columns that appear in the table and the sort order for the rows in the table. The following image provides an overview of the common navigation and display controls that are included.



Status	Color	Ch.	Server Request	Probe	Latency % Over Thres.	Latency	Throughput	CPU (Avg)	Info
		<input checked="" type="checkbox"/>	Background - Database	WAS61_T...		20 m, 1...	12 / hr	15.6 ms	
		<input checked="" type="checkbox"/>	CallChainImplService:...	WLS10_W...	3.8%	1.2 s	12 / min	157.5 ms	
		<input checked="" type="checkbox"/>	_HelloStub.someOpera...	ovruxt28-jr...		2.3 s	8 / min	1.4 ms	
		<input checked="" type="checkbox"/>	CallChain2ImplService:a	WLS10_W...	101.4%	109.6 ...	12 / hr	0.0 μs	
		<input checked="" type="checkbox"/>	HelloImpl.someOperati...	ovruxt28-jr...		2.3 s	8 / min	481.8 μs	
		<input type="checkbox"/>	TestService2::Withdrawal	L81_TestS...		33.6 ms	32 / min	25.3 ms	
		<input type="checkbox"/>	/PlantsByWebSphere/...	L81_WAS...		65.0 ms	72 / min	61.2 ms	

- **Entity Table Header.** From the graph entity table header, you control which columns are displayed in the table and the order in which they appear. You also control the sort order of the entities displayed.

For details about customizing the columns, see "How to Customize Table Columns" on page 141.

For details about sorting the table, see "How to Customize Table Sort Order" on page 144.

- **Entity Rows.** The entities in the table correspond to those presented in the graph. The entity shows you the details of the performance of your application and enables you to drill to more performance details.

The entities that are listed in the table and the order in which they appear changes as Diagnostics updates the information in the view.

Several of the columns that appear in the graph entity table are the same for most of the predefined detail layouts. Other columns may be specific to a particular view. The columns common to most of the predefined detail layout views are:



- **Status.** This indicator shows how the listed entity or any of its child entities is performing relative to the thresholds that you set for their metrics. See "About Status Indicators" on page 110.
- **Color.** This column contains a colored block. When the **Chart** check box for an entity in the graph entity table is selected, trend lines or stacked areas for the entity are included in the graph. The color of the block in the Color column is the color that Diagnostics uses to chart the metrics for the entity. When Diagnostics has charted more than one metric for an entity, the trend lines or stacked areas for the entity's metrics appear in the graph with the same color.
- **Chart.** This column contains a check box to select the entities whose metrics should appear in the graph. When the entity is selected, Diagnostics charts a trend line or stacked area for each of the entity's metrics selected in the Details pane. For more information about charting the metrics for an entity in the graph entity table, see "Using the Chart Column in the Table" on page 102.
- **Metric Columns.** The metric columns that Diagnostics displays in the graph entity table vary depending on the displayed view and table customizations. If the value of a metric in a metric column has exceeded the threshold that you defined for the metric, the cell in the metric column that contains the offending metric is displayed with a threshold violation indicator. The threshold violation indicator appears as a colored outline, either yellow or red. This threshold violation indicator corresponds to threshold violation indicator for the metric in the Details pane.

For details about metrics thresholds, see Chapter 4, "Working with Thresholds and Metrics in the Details Pane".

- **Info.** This column indicates that the entity has alert rules or comments. One or more of the following icons can appear in this column:



- **Active Alert Rule.** Indicates that an alert notification rule was created for the selected entity and the rule is active.



- **Disabled Alert Rule.** Indicates that an alert notification rule was created for the selected entity but the rule is currently disabled. (You can disable an alert rule in the Alert Rule dialog box by de-selecting both checkboxes for Alert Triggers.)



- **Custom Comments.** Indicates that comments were created for the selected entity.

For details about alert notifications, see Chapter 5, "Working with Alerts".

For details on entering comments, see "Manage Comments for an Entity" on page 123.

About Status Indicators



- **Status.** This indicator shows how the listed entity or any of its child entities is performing relative to the thresholds that you set for their metrics.

For example, a probe can have a critical status even when none of the probe's own metrics have exceeded their thresholds. The probe's status will become critical when any of the server requests for that probe become critical.

The color of the indicator stands for the severity of the threshold violation. If you hold the mouse pointer over the status indicator in a row of the table, Diagnostics displays a tooltip with a description of the current status:

Status Indicator Color	Description
Green	Good – The entity is performing within defined thresholds.
Yellow	Warning – The component is occasionally but not consistently exceeding defined thresholds. If the metric that is exceeding the defined threshold is displayed in the entity table, the cell for the metric is also outlined in yellow.
Red	Critical – The component is consistently exceeding defined thresholds. If the metric that is exceeding the defined threshold is displayed in the entity table, the cell for the metric is also outlined in red.
Grey	No status information available. Either no recent data was received for the metric or no threshold was set.

If the metrics that are exceeding the threshold are displayed in the row of the graph entity table, they are highlighted with a threshold violation indicator.

The following describes, in general, how a status of RED or YELLOW is determined:

- ▶ Diagnostics compares the average value for the metric in the last 5 minutes to the threshold. If the threshold is crossed, the status is RED.
- ▶ If not, Diagnostics counts the total number of points (5-second averages from the probe) that violate the threshold. If the count is greater than three but the average is still under the threshold (current status is not RED), the status is YELLOW; otherwise the status stays at GREEN.

- ▶ If none of the above, the status is GREEN.

Status indicators can progress in severity from yellow and then to red as the metric's performance deteriorates. However, the indicators do not regress in the same way. Once a red indicator has been issued, it continues to be displayed as red until the metric value returns to normal levels.

For information about setting thresholds, see "Threshold Definitions" on page 158.

Additional Information about the Entities in the Table

The columns in the graph entity table contain a subset of the attributes and performance metrics that Diagnostics has gathered for the listed entities. However, the information in the table is just a high-level summary of the information available to help you analyze the performance of your applications. Diagnostics provides many ways for you to dig deeper into the metrics associated with the entities listed on the graph entity table.

This section includes:

- ▶ "View Tooltips for the Entity" on page 112
- ▶ "View Metric Details for the Entity" on page 113
- ▶ "Drill-down Navigations for the Entity" on page 114

View Tooltips for the Entity

Many of the cells in the rows of the graph entity table display tooltips when you hold the mouse pointer over the cell.

- The tooltip for the entity column contains configuration information for the listed entities. The content of the tooltip varies depending on the entities that are listed in the table. The following example shows the tooltip for the entities in the Probes view:

Status	Color	Chart?	Probe	Lat	CPI (A..	Thrx	Tim	Exc	Info
		<input checked="" type="checkbox"/>	L81_WAS60_Portal...	56...	10...	69...	30...	0	1
		<input checked="" type="checkbox"/>	P91_Bac8_0_J2EE_...	42...	11...	38...	20...	0	390
		<input checked="" type="checkbox"/>	ovrntt127_						
		<input checked="" type="checkbox"/>	L81_WLS9						
		<input checked="" type="checkbox"/>	WLS10_Me						
		<input type="checkbox"/>	WLS10_VV						
		<input type="checkbox"/>	L91_WAS_						

Probe Details	
Probe:	ovrntt127_aqualogic
Probe Group:	IRD_05_DEMO
Host:	ovrntt127.ovrtest.adapps.hp.com
Probe Type:	Java

- The tooltip for the columns that contain metric values display the same value that is displayed in the selected cell. This is useful if you have resized the column and want to know the value of the listed metric without changing the width of the column again.

View Metric Details for the Entity

When you select an entity in the graph entity table, Diagnostics displays the metrics for the selected entity in the Details pane.

Details	
Find:	<input type="text"/>
<div style="border: 1px solid gray; padding: 2px;"> L81_TestService2.WebService.NET_OVRN... </div>	
Default Name	L81_TestService2.Web...
Host IP Address	15.8.152.24
Host Name	ovrntt209.ovrtest.adap...
Probe Type	.NET
app_server_topology	iis:data_name##Micros...
Count	688
Exceptions	192
Throughput	138 / min
Timeouts	0
Custom Attributes	
Latency	
Avg	25.1 ms

For more information about the Details pane, see Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

Drill-down Navigations for the Entity

Most of the Diagnostics views provide a way for you to drill to other Diagnostics views to see more detailed performance metrics for an entity selected in the Details pane. You can select a drill to link in the Navigations pane or right-click the entity in the table and select a drill to option from the pop-up menu. You can also double-click an entity in the table and drill to a relevant view if a navigation is available.

For some entities you can open the Diagnostics Profiler to get more detailed information and analyze memory problems.

For details on relevant navigations for an entity, see "Common Navigations" on page 125.

Table Header Controls

The graph entity table headers provide controls that enable you to specify which columns are to appear in the table and the order in which the columns are to be displayed, as well as specifying which columns are to determine the sort order for the rows in the table.

The default columns that appear in the status views and the graph entity table of other views help you identify the entities that are included in the view, understand the status of each entity's performance, and see some of the performance metrics that have been gathered for each entity.

Note: Graph entity tables for server requests and similar data like web services only show the top 100 records by metric value.. The 100 limit is per mediator. If you change the metric column you are sorting by, you can get a different set of 100 entities. This obviously applies only where there are more than 100 things in the dataset, which can happen in large deployments. With a large number of server requests you can use the filter to display all server requests for a probe group or type or view server requests by probe. Also see "UI Properties" on page 116 for some UI configuration settings for large environments.

For details on customizing columns to arrange the data in the table as you like, see "How to Customize Table Columns" on page 141.

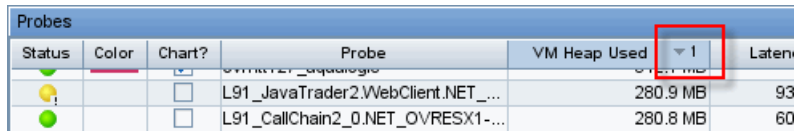
Note: If you customize one of the standard views, and would like to save the customizations for future use, save the view as a custom view. If you do not save the changes as a custom view, they are lost when you close the Diagnostics UI.

For details on saving custom views, see "How to Save a Custom View" on page 227.

Most of the Diagnostics tables are sorted in ascending order on the first metric column in the table, and then ascending order on the entity name column. You can customize the sort order using the controls built into the column headers in the table.

For details on custom sort order, see "How to Customize Table Sort Order" on page 144.

For example, in the Probes view, the default primary sort is based on the values in the **VM Heap Used** column. The following example shows the column headers for the Probes view:



Status	Color	Chart?	Probe	VM Heap Used	Latency
		<input type="checkbox"/>	L91_JavaTrader2.WebClient.NET_...	280.9 MB	93
		<input type="checkbox"/>	L91_CallChain2_0.NET_OVRESX1-...	280.8 MB	60

In the example above, the VM Heap Used column has the superscript down arrow followed by the number **1** indicating that the column is sorted in descending order, and the number **1** indicates that this column is the primary sort for the table.

UI Properties

You can set global UI properties in the **ui.properties** file on the server. The user interface reads this property file once on startup.

Configuring for a Large Number of Concurrent User Interface Sessions

When you have a large number of concurrent user interface sessions you may want to configure the frequency of the queries that the user interface is executing so as to not inundate the server.

In the **ui.properties** file on the commanding server you can set properties to throttle the update frequency of ui queries to the server. Set **ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.builder.handles.<query type>** to a multiplier greater than one.

The Enterprise user interface data is updated based on the time control (last 5 minutes, 20 minutes, 1 hour, and so on) and query type. Adjust the multiplier upward as the number of users increases.

For example trend data that populates charts is updated every 5 seconds when the time control is set at last 5 minutes. When the property `ui.query.update.frequency.multiplier.com.mercury.diagnostics.common.data.query.builder.handles.TrendDataHandle=2` is enabled then the chart will update every 10 seconds ($5\text{ s} * 2$ multiplier).

You can get a list of active users seen by the Diagnostics server in the last 60 seconds and the Queries/sec load the user generates. From the main Diagnostics UI select **Configure Diagnostics > query > Active Users** link.



IP-Address	User Name	Last	First	Queries/sec
15.8.157.151	admin	Mon May 31 09:13:34 PDT 2010	Mon May 31 04:37:35 PDT 2010	4.19
127.0.0.1	mercury	Mon May 31 09:13:32 PDT 2010	Sat May 29 07:19:00 PDT 2010	0.04
16.93.25.245	admin	Mon May 31 09:13:35 PDT 2010	Mon May 31 05:14:16 PDT 2010	1.16
16.93.5.10	admin	Mon May 31 09:13:35 PDT 2010	Sat May 29 07:18:46 PDT 2010	1.19
16.17.242.157	admin	Mon May 31 09:13:29 PDT 2010	Mon May 31 02:22:16 PDT 2010	0.76

HP Diagnostics Server "server-OVRNTT150", version 9.00.72.1046

Configuring for a Large Number of Probes Reporting to a Mediator

When you have an environment with thousands of probes reporting to a mediator you can limit the number of entities returned on user interface queries from a mediator to the commanding server.

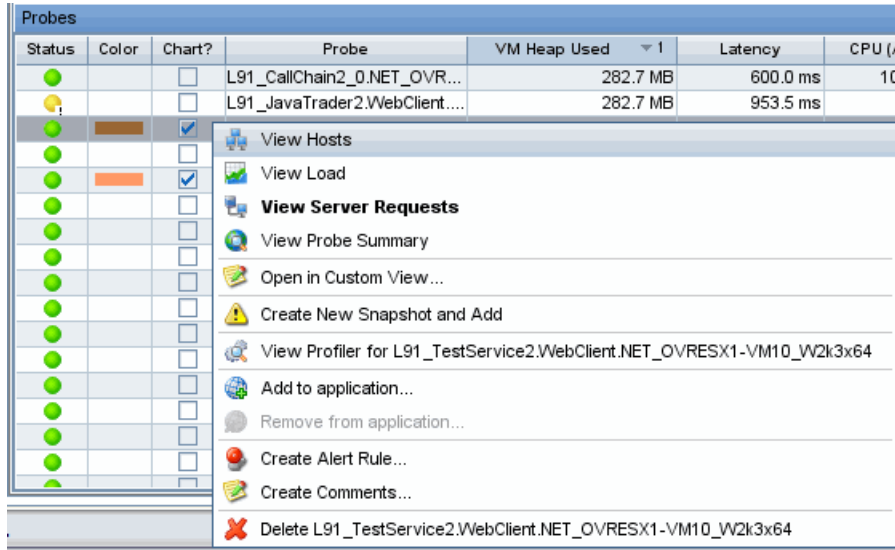
In the `ui.properties` file on the server you can set the `ui.topn` property so the mediator returns the topN by metric. Use this property when viewing probe views of the Entire Enterprise. Only Probe views can be limited by enabling this value.

About Common Tasks and Navigations

Diagnostics provides access to the common tasks and navigations that are available for an entity in the table in two ways:

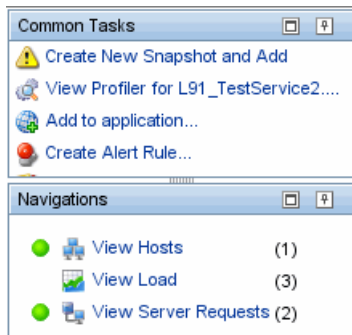
- ▶ Right-clicking on the row in the table to display the entity pop-up menu.
- ▶ Using the Common Tasks and Navigation pane above the Details pane on the right side of the Diagnostics views.

The following image shows an example of the entity pop-up menu that is displayed for a probe entity in the Probes view.



The Common Tasks pane contains tasks you may want to perform based on the currently selected entity. The Navigations pane contains links to navigations you may want to use based on the currently selected entity.

The navigation links listed also show the status of the related entities and their count, where this information is available.



When you select a row in the graph entity table, Diagnostics updates the right-click menu items and the Common Tasks and Navigations panes so that they contain only **relevant** tasks and navigations that are appropriate for the selected entity.

See the following related topics:

- "Common Tasks" on page 119
- "Common Navigations" on page 125

Common Tasks

The following common tasks are described in the sections that follow:

- "Create a New Snapshot and Add the Entity" on page 119
- "Open in Custom View" on page 120
- "Add and Remove an Entity from an Application" on page 121
- "Manage Alert Rules for an Entity" on page 123
- "Manage Comments for an Entity" on page 123
- "Delete an Entity" on page 124
- "Purge Data for an Entity" on page 124

Create a New Snapshot and Add the Entity

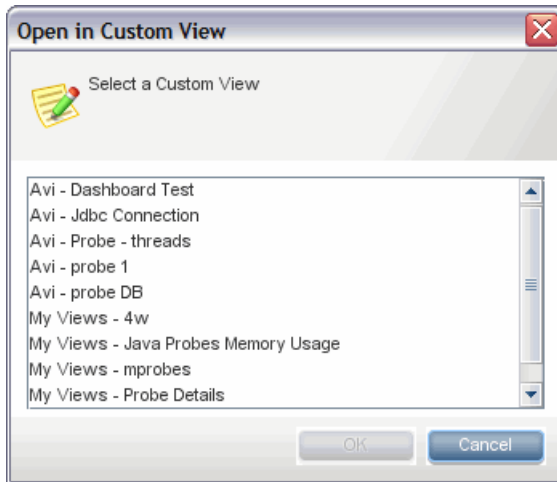
Selecting **Create New Snapshot and Add** from the entity pop-up menu or from the Common Tasks pane causes Diagnostics to create a new snapshot including the selected entity and all of the metrics that were charted for the entity. Diagnostics switches from Monitoring mode to Snapshot Analysis mode and displays the Analyze Snapshots tab with the newly created snapshot.

For details on snapshot analysis, see Chapter 6, "Performing Snapshot Analysis".

Open in Custom View

Selecting **Open in Custom View** from the entity pop-up menu allows you to open a custom view in the context of a selected entity.

From a view such as the Probes view, select an entity (in this case a probe) and right-click to select Open in Custom View in the pop-up menu.



Custom views that apply to the type of entity selected, in this case probes, are listed in the dialog box for you to select. The custom view is then displayed in the context of the entity selected. This allows you to use custom views with different instances of an entity.

For probes, two custom views are provided out of the box: Probe Details and Java Probes Memory Usage.

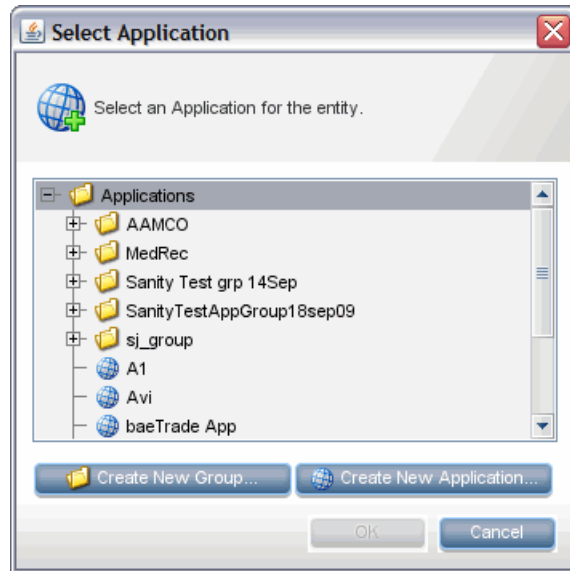
The list of custom views displayed in the Open In Custom View dialog box is very broad so even though the custom views generally relate to the type of entity selected, a given custom view may not apply to the specific entity selected because of other characteristics in the custom view.

Avoid opening in a custom view if it contains a feature that is graphing based on the My Selections filter, unless the selections make sense for the entity selected. So, for example, it would be fine to open the custom view on a probe that contains a feature with server requests on the selected probe.

For details on custom views, see Chapter 7, "Working with Custom Views" for details on custom views.

Add and Remove an Entity from an Application

Selecting **Add to Application** from the entity pop-up menu or from the Common Tasks pane displays the Select Application dialog box. You select an application from this list and the entity will be added to that application.



An entity can be added to any number of applications. See Chapter 2, "Working with Applications" for how to work with applications.

You can select multiple entities in the graph entity table to add or remove from an application. See "How to Select Multiple Rows in the Table" on page 147 for how to select multiple rows. When selecting multiple entities you should select groups of entities that you want to add to or remove from the same application.

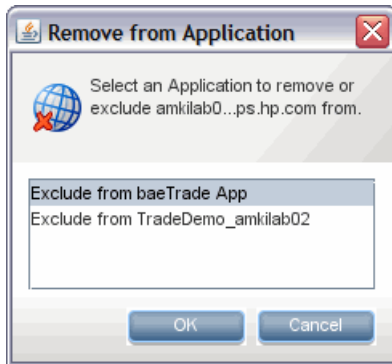
If the user you are logged in as has the appropriate permissions, you can also **Create a New Application Group** and **Create a New Application** from this dialog box. When you create a new group or application, that group or application will automatically be selected.

Note: If you create a new application (or multiple applications) from this dialog box, but then cancel out of the dialog box, the created applications will still exist.

Selecting **Remove from Application** from the entity right-click menu or from the common Tasks menu displays the Remove from Application dialog box. You can select an application to remove or exclude the entity from. The user you are logged in as, must have the appropriate permissions for you to be able to remove or exclude entities from an application.

The easiest way to remove entities you do not want is from the Topology view. Select the entity in the Topology view and right-click to select **Remove from Application**.

You cannot remove or exclude an entity if it is a parent of an entity that was manually added. For example, if you add the server request **/login.do**, you will not be able to remove or exclude the probe where this server request is running since this is the parent of the server request entity. This limitation exists to ensure that you will always be able to navigate to anything you have added to your application.



Note: In the Remove from Application dialog box you may see Remove from or Exclude from actions. This distinction is based on internal relationships between entities but both have a similar function so you can ignore the distinction.

You can remove entities from applications in either the Entire Enterprise or a selected application context.

For details on how to define applications, view data in the context of an application, add or remove entities from an application, see Chapter 2, "Working with Applications" for how to work with applications.

Manage Alert Rules for an Entity

You can create and edit alert rules for an entity listed in the graph entity table by right-clicking on the entity and selecting one of the alert options from the pop-up menu. The same menu options are also available on the Common Tasks pane. The options displayed in the menu vary depending on whether alert rules exist or not.

You can create an alert rule based on a metric collected for an entity by selecting the Alert Rule indicator icon next to the metric in the Details pane.

You can select multiple entities in the graph entity table and then select the menu item to create, replace or delete alert rules. See "How to Select Multiple Rows in the Table" on page 147 for how to select multiple rows. Typically, when selecting multiple entities you should select groups of entities that you want to have the same alert rules.

For details on alerts, see Chapter 5, "Working with Alerts".

Manage Comments for an Entity

You can create and edit comments for an entity listed in the graph entity table by right-clicking on the entity and selecting one of the comment options from the pop-up menu. The options displayed in the menu vary depending on whether comments exist or not. The same menu options are also available on the Common Tasks pane.

You can select multiple entities in the graph entity table and then select the menu item to create, replace or delete comments. See "How to Select Multiple Rows in the Table" on page 147 for how to select multiple rows. Typically, when selecting multiple entities you should select groups of entities that you want to have the same comment.

Delete an Entity

There are times when you may want to remove a probe or get rid of old metrics sooner than the six month automatic purge (see "Purge Data for an Entity" on page 124). The **Delete <entity>** in the entity pop-up menu in the graph entity table provides a way for you to delete a probe or old metrics from Diagnostics. The same option is available on the Common Tasks pane.

When you delete a probe that is actively sending data to Diagnostics, the old data for the probe is deleted and the probe disappears from the graph entity table. However, the next time that the probe sends its metrics, the probe will reappear in the graph entity table with only the new metrics that were just received.

When you delete a probe that is no longer sending data to Diagnostics, the old data for the probe is deleted and the probe disappears from the graph entity table. You can no longer reference the probe in Diagnostics.

Purge Data for an Entity

Diagnostics automatically purges an entity from the Diagnostics data after six months have passed since data was last received from the probe. Once an entity (such as a probe) is purged, access to its trends and summaries is no longer possible.

This is also the case when a probe is renamed. Data for the probe under the old name is only available for 6 months after the rename. Even if you don't change the probe but the monitored application is updated and server requests change then the old server requests data will not be available after 6 months.

If you have redeployed an application or made other changes to its configuration that are likely to significantly alter the applications performance, you may want to purge the old data that the probe monitoring the application collected so that the old performance metrics do not distort the current application's performance characteristics.

The purging interval can be changed in **server.properties**, see the *HP Diagnostics Installation and Configuration Guide*.

Common Navigations

Some commonly used Navigations are described in the sections that follow:

- ▶ "Navigate to Other Diagnostics Views" on page 125
- ▶ "View the Diagnostics Profiler for a Probe Entity" on page 126

Navigate to Other Diagnostics Views

Many of the Diagnostics views allow you to navigate from the entities listed in the graph entity table to other views so that you can analyze their performance in greater detail and diagnose performance problems. Diagnostics updates the Navigations panes so that it lists only **relevant** navigations that are appropriate for the selected entity.

As you navigate from one view to the next, the breadcrumb at the top of the view keeps track of the path that you have taken so that you can navigate backwards. If you navigate to an entity screen that is already in the breadcrumb, then the navigation will move up the breadcrumb to the appropriate level.

Note: Not all views have drill down navigation options.

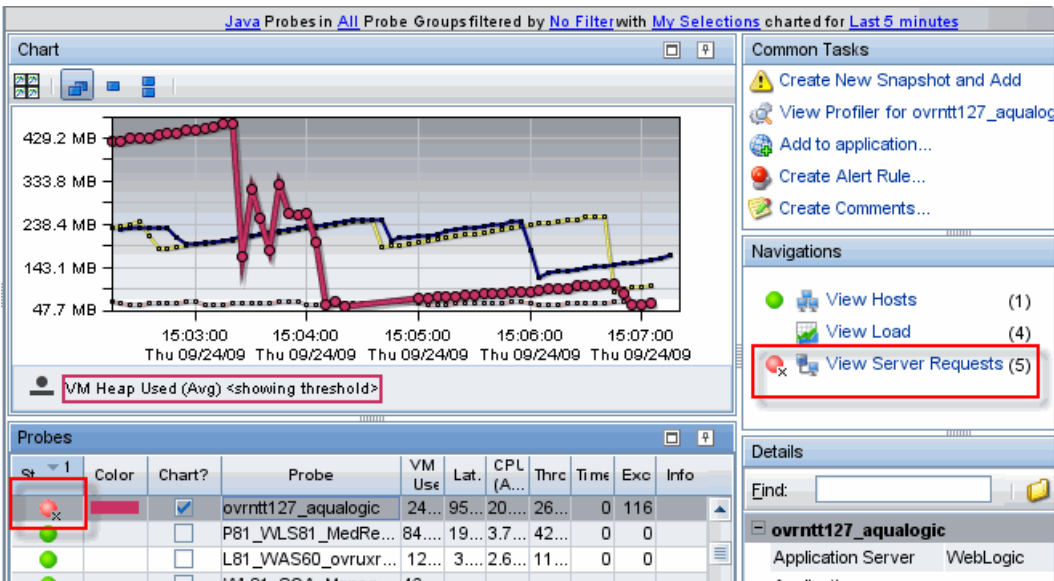
There are three ways to navigate from an entity in the graph entity table to other views:

- ▶ Right-click a row in the table. Diagnostics displays the pop-up menu for the entity listing the available navigation options.

- ▶ Double-click a row in the table to drill down directly into the default lower level detail layout view. This is a shortcut that allows you to drill down without using a menu option. The default drill down option is shown in bold in the entity pop-up menu.
- ▶ Select the navigation option from the Navigations pane.

The navigation links provide you with suggestions for how to further analyze a performance problem and diagnose the cause.

For example if you see that the status of a probe is red and in the Navigations pane you see that there are 5 server requests and the status is also red you may want to select the View Server Requests link in the Navigations pane to get more details on the server request's performance.



View the Diagnostics Profiler for a Probe Entity

Selecting **View Profiler for...** from the entity pop-up menu or from the Navigations pane causes Diagnostics to open the Diagnostics Profiler for the probe. If the entity is a Java probe, Diagnostics opens the Diagnostics Profiler for Java in a new window in the same browser. For details on the Diagnostics Profiler for Java, see Chapter 37, "Diagnostics Profiler for Java" for details.

If the entity is a .NET probe, Diagnostics opens the Diagnostics Profiler for .NET in a new browser. For details on the Diagnostics Profiler for .NET, see Chapter 38, "Diagnostics Profiler for .NET" for details.

Tasks

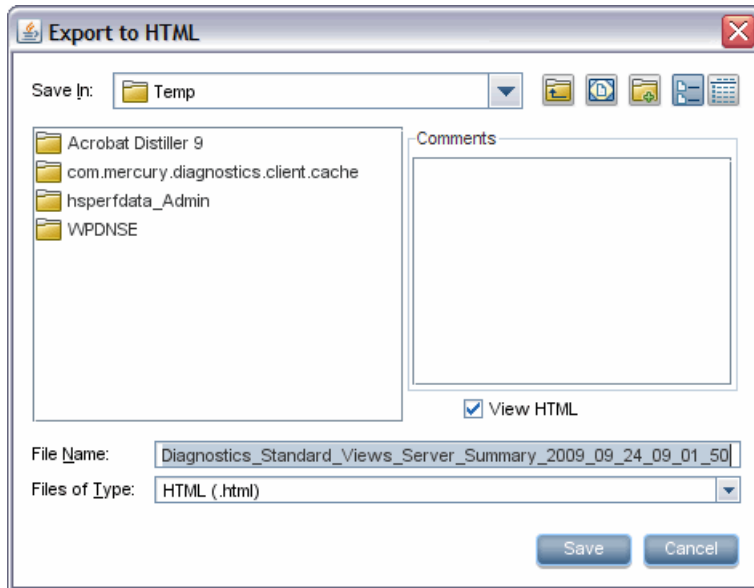
How to Save or Export Data



This task describes how to use the toolbar button to save or export view information to an HTML Web page or CSV file.

To save view information to an HTML Report:

- 1 Select **Save as HTML...** from the **Saves this view as an HTML web page or CSV file** drop down menu. The Export to HTML dialog box opens.



- 2 Enter a name for the file to which you are exporting the view's performance metrics.
- 3 In the **Comments** box, enter any comments that you would like to have appear on the report, for example, instructions or questions.
- 4 To view the newly created HTML file, select **View HTML**. This causes Diagnostics to display the HTML file in your browser after it has been saved.

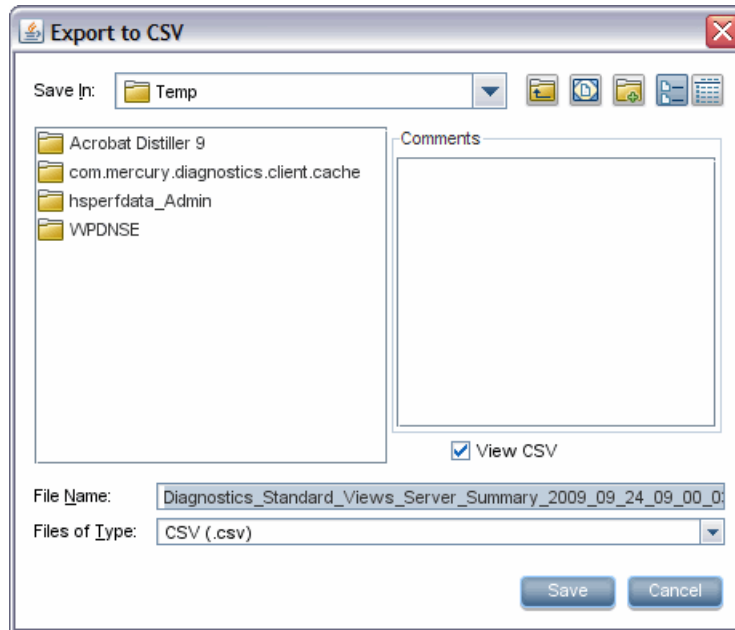
- 5 Click **Save** to save the performance metrics of the view, along with any comments that you entered. If you selected **View HTML**, Diagnostics displays the HTML report in your browser.

Note: If you select **Save as HTML...** from a tabbed view, data from only one of the tabs is saved. For example, with the Topology view, data from the Probe Connections tab is saved.

If you select **Save as HTML...** from a status type view, data from only the top level headings is saved. For example, with the Status view, data for the probe group is saved but not for the probe or host tables.

To save the data in the chart to a CSV file

- 1 Select **Save as CSV...** from the **Saves this view as an HTML web page or CSV file** drop down menu. The Export to CSV dialog box is displayed as shown below.



- 2 Enter a name for the file to which you are exporting the chart's performance metrics.
- 3 In the **Comments** box, enter any comments that you would like to have appear in the CSV file, for example, instructions or questions.
- 4 To view the newly created CSV file, select **View CSV**. This causes Diagnostics to display the CSV file in Microsoft Office Excel after it has been saved.
- 5 Click **Save** to save the charted metrics of the view, along with any comments that you entered. If you selected **View CSV**, Diagnostics displays the CSV file in Microsoft Office Excel.

When you save the data in a CSV file, only the chart data is exported. The Save as CSV menu item is not available for every view, it is dimmed when not available.

In Microsoft Office Excel you can format the data as you like. The first two columns show the date/time. The first one is the Excel serial date-time (GMT), the second column has the date-time in a string format (in the local timezone). Excel cell formatting can be used on either one. To show seconds, use a time format or a custom format. Also use cell formatting to format the numbers in the other columns as needed.

For example, select the **Data.Time** cell and select **Format > AutoFormat... > Classic 3** to display the data in easy to read columns.

The data displayed in the CSV file includes:

- **Description.** The heading data at the top of the CSV display gives a description of the chart and the filters you had set in the view you exported.
- **Time Range.** Time range for the data in the chart.
- **Customer Name.** The name of the customer you're logged in with. When in a LoadRunner/Performance Center run, it shows Run Start and Run ID instead of customer name.
- **Context.** How you navigated to the view you are exporting.
- **Data columns.** The columns of data that could be exported for the chart.
- **Min/Max/Avg.** The minimum, maximum and average values for each column.

- **Comments.** Any comments you entered in the Export to CSV dialog box.

When there are multiple charts displayed in a view, the data for all the charts is exported to the CSV file. The data can be displayed in a single table if the time range is the same for all the charts. The data will be displayed in separate tables if the time ranges vary.

If you have zoomed in on the chart, the data for this zoomed in time range is what gets exported to the CSV file.

Other export options:

You can use the **Data Export** function to export performance metrics directly from the Diagnostics Time Series database (TSDB), which is the repository for all persistent Diagnostics data. For more information on data export, see the *HP Diagnostics Installation and Configuration Guide*.

Diagnostics does provide an advanced **Query API** which you can use to create a more customized HTML dashboard. Documentation of the Query API is beyond the scope of these manuals but an example is provided in the `/contrib` directory of the use of the Diagnostics Query API. A readme is provided along with sample code. The Query API is accessed from the Diagnostics Welcome screen by selecting Configure Diagnostics, and then in the Components window selecting **query**.

How to Work with View Filters

This task describes how to use the View Filters.

This task includes the following:

- "Set a Filter" on page 132
- "Filter by Type of Data" on page 132
- "Filter by Probe Group" on page 132
- "Filter by Custom Filters" on page 136

Set a Filter

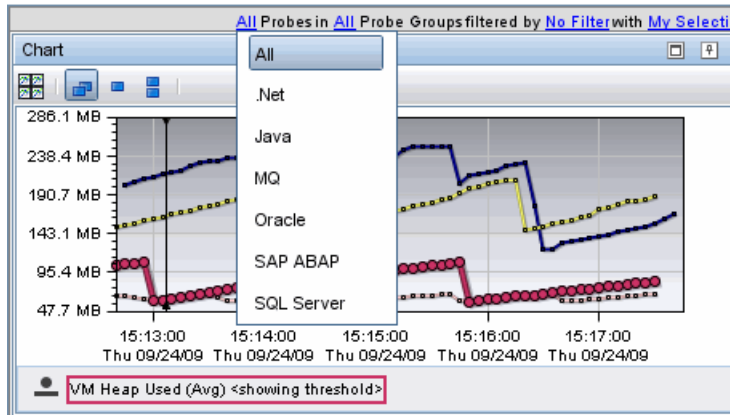
The view filters, a set of drop-down menus that are included in the view title of many of the Diagnostics views, are used to control the information displayed in the graph, graph entity table, and Details pane. The filters that appear in each view title vary according to the displayed view. To set a filter, click a link and select a value from the drop-down menu.

Filter by Type of Data

The **Type** filter appears in many of the Diagnostics views. This filter varies according to the type of entities in the view.

Example:

The following is an example of the **Type** filter menu for the Probes view:

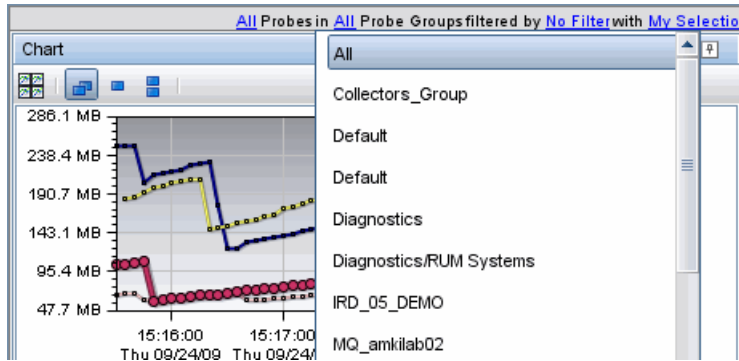


Filter by Probe Group

You use the **Probe Group** filter display metrics for a specific probe group, or for all probe groups that are defined for the monitored application.

Example:

The following is an example of a **Probe Group** filter menu:

**Filter by Graph Metrics**

You use the **Graph** filter to determine which how many entities from the graph entity table should have their metrics charted on the graph (for example Top 3 or Top 5). The choices in the **Graph** filter vary according to the type of entities in the detail layout. The metric in this filter is based on the sorted column (the metric column with a sorting control arrow) in the table.

These filters are based on the values of metrics or thresholds, and the filter is reapplied whenever the view is refreshed. For this reason, the graph entities and metrics will frequently change.

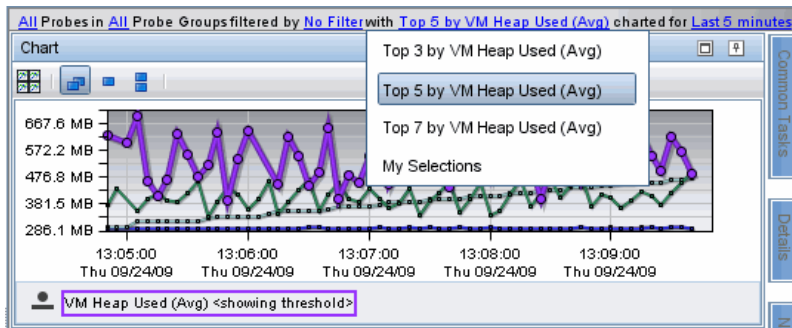
When you select any option (except **My Selections**) from the **Graph** filter menu, the entity selection in the **Chart** column of the graph entity table is modified so that only the entities that correspond to the graph filter are selected. At the same time, the graph is updated so that only the metrics for the entities selected in the **Chart** column are charted.

When you select **My Selections**, the entities that are selected in the **Chart** column of the graph entity table become fixed and change only when you chart different entities, or when you select another option from the **Graph** filter menu.

Note: An entity selected in the **Chart** column of the graph entity table may disappear from the table and the graph if no metrics for that entity were captured during the display time range.

Example:

The following example shows the **Graph** filter menu for the Probes view:



Filter by Time

If you do not modify this filter, the view time is determined by the time period set initially in the Diagnostics global **Viewing Time** filter, as described in Viewing Time Filter.

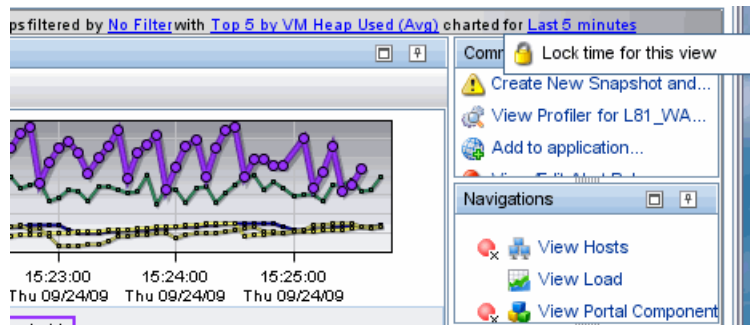
In this view, you use the **charted for** filter to lock a view at a specific time range so it will override any changes to global Diagnostics **Viewing Time** filter. Select **Use global time for this view** from the filter menu to unlock the time range for a view.

The value of the **charted for** filter in the view title indicates that the time range for the view is locked.

Note: If you are using Diagnostics with LoadRunner or Performance Center as part of a load test, the default time range is **Whole Scenario**, the time that has elapsed since the beginning of the load test.

Example:

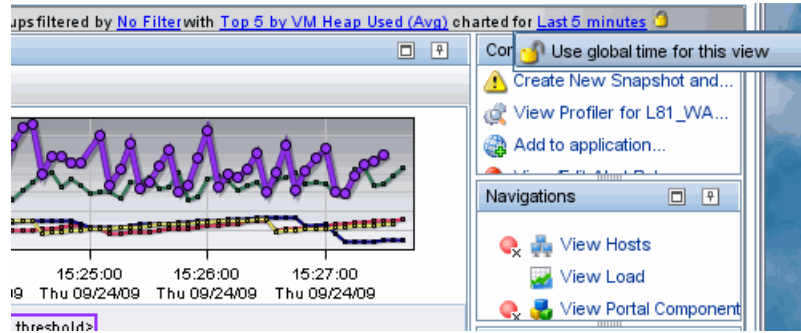
To lock the time range for a view, select the **charted for** filter in the view title and select **Lock time for this view** from the filter menu as shown below:



The value of the **charted for** filter in the view title indicates that the time range for the view is locked.

Example:

To unlock the time range for a view select the **charted for** filter in the view title and select **Use global time for this view** from the filter menu.

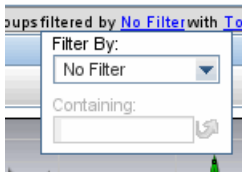


Filter by Custom Filters

You use the **Custom** view filter to display entities that match a specific value of an entity's attribute. For example, you can filter by probe groups containing Production. This filter varies according to the type of entities in the detail layout.

To select a Custom Filter:

- 1 Click the No Filter link to open the **Custom** filter menu.



- 2 Choose a filter from the **Filter By** drop down menu.
- 3 Enter a filter value. To view a list of applicable filter values, check the values in the graph entity table and the Details pane.
- 4 Apply the filter by clicking **Enter** or the Apply Change icon.



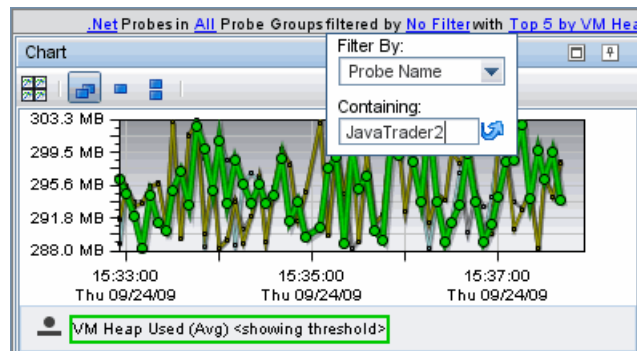
The following image shows an example of the **Custom** filter after a filtering attribute has been selected.



5 To remove a custom filter, select **No Filter** attribute and click **ENTER**.

Example:

The following is an example of the **Custom** filter menu:



How to Zoom In and Out on a Graph

This task describes how to use the zoom features on a graph.

This task includes the following steps:

- "Zoom In on Charted Metrics" on page 138
- "Zoom Out on Charted Metrics" on page 141

Zoom In on Charted Metrics

You can zoom in to see more detail for the metrics in a particular time period on the graph.

Click the graph at the beginning of the time range on which you want to zoom in, and drag the pointer to select the zoom range.

As soon as you move your mouse after you have clicked, the pointer changes to a zoom icon that indicates that you are zooming in on the metrics, and the background of the selected zoom range darkens.

The zoom icon that you see indicates whether Diagnostics is using a magnification zoom or a resolution zoom.



- For a description of magnification zooming, see "Magnification Zooming" on page 107.



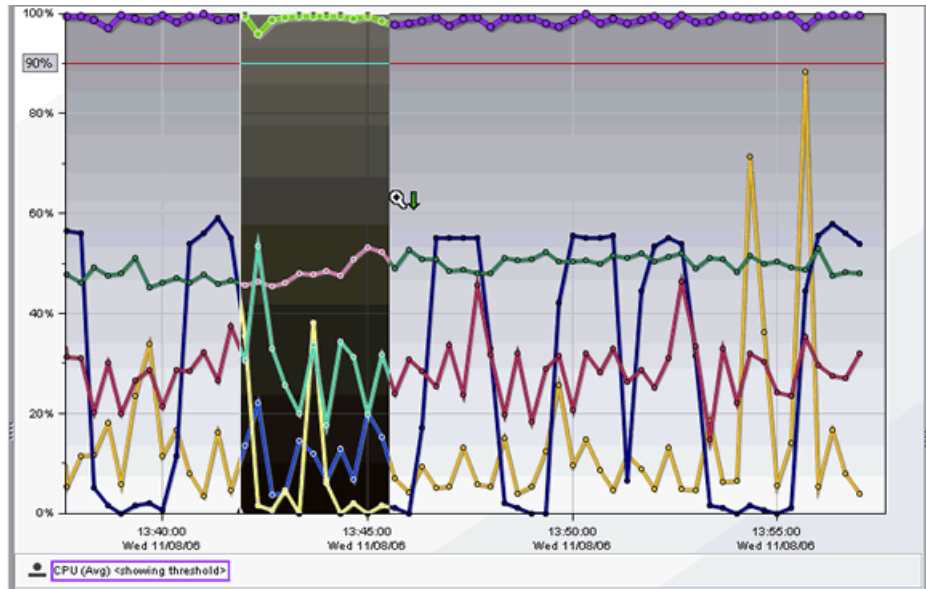
- For a description of resolution zooming, see "Resolution Zooming" on page 107.

The **Resolution Zoom** icon switches to the **Magnification Zoom** icon when Diagnostics determines that the zoom range that you specified requires too much higher-resolution data to be retrieved. Select a smaller zoom range if you would rather zoom using the resolution zoom.

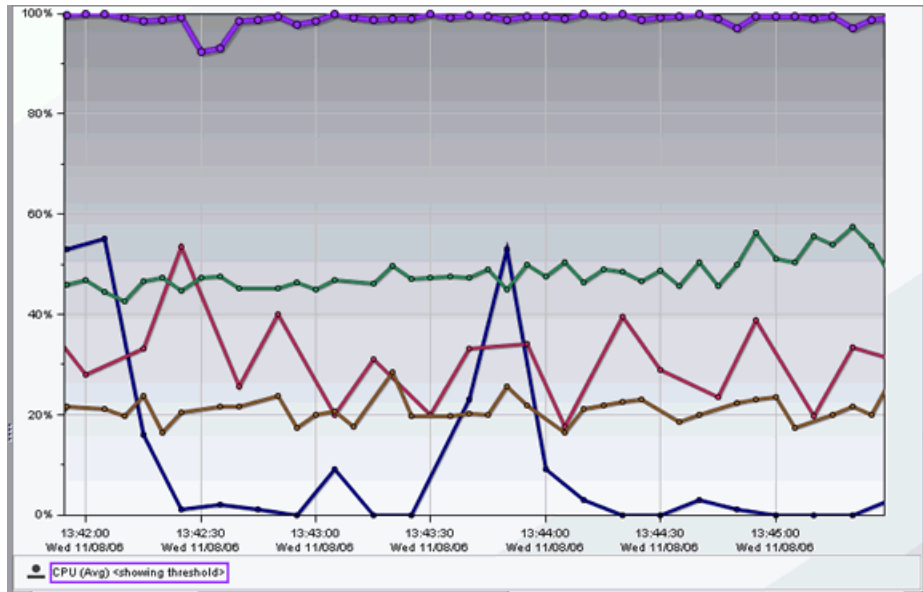
Release the mouse button to indicate that you have marked the desired zoom range. Diagnostics zooms in on the selected range and displays the metrics.

Example:

The following example shows a zoom range request where Diagnostics is performing a resolution zoom:



The following image shows the graph after zooming in:



Note the following on the graph:

- ▶ Diagnostics displays the metrics for the time range that you requested when zooming in. The requested zoom range fills the entire visible graph. Use the x-axis labels on the graph to determine the range and resolution of the metrics. Both the x-axis and y-axis are rescaled to correspond to the data in the zoomed range. In the example above, the zoom range is approximately **13:42** through **13:45**.
- ▶ Diagnostics retrieves the higher resolution metrics for at least the time range that you indicated. It may also retrieve metrics for a larger time range. Diagnostics makes this additional data available to you by adding a scroll bar under the graph. You may use the scroll bar to view the data outside of the zoom range that you requested. In the example above, the scroll bar that was added allows you to scroll to view data prior to **13:42** and after **13:45**.

Note: When the scroll bar is visible under the graph, you can also right-click the graph and drag the pointer back and forth to scroll the graph.

- ▶ If a resolution zoom is used, the selection in the **Time Range** view filter is set to the specific time range for the zoom range you selected.

Zoom Out on Charted Metrics



To zoom out of a graph that you previously zoomed in on, click the **Zoom Out** button.

- ▶ Clicking the **Zoom Out** button causes Diagnostics to zoom out to lower resolution data. The time period for the original zoom range that was displayed when you started to zoom out continues to be included in the wider, lower resolution data charted on the graph.

Note that the Zoom Out button will always go the next resolution in the time range filter when one of the fixed time ranges is selected, but if you are in a custom time range (as often happens when you zoom in), the zoom out function will just keep increasing the time range of the custom option. It will never go back to one of the fixed time ranges.

- ▶ As an alternative you can use the **Fit data to size** button which will undo any magnification zooms at the current resolution.
- ▶ Or you can use the time range filter to go back to whatever time range you were originally looking at before you zoomed in on the data.

How to Customize Table Columns

This task describes how to customize table columns.

This task includes the following steps:

- ▶ "Right-click in the table header to display the header menu." on page 142
- ▶ "Select Columns to Make Visible" on page 142

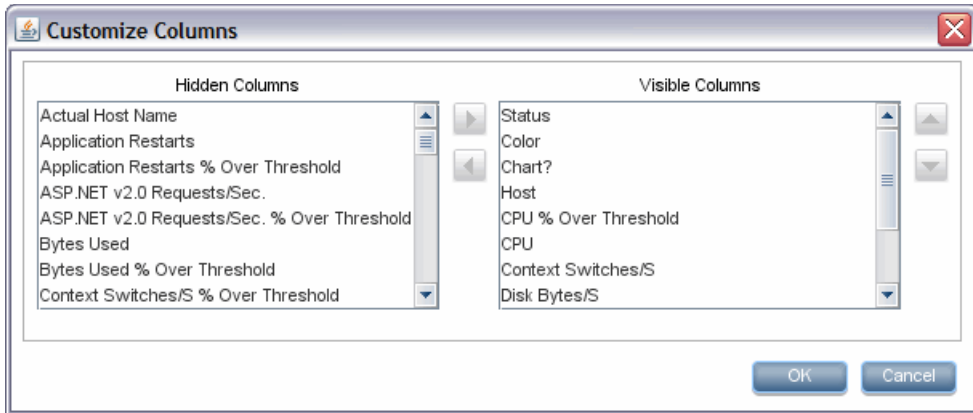
- ▶ "Select Columns to Hide" on page 143
- ▶ "Rearrange Columns" on page 143
- ▶ "Select OK and Save Your Changes" on page 143

1 Right-click in the table header to display the header menu.

Click **Customize Columns** to open the Customize Columns dialog box.

Example:

The Customize Columns dialog box for the Probes view is shown in the following example:



2 Select Columns to Make Visible



Make additional columns visible in the table by selecting one or more columns in the **Hidden Columns** list and using the right-arrow button to move your selection to the **Visible Columns** list. Repeat this step until all of the columns that you want to make visible in the table have been moved.

Note: Diagnostics automatically adjusts the width of the columns in the table to make all of the selected columns visible. Some columns may appear too small to see the data. You can resize the column or hold the mouse pointer over the column to see the tooltip with the value of a particular metric in the table.

3 Select Columns to Hide



Hide columns from the table by selecting one or more columns in the **Visible Columns** list and using the left-arrow button to move your selection to the **Hidden Columns** list. Repeat this step until all of the columns that you want to hide in the table have been moved.

4 Rearrange Columns



Rearrange the order in which the columns are displayed in the table by selecting one or more columns in the **Visible Columns** list, and using the up arrow or down arrow buttons to move the column. Repeat this step until the columns in the table are in the desired sequence.



5 Select OK and Save Your Changes

Click **OK** to close the Customize Columns dialog box and save your changes. Click **Cancel** to close the Customize Columns dialog box and discard your changes.

When the table columns are organized to your satisfaction, you can resize columns or change the sort order to refine your column customizations.

Note: If you customize one of the standard detail layout views, and would like to save the customizations for future use, save the view as a custom view. If you do not save the changes as a custom view, they are lost when you close the Diagnostics UI.

For details on saving a custom view, see "How to Save a Custom View" on page 227.

How to Customize Table Sort Order

This task describes how to customize the sort order using the controls built into the column headers in the table.

This task includes the following steps:

- ▶ "Select the Column to be Used for the Primary Sort" on page 144
- ▶ "Switch from Ascending Order to Descending Order (Optional)" on page 145
- ▶ "Remove a Column from the Sort (Optional)" on page 145
- ▶ "Add a Secondary Sort Sequence (Optional)" on page 145

1 Select the Column to be Used for the Primary Sort

Most of the Diagnostics tables are sorted in ascending order on the first metric column in the table, and then ascending order on the entity name column.

Click the header of the column that is to be used for the primary sort.

The indicators for the previous sort sequence are removed from all columns in the table.

The selected column is marked with a superscript up arrow and the number **1**, indicating that the column is sorted in ascending order, and that the values in the column determine the primary sort sequence.

2 Switch from Ascending Order to Descending Order (Optional)

To switch from ascending order to descending order click the column header again. The up arrow switches to a down arrow, indicating that the column is sorted in descending order.

3 Remove a Column from the Sort (Optional)

To remove the column from the sort, click the header one more time. The superscript sort indicator is removed from the column.

Note: When you remove the primary sort column from the sort, any secondary sort orders that you have defined are removed as well.

4 Add a Secondary Sort Sequence (Optional)

Press CTRL and click the header of the column that is to be used for the secondary sort. The selected column is marked with an superscript up arrow and a number. The up arrow indicates that the column is sorted in ascending order, and the number indicates where the column falls in the sort sequence.

If this is the first column that you have defined after defining the primary sort column, then the number **2** appears next to the arrow. To any subsequent columns that you add to the sort sequence, an increment superscript is displayed to indicate the sorting order.

To switch from ascending order to descending order, press CTRL and click the header again. The up arrow changes to a down arrow, indicating that the column is sorted in descending order.

To remove the column from the sort, press CTRL and click the header one more time. The superscript sort indicator is removed from the column.

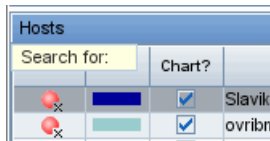
Note: When you remove a secondary sort from a column, any other columns that are used as secondary sort columns are renumbered to maintain the sorting sequence.

How to Find an Entity in the Table

This task describes how to search for and find an entity (such as a probe) in the graph entity table.

To find an entity in the graph entity table:

- 1 Press CTRL + F. Diagnostics displays the **Search For:** pop-up above the graph entity table column headers as shown in the following image:



- 2 Type all or the first part of the name of the entity that you would like to locate. As you type each character, Diagnostics begins the search. To locate an entity starting with a / character you must include the / in the search field.
- 3 If a match is found, Diagnostics selects the first row fitting the search criteria from the graph entity table.
- 4 If no match is found, Diagnostics changes the color of the text in the **Search For:** pop-up to red, and adds the **(no match)** string, following your search criteria.
- 5 To exit the search, press **Esc**.

How to Select Multiple Rows in the Table

If you want to apply a common task to multiple items you can select multiple rows in a table and then select one of the following types of tasks:

- Add to Application
- Remove from Application
- Delete Alert Rule (delete from the Alert Rule dialog box or delete entity status rules from the graph entity table)
- Create or Replace Alert Rule (entity status rules from the graph entity table)
- Delete Comments
- Create or Replace Comments
- Create New Snapshot and Add or Add to Active Snapshot

To select multiple rows in the graph entity table:

- 1** Select a row in the graph entity table.
- 2** Scroll down to select all the rows in between and select **SHIFT + left mouse click**.

Or hold the **SHIFT** key down and use the arrow keys to select multiple contiguous rows.

Or press **CTRL + left mouse click** to select multiple individual rows in the table.

Or press **CTRL + A** to select all rows.

- 3** Right-click to select a task from the menu of options you can apply to the multiple selected rows.

This is supported in all tables except Alert events, status views and the probe groups of the Server Summary screen.

Only those tasks that can be applied to all selected entities will be available in the menu.

For example in the Probes view if you select 5 probes with active alert rules set, you will see menu items for Replace Alert Rule and Delete Alert Rule. If you select probes with no alert rules set you will see the Create Alert Rule menu option. If you select a combination of probes with alert rules set and probes with no alert rule set or disabled alert rules, you will not see any alert rule options in the menu.

When you multiselect many entities then select **Remove from Application**, the message displayed (in hover text) is for a single entity. However what will actually be removed is - all selected entities from the application. To avoid confusion you would most likely use the multi-line remove when you want to select everything and remove from a specific application.

4

Working with Thresholds and Metrics in the Details Pane

This chapter describes the Details pane in Diagnostics views.

This chapter includes:

Concepts

- About the Details Pane on page 150
- Custom Attributes on page 154
- Chart a Metric in the Graph on page 156
- Include a Metric in a Snapshot on page 157
- Set an Alert Rule for a Metric on page 157
- About Metric Thresholds on page 158
- Threshold Definitions on page 158
- View Metric History on page 162
- Query Page to View All Thresholds Set on page 162
- Audit Log of Threshold Changes on page 163
- Using Thresholds in Monitoring Business Applications on page 163
- Investigating Threshold Violations on page 165
- Probe CPU Utilization Metrics on page 167
- Configure Metrics on page 168

Reference

- Set Threshold Dialog Box Description on page 173

Concepts

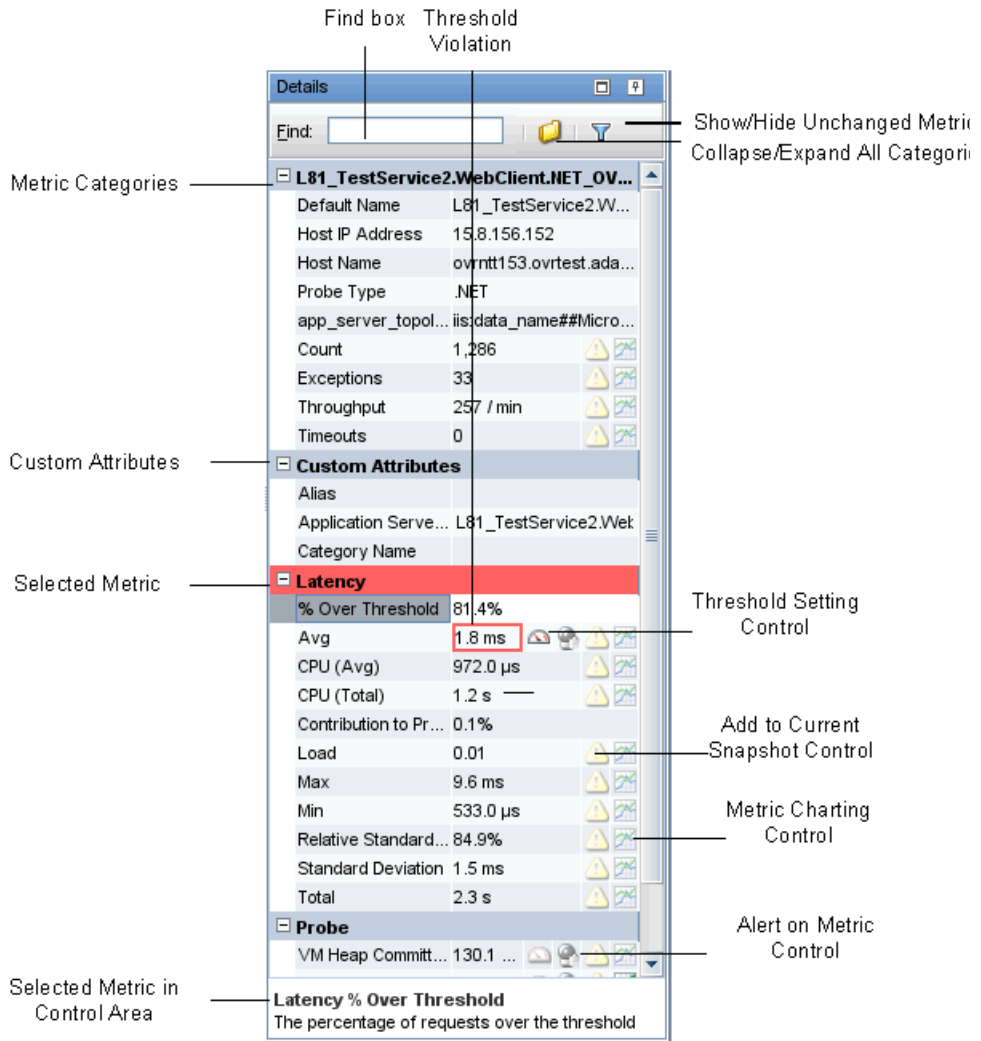
About the Details Pane

The Details pane in Diagnostics views displays the performance metrics collected by the probe for the selected entity. The Details pane is also where you select which metric to graph, set metric thresholds to be used in determining the status of an entity and you can also define rules for alert triggers and notifications.

Using the Details pane, you can do the following:

- ▶ Control the metrics that Diagnostics charts in the graph.
- ▶ Specify the metrics included in a snapshot.
- ▶ Set thresholds for the metrics.

The Details pane has a table with a list of metrics and their values, grouped by metrics categories. Note that one of the metric categories represents the applications the entity you selected in the graph entity table belongs to. The metrics displayed in the table can be, for example, Java metrics, system metrics, or JMX metrics.



Note: For information on configuring the system metrics or JMX metrics that appear in the Details pane, see the *HP Diagnostics Installation and Configuration Guide*.

This section also includes:

- "Descriptive Entity Information in the Details Pane" on page 152
- "Metrics Categories in the Details Pane" on page 152
- "Show or Hide Metrics that Have Not Changed" on page 153
- "Collapse or Expand All Categories in the Details Pane" on page 153
- "Search/Find Metrics in the Details Pane" on page 154
- "Metric Definitions and Icons for Optional Actions" on page 154

Descriptive Entity Information in the Details Pane

The Details pane displays the descriptive information and detailed performance metrics for the entity you selected in the Graph Entity Table.

The header of the first category in the Details pane displays the name of the selected entity.

Metrics Categories in the Details Pane

Diagnostics groups the entity attributes in the Details pane into categories to make it easier to understand the performance characteristics of the selected entity and to make it easier for you to find an attribute in the listed metrics. You can expand or collapse the list of metrics in a category.

More details about the probe entity that was selected from the graph entity table are provided in the first category in the Detail pane. The Custom Attributes category includes entries you can customize for the entity (see "Custom Attributes" on page 154). The metric category **Latency** contains metrics that depict the latency for processing server requests on the probe. Additional metric categories specific to the type of entity are also shown in the Details pane.

Show or Hide Metrics that Have Not Changed

By default, the Details pane shows all metrics during the time period set in the **Viewing Time** filter of the graph.



- To filter out the metrics that have not changed, click the **Hide metrics which have not changed** button, located at the top right corner of the Details pane. Enlarge the Details pane until you see this button if it is not visible.



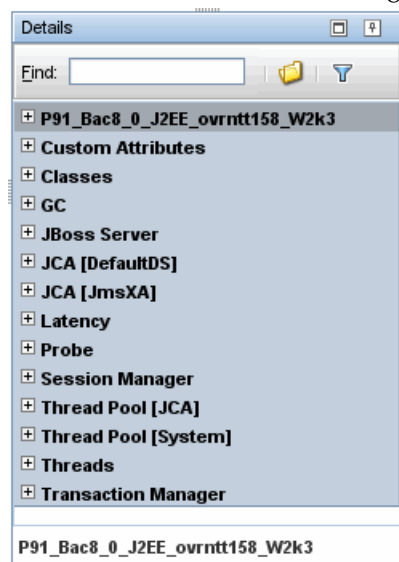
- To display all the metrics, including those that have not changed, click the **Show all metrics** button, located at the top right corner of the Details pane. Enlarge the Details pane until you see this button if it is not visible.

Collapse or Expand All Categories in the Details Pane

By default, the Details pane shows all metrics categories expanded.



- To collapse all categories in the Details pane, click the **Collapse all categories** button, located at the top right corner of the Details pane. When there is a very long list of metrics, collapsing the categories makes it easier to find the category you are looking for.



Enlarge the Details pane until you see this button if it is not visible.

- Click the button again to expand all categories.

Search/Find Metrics in the Details Pane

You can search for a metrics in the Details pane, using the **Find** box

To find a metric in the details pane:

In the **Find** box (situated at the top of the Details pane), type all or the first part of the name of the metric that you would like to locate.

- If a match is found, Diagnostics selects the first row in the Details pane that matches the search criteria.
- If no match is found, the color of the Find box changes to red.

Metric Definitions and Icons for Optional Actions

When you select a metric in the Details pane, the metric's definition appears in the metric control area at the bottom of the Details pane.

In the Details pane you will see an icon next to the metric for each type of setting applicable to the metric. The following optional settings are available for selected metrics:

- Threshold icon indicates the metric can have an associated threshold.
- Charting icon indicates the metric can be charted in the graph.
- Snapshot icon indicates the metric can be included in a snapshot.
- Alert icon indicates you can define alert rules based on a metric status change.

Custom Attributes

In the list of metrics there is a category called **Custom Attributes** that lists attributes associated with the entity you selected in the graph entity table.

The attributes that are listed under Custom Attributes vary depending on the current view. Most views have an **Alias** attribute and a **Category Name** attribute. Using the **Alias** attribute you can assign names to entities so that they are more recognizable in the Diagnostics views. The **Category Name** attribute allows you to assign a name to an entity that can be used to filter and group the information that is displayed in a Diagnostics view. Diagnostics SOA Services - Operations views include a **Web Service Alias** attribute that allows you to change the displayed name of a Web service.

How to Set a Custom Attributes

You can assign values to the attributes in the Custom Attributes category of the Details pane by selecting the row with the attribute and entering the values directly into the field in the row. The value that you enter is applied to all occurrences of the entity throughout Diagnostics.

To alter the scope of the change so the change applies to either the selected occurrence or to all occurrences, you use the pop-up dialog boxes to assign values to the custom attributes.

You need to have **change** permissions to set custom attribute values. For more information about user permissions, refer to the *HP Diagnostics Installation and Configuration Guide*.

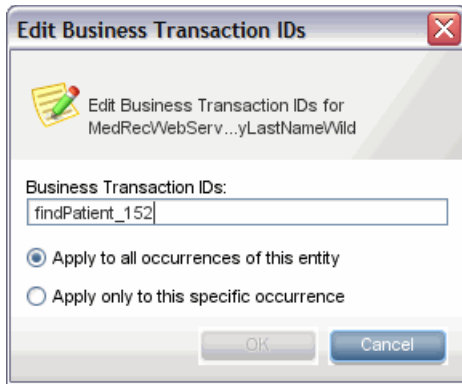
Note: The scope for custom attributes is limited to the occurrences of the entity for a given Diagnostics Server in Mediator mode. Occurrences of the entity that are captured by probes reporting to a different Diagnostics Server in Mediator mode will not be impacted by the custom attributes.

To define the scope of the custom attributes:

- 1 Click in the column to the right of the custom attribute that you want to update from the Details pane.



- 2 Click the button that appears at the end of the selected custom attributes row. Diagnostics displays a dialog box similar to the following:



- 3 Enter the value for the attribute.
- 4 Select the scope for the attribute, **Apply to all occurrences of this entity** or **Apply only to this specific occurrence**.
- 5 Click **OK** to accept and apply your changes.

Chart a Metric in the Graph

When the **Metric Charting** icon appears in a row in of the Details pane, the metrics in the row can be charted in the graph.



When the icon does not have check mark, the metric is not charted in the graph. You can cause the metric to be charted by clicking the **Metric Charting** icon in the row or by right-clicking on the row and selecting **Start Charting this Metric**.



When the icon includes a check mark, the metric is charted in the graph. You can remove the metric from the graph by clicking the **Metric Charting** icon in the row or by right-clicking on the row and selecting **Stop Charting this Metric**.

Include a Metric in a Snapshot



The rows that contain metrics that can be included in a Diagnostics snapshot are marked with the Snapshot Analysis icon. The Snapshot Analysis icon behaves differently depending on whether you are in Snapshot Analysis mode (the Analyze Snapshots tab is selected) or Monitoring mode (the Monitor and Investigate tab is selected).

Including a Metric in a New Snapshot

When you are looking at a Diagnostics view in Monitoring mode, clicking the Snapshot Analysis icon causes Diagnostics to create a new snapshot that includes the entity selected with the metric selected from the Details pane charted in the graph.

Including a Metric in an Active Snapshot

When you are looking at a Diagnostics view while in Snapshot Analysis mode, clicking the Snapshot Analysis icon adds and removes the metric selected from the Details pane from the snapshot.



When the Snapshot Analysis icon includes a check mark, clicking the icon causes Diagnostics to remove the selected metric from the active snapshot.



When the Snapshot Analysis icon does not include a check mark, clicking the icon causes Diagnostics to add the selected metric to the active snapshot. This metric will then be charted in the active snapshot.

Set an Alert Rule for a Metric

When the **Alert Rule** icon appears in a row in of the Details pane, you can set an alert rule for the metric.

There are two types of alert notification rules:

- ▶ Those based on entity status change occur when any metric for the entity violates a threshold.
- ▶ Those based on metric status change occur only when the specified metric violates a threshold.

Alert rules based on metric status change are defined from within the Details pane. Only those metrics that can be used in an alert rule will show the Alert Rule indicator icon.



When the **Alert Rule** icon is greyed out, no alert rule has been set but the metric can be used in an alert rule. You can create an alert rule for the metric by clicking the Alert Rule icon in the row or by right-clicking on the row and selecting **Create Alert Rule for Metric**. Use the Alert Rule dialog box that is displayed to define an alert rule for the metric.

For details, see Chapter 5, "Working with Alerts".



When the Alert Rule icon is highlighted (red), the metric has an alert rule set. To edit or delete an alert rule for the metric, right-click the Alert Rule indicator icon and select **Edit Alert Rule for Metric** or **Delete Alert Rule for Metric**.



If an alert rule has been set for a metric but the alert notification is disabled the Disabled Alert Rule icon is displayed.

About Metric Thresholds



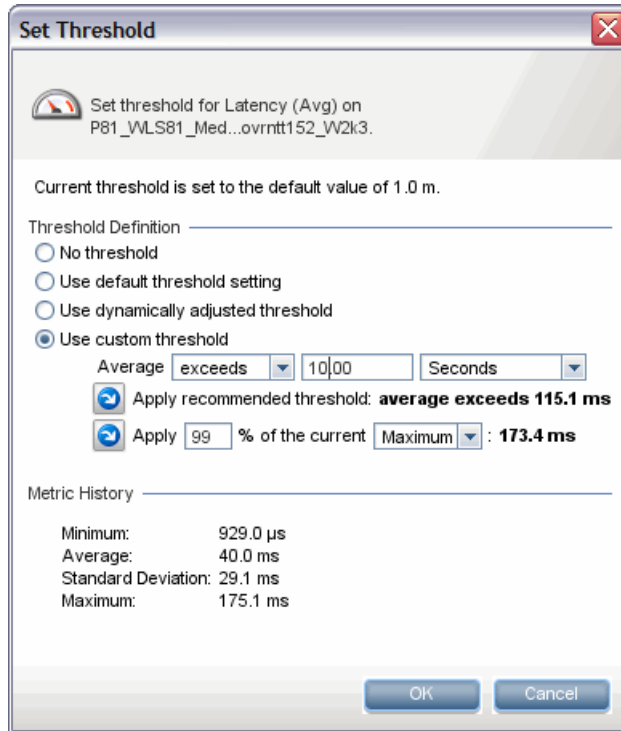
When the **Metric Threshold** icon appears next to a metric in the Details pane, the selected metric can have an associated threshold. If the icon is dimmed, the threshold has not yet been set for the metric. If the icon is active, a threshold value has been set for the metric.

To set or edit a threshold, click the Metric Threshold icon or right-click the metric in the details pane and select **Set Threshold for Metric**. The Set Threshold window is displayed. See "Set Threshold Dialog Box Description" on page 173 for more information.

Threshold Definitions

There are a number of possible threshold definition options you can use in the Set Threshold dialog box, though not all options are appropriate for all metrics.

An example of the Set Thresholds dialog box is shown below:



The options are:

- **No threshold.** This radio button will be selected if no threshold has been set for a metric. If no threshold is set, there will be no alerts and no status displayed for the entity. However, if there is no threshold for a 'parent' entity a status may still be displayed based on the status of 'child' entities (for example, probes and server requests or Web services and operations).

Once a threshold is set, you can select the **No Threshold** option to remove the threshold. After you've removed a threshold you can always come back later and set a threshold.

- **Use default threshold setting.** If selected, this radio button sets the threshold to the default value in the `<server>/etc/thresholds.configuration` file. Selecting this option clears any user-defined threshold settings. If there is no default threshold for this metric in the property file, then no threshold value is set, by default, on this metric. When there is no default, you should select another option such as Use Custom Threshold to set a threshold for the metric. After selecting the Use Default Threshold Setting option you can see what default value was used by exiting the Set Thresholds dialog box and re-entering the screen. The message at the top of the dialog will indicate, for example: Current threshold has a system-provided default value of 0.10.

The `thresholds.configuration` file contains default threshold values for different entity types. The file also specifies DYNAMIC as the default setting for latency metrics. Changes to the default settings are made in this property file. See an example below:

```
#####
### Default thresholds for metrics.
###
### Syntax is
### <metric-parent-node-classname>.<metric-name> = [-] value
### or
### <collector-name>.<metric-name> = [-] value
###
### Value is interpreted as a double. value of '0' is not allowed.
### Negative value indicates it is a lower-bound threshold.
#####
# Threshold on avg. latency of fragments (in microseconds)
com.mercury.diagnostics.common.data.graph.node.FragmentData.latency = 60000000:DYNAMIC

# Threshold on avg. latency of web service operations (in microseconds)
com.mercury.diagnostics.common.data.graph.node.WebServiceData.latency = 60000000:DYNAMIC

# Threshold on avg. latency of R3 Server Requests (in microseconds)
com.mercury.diagnostics.common.data.graph.node.R3ServerRequestData.latency = 60000000:DYNAMIC

# Threshold on avg. latency of methods (in microseconds)
com.mercury.diagnostics.common.data.graph.node.MethodData.latency = 60000000

# Threshold on avg. latency of Portal component operations (in microseconds)
com.mercury.diagnostics.common.data.graph.node.PortletData.latency = 60000000
```

The defaults cannot be changed in the Set Thresholds dialog box.

- **Use dynamically adjusted threshold.** This radio button is enabled only when you have selected a latency metric; for other metrics this option will be greyed out.

A dynamic threshold is automatically adjusted by the system based on historical data. The last seven days of data is used by default. If there aren't seven days of data yet, the default value for threshold is used initially and then adjusted afterwards based on historical data.

When a dynamic threshold is set, you will see the following description in the Set Threshold dialog box: Current threshold value of <value> is periodically adjusted by the system to fit historical data.

The dynamic threshold adjustment job runs once a day. It can be run manually on the probe from the Probe Administration Welcome page, select **Advanced Options** > **Scheduler** from the Scheduler page.

- **Use custom threshold.** Select this radio button to set a fixed, custom threshold value. Since the value is fixed, as application usage patterns change, it may be necessary to select this button again to set a different threshold for the changed usage patterns. You enter the custom threshold on the line below:

Average <exceeds/falls below> <value> <units>

You can make your selections and enter a value directly on this line or you can use the two actions described below to help you determine a custom threshold value. Note that you cannot have a value of zero.

- **Apply recommended threshold: average exceeds <value>.** This action is available only for latency metrics. It uses heuristics to recommend a threshold for the metric given the recent metric history statistics.



When you select the action button, the value recommended will be entered on the line above as the custom threshold value. The metric setting is static, so it may be necessary to reset it from time to time as your application or workload evolves.

- **Apply <percent> % of the current <minimum/maximum/average>: <value>.** This action like the apply recommended threshold action, helps you determine a custom threshold value. It can be useful for metrics where selecting a threshold by percent makes sense, e.g., latency metrics, size metrics.



When you select the action button, the value recommended will be entered on the line above as the custom threshold value. The metric setting is static, so it may be necessary to reset it from time to time as your application or workload evolves.

View Metric History

The metric history panel in the Set Threshold dialog box shows up to six hours of statistical history about the metric.

The four statistical history values that may be shown are:

- ▶ Minimum value
- ▶ Maximum value
- ▶ Average value
- ▶ Standard deviation

This information is provided to give you some guidance in selecting a good threshold for the metric, that is, a threshold that will trigger when there is an emerging problem and not when there is simply a temporary spike in usage. Although these values can be helpful, you may want to look over the metric history for a longer period of time e.g., several months, to examine metric trends and behavior during incidents.

Not all metrics will have the same statistical metric values because Diagnostics does not keep certain statistics on all metrics. For example standard deviation is tracked for latency metrics, but is not tracked for host CPU utilization. If a historical statistic is not shown, it is because it is not tracked for that metric.

If the value of the selected metric exceeds the threshold value, a visual threshold violation indicator is immediately shown in the Details pane.

Query Page to View All Thresholds Set

The server has a query page that lists all custom and dynamic thresholds from all the mediator systems. You can access the query pages from the URL <http://<Diagnostics Server>:2006/query> or from the Diagnostics Server Administration page select **Configure Diagnostics > query** and then select **View Thresholds**. Note that the page may take a while to load. This query is used just for information by support in debugging alerts.

Audit Log of Threshold Changes

When a user changes a threshold, an entry is logged in the **server.log** file on the corresponding server. For example:

```
2008-03-03 90:27:18,071: INFO configure : User admin set latency metric
threshold to 15 seconds for entity L81_WLS91_MedRec_ovrnett155_W2K3
```

Using Thresholds in Monitoring Business Applications

Metric thresholds should be set to monitor the performance of important business applications. It is not necessary or desirable to set thresholds on every possible metric; in fact setting too many thresholds may be counter-productive, overloading the IT staff with alarms when there is no real problem, or distracting them with trivial issues when there is a more important problem to solve.

When set correctly, thresholds can be useful for:

- ▶ Verifying whether service level objectives are being met or not.
- ▶ Providing advanced warning when the application workload is changing due to an increasing demand that may result in degraded performance if application provisioning changes are not made.
- ▶ Notifying the IT staff when application service level objectives are in danger of not being met, or are not being met due to infrastructure problems, application problems, or workload changes.

To achieve these objectives, the business requirements for the application and its recent workload history should be considered. The business requirements should be considered first. Some questions to consider are:

- ▶ Is this an important application to the business customer?
- ▶ What is the expected average transactions/second? What is the expected minimum and maximum transaction rate?
- ▶ What is the desired average transaction response time? What is the maximum acceptable transaction response time?

- ▶ What is the business impact of failure to meet transaction rates and response times?
- ▶ How willing or able is IT able to deal with false alarms and at what cost? An overly aggressive threshold can result in more false alarms; an overly lax threshold can result in delaying recognition of emerging problems. There is an important cost/benefit trade-off to consider.

Once the business requirements are understood, the current application characteristics should be considered:

- ▶ How is the application structured?
- ▶ What are the key transactions? What are the key components supporting these transactions e.g., service requests, SQL statements, hosts, etc.
- ▶ How should the business requirements be mapped to lower level component threshold requirements? What thresholds should be set on which components? Not every component requires a threshold for good application performance management.
- ▶ What is the workload history over the past months? Can the business requirements be met with the current deployment or are changes required?
- ▶ How frequently should thresholds be reviewed and updated?

This stage will likely require a team effort from applications and operations experts.

Investigating Threshold Violations

When the value of a metric exceeds the threshold value that you have set, or in the case of a negative threshold, dips under the threshold, Diagnostics displays threshold violation indicators in the Details pane, in the status column of the graph entity table, and in the cell of the graph entity table where the metric is displayed.

In the Details pane, a threshold violation is indicated by outlining the cell that contains the metric value with the color that is appropriate for the severity of the violation. The row in the table that contains the category name for the metric that violated the threshold is highlighted with the same color as well.

The color highlight in the Details pane indicating a threshold violation determines its severity:

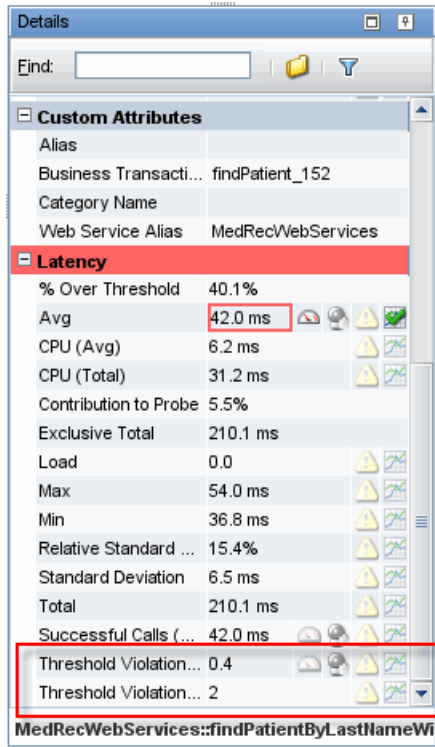
- ▶ **No outline around the metric or highlighting in the table.** Normal – There has been no threshold violation or there is no threshold defined for the metric.
- ▶ **Yellow.** Warning – The component is occasionally but not consistently exceeding the defined threshold.
- ▶ **Red.** Critical – The component is consistently exceeding defined thresholds.

The following describes, in general, how a status of RED or YELLOW is determined:

- ▶ Diagnostics compares the average value for the metric in the last 5 minutes to the threshold. If the threshold is crossed, the status is RED.
- ▶ If not, Diagnostics counts the total number of points (5-second averages from the probe) that violate the threshold. If the count is greater than three but the average is still under the threshold (current status is not RED), the status is YELLOW; otherwise the status stays at GREEN.
- ▶ If none of the above, the status is GREEN.

Status indicators can progress in severity from yellow and then to red as the metric's performance deteriorates. However, the indicators do not regress in the same way. Once a red indicator has been issued, it continues to be displayed as red until the metric value returns to normal levels.

In the following example, the cell containing the value of the **Latency (Avg)** metric is outlined in Red and the category for the metric (Latency) has a red background because the metric value, **22.3s**, exceeded the threshold.



You will see the **Threshold Violations (Avg)** and **Threshold Violations (Total)** metrics in the details pane, if you have set a CUSTOM latency metric threshold and there have been threshold violations and at least five minutes have elapsed since you set a custom latency threshold.

This shows the average and total number of latency threshold violations across ALL instances. (You may need to select the Show All Metrics filter button in the Details pane to see these two metrics if the you have Hide Metrics Which have not Changed selected).

You can set a threshold on the Threshold Violations (Avg) metric, as well as chart the metric and set an alert rule for the metric.

For example: if for a 5 minute period there are 12 samples and each sample has 2 instances and twelve of the instances have a violation then we have total=12, and avg=12/24 or .50. You could set a custom threshold that triggers an alert if the Threshold Violations (Avg) metric exceeds .50.

Note: If you want to set a threshold for the Threshold Violations (Avg) metric it must be set as a custom threshold because there is no default value.

Probe CPU Utilization Metrics

Diagnostics collects probe CPU utilization metrics per Java process and displays them under the **Probe** metrics section of the Details pane.

Probe					
VM Heap Free (Avg)	13.2 MB				
VM Heap Total (Avg)	130.9 MB				
VM Heap Used (Avg)	117.7 MB				
Availability % Under Threshold	5.3%				
Availability (Avg)	100%				
Long Requests (Avg)	0				
Normalized CPU Utilization (Avg)	1%				
Total CPU Utilization (Avg)	3%				

For a probe on a supported system you'll see the following metrics:

- **Normalized CPU Utilization (Avg).** The value for process CPU utilization normalized by the number of CPUs on the host.

This metric is useful when you want to compare the CPU utilization of two or more probes running on the same host and compare the probe CPU utilization with the system CPU utilization. In this case, the probe CPU utilization metric needs to be adjusted by the number of CPUs of the host so that each of the metrics is less than 100% and also consistent with the host system CPU utilization metric.

- ▶ **Total CPU Utilization (Avg)**. The value for process CPU utilization for all CPUs on the host.

This metric is useful when you want to compare the CPU utilization of the same application or application server running on different hosts. These hosts may have different numbers of CPUs. So in this case, using the probe's total CPU utilization (total of all CPUs) is useful for comparisons. Note that this total process CPU utilization metric is not adjusted by the number of CPUs so it may be greater than 100% if the probe is running on a multi-processor system.

Configure Metrics

You can control the metrics that appear in the Details pane by setting the appropriate Diagnostics properties. For information on configuring the system metrics or JMX metrics that appear in the Details pane, see the *HP Diagnostics Installation and Configuration Guide*.

You can configure CPU time metrics that appear in the Details pane as described in the following sections:

- ▶ ".NET Probe and CPU Time Metrics" on page 169
- ▶ "Java Probe and CPU Time metrics" on page 169

The CPU Time metrics appear in the Details pane for various views such as the Server Requests view, the Transaction view, the Probes view, the Call Profile view, the Portal Components view and the Diagnostics Profiler. You can enable, disable and configure the collection of CPU time metrics. The CPU time metrics are **CPU (Avg)** and **CPU (Total)**. If collection of CPU time metrics is disabled or not configured for methods, you will see N/A in the Details pane for these metrics.

The CPU Time metrics rely on CPU timestamping which is generally supported on the following platforms: Windows, Solaris, AIX, HP-UX and Linux kernels 2.6.10 or later (for example RedHat 5.x, SUSE 10.x).

Note: Support for CPU timestamping can vary however, not only by operating system, but also by platform architecture (for example SPARC versus x86).

Please see the Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp. for information on support for CPU Time on specific platform versions and architecture.

Caution: In VMware, the CPU time metric is from the perspective of the guest operating system and is affected by the VMware virtual timer. See the VMware whitepaper on timekeeping at http://www.vmware.com/pdf/vmware_timekeeping.pdf. Also refer to the *HP Diagnostics Installation and Configuration Guide* section on Time Synchronization for probes running on VMware.

By default, collection of CPU time metrics is enabled for server requests. You can disable CPU time metric collection and configure collection of CPU time metrics for both Java and .NET probes as described below. You can configure collection of CPU Time metrics for the following:

- Server Requests only
- Server Requests and Portlet Methods
- Server Requests and All Methods

.NET Probe and CPU Time Metrics

For a .NET probe the collection of CPU time metrics is controlled by the XML element **cputime** in **probe_config.xml**. The **cputime** element can be set to the following: none, serverrequest, method. The default is serverrequest.

Java Probe and CPU Time metrics

Collection of CPU time metrics can be configured in property files or using the Java Profiler UI.

For a Java Probe, configure the following property files for the collection of CPU Time metrics:

- **use.cpu.timestamps** property in `<probe_install_directory>\etc\capture.properties`.

This property is set to **true** by default which enables collection of CPU time metrics. But what, if any, CPU timestamps are collected is controlled by a second property listed below. If you set the `use.cpu.timestamps` property to false, the CPU time metrics will not be collected for any server request or method reported by the probe

- **cpu.timestamp.collection.method** property in `<probe_install_directory>\etc\dynamic.properties`.

Caution: Use caution when configuring the collection of CPU timestamps because of the increase in Diagnostics overhead. The increased overhead is caused by an additional call for each method that is needed to collect the timestamp.

Cpu.timestamp.collection.method can be set to one of the following:

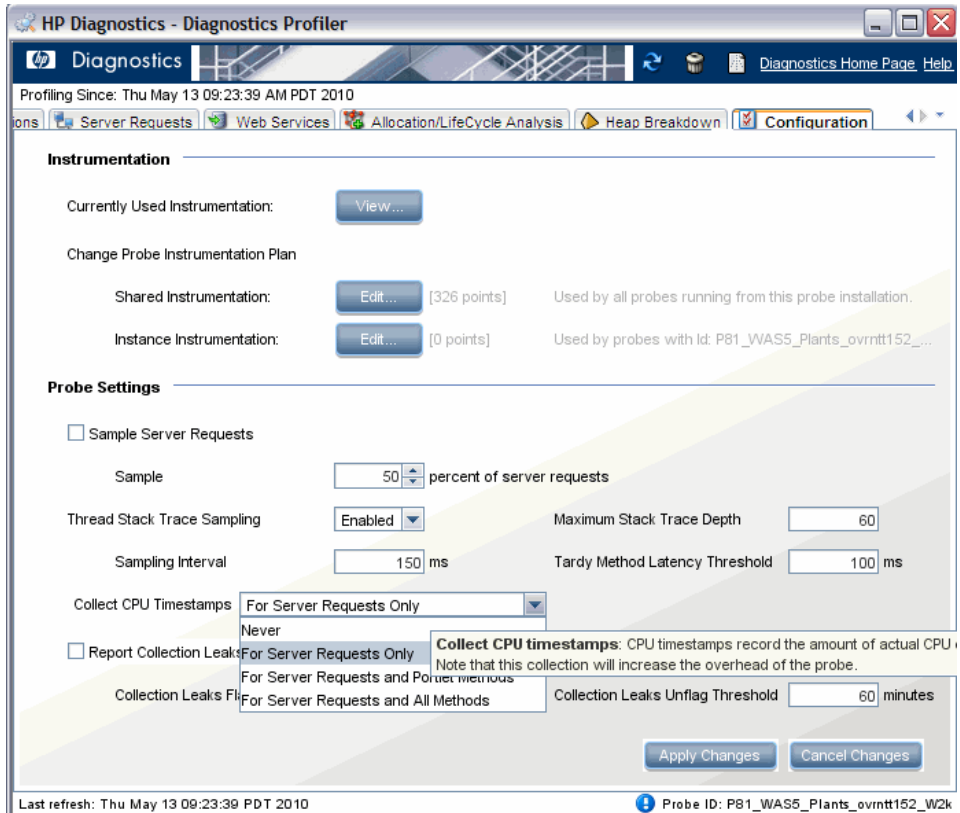
- **0** – No CPU timestamping
- **1** – CPU timestamps will be collected only for server requests.

The default value is **1** which means that CPU times can be reported at the server request level but not at the transaction level. However, if the setting is removed or commented out of the properties file, the default is then 0.

- **2** – CPU timestamps will be collected for All server requests and ALL methods.
- **3** – CPU timestamps will be collected for ALL server requests and the life cycle methods instrumented for Portal Components.

Or you can set the `cpu.timestamp.collection.method` property using the Configuration tab in the Java profiler as follows:

- 1 In the Profiler UI select the **Configuration** tab. The profiler does not need to be started to make this probe configuration change.



- 2 In the Configuration screen select a **Collect CPU Timestamps** option from the drop down list.

CPU Timestamp Collection Method	Description
None	No CPU Timestamps.
For Server Requests Only	CPU timestamps will only be collected for server requests.

CPU Timestamp Collection Method	Description
For Server Requests and Portlet Methods	CPU timestamps will be collected for ALL server requests and the life cycle methods instrumented for portal components.
For Server Requests and All Methods	CPU timestamps will be collected for ALL server requests and ALL methods.

- 3** When you have completed making changes to the Configuration tab, click **Apply Changes**.

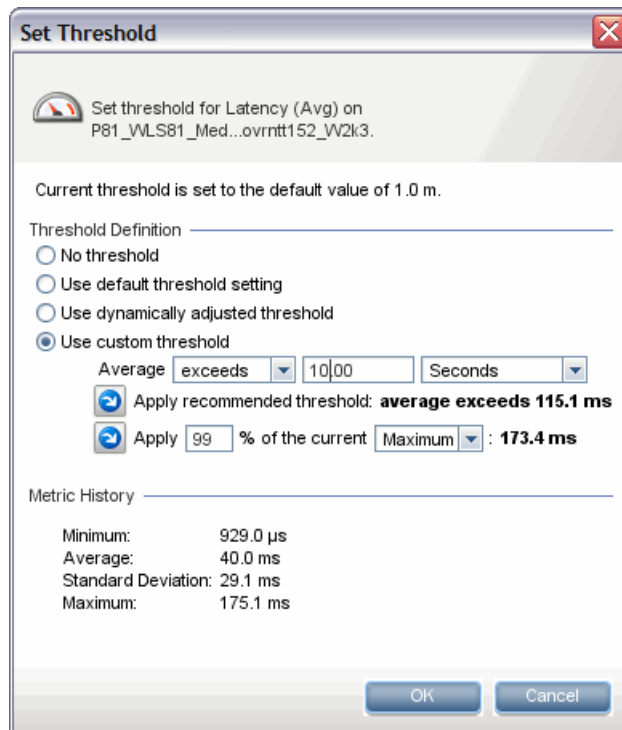
Note: Your changes take effect immediately. There is no need to restart the probe.

Reference

Set Threshold Dialog Box Description

Displays the Set Threshold dialog box, used to define thresholds and view metric history.


The following is an example of the Set Threshold dialog box for setting a latency threshold on a server request.



The dialog box is titled "Set Threshold" and contains the following information:

- Title:** Set Threshold
- Content:** Set threshold for Latency (Avg) on P81_VLS81_Med...ovrntt152_VW2k3.
- Status:** Current threshold is set to the default value of 1.0 m.
- Threshold Definition:**
 - No threshold
 - Use default threshold setting
 - Use dynamically adjusted threshold
 - Use custom threshold
 - Average: exceeds 10.00 Seconds
 - Apply recommended threshold: **average exceeds 115.1 ms**
 - Apply 99 % of the current Maximum: **173.4 ms**
- Metric History:**

Minimum:	929.0 μ s
Average:	40.0 ms
Standard Deviation:	29.1 ms
Maximum:	175.1 ms
- Buttons:** OK, Cancel

To access 	To set or edit a threshold,click the Metric Threshold icon or right-click the metric in the details pane and select Set Threshold for Metric .
Important information	For information on configuring the system metrics or JMX metrics that appear in the Details pane, see the <i>HP Diagnostics Installation and Configuration Guide</i> .
Relevant tasks	See "Chart a Metric in the Graph" on page 156
See also	► "Threshold Definitions" on page 158

The following user interface elements are included:

UI Elements	Description
Threshold Definition	<p>There are a number of possible threshold definition options, though not all options are appropriate for all metrics. The options are:</p> <p>No threshold. This radio button will be selected if no threshold has been set for a metric. If no threshold is set, there will be no alerts and no status displayed for the entity.</p> <p>Use default threshold setting. If selected, this radio button sets the threshold to the default value in the <code><server>/etc/thresholds.configuration</code> file. Selecting this option clears any user-defined threshold settings. After selecting the Use Default Threshold Setting option you can see what default value was used by exiting the Set Thresholds dialog box and re-entering the screen. The message at the top of the dialog will indicate, for example: Current threshold has a system-provided default value of 0.10.</p> <p>Use dynamically adjusted threshold. This radio button is enabled only when you have selected a latency metric; for other metrics this option will be greyed out.</p> <p>When a dynamic threshold is set, you will see the following description in the Set Threshold dialog box: Current threshold value of <code><value></code> is periodically adjusted by the system to fit historical data.</p>

UI Elements	Description
Metric History	<p>The metric history panel shows up to six hours of statistical history about the metric. The four statistical history values that may be shown are:</p> <ul style="list-style-type: none">▶ Minimum value▶ Maximum value▶ Average value▶ Standard deviation

5

Working with Alerts

This chapter describes how to create alert rules and notifications and view alerts in Diagnostics.

This chapter includes:

Concepts

- ▶ About Alerts on page 180
- ▶ Alert Properties on page 181
- ▶ About Alert Notification Rules on page 183
- ▶ Alert Event and Alert Notification Logs on page 184

Tasks

- ▶ How to Access the Alert Properties Page on page 186
- ▶ How to Enable and Disable Alert Notifications on page 187
- ▶ How to Configure SNMP or Email Alert Notifications on page 188
- ▶ How to Create or Maintain Alert Notification Rules for an Entity on page 189
- ▶ How to Create or Maintain Alert Notification Rule on a Metric on page 190

Reference

- ▶ Alert Notifications User Interface on page 194

Concepts

About Alerts

Diagnostics indicates that a threshold alert condition has occurred in the following ways:

- ▶ By changing the color of the status indicator to red or yellow when displayed in the Status view, the Alert Rules view, and the graph entity table of the views with the detail layout.
- ▶ By adding a red or yellow border to the cells of the graph entity table and the details pane that contain the metric values that have exceeded their thresholds.

For details on setting thresholds, see "About Metric Thresholds" on page 158.

In addition to the threshold alert indicators that Diagnostics displays in the views, you can set up rules that instruct Diagnostics to send alert notifications via e-mail or SNMP when an entity's status becomes critical.

There are two types of alert notification rules:

- ▶ Those based on entity status change occur when any metric for the entity violates a threshold.
- ▶ Those based on metric status change occur only when the specified metric violates a threshold.

Diagnostics can trigger alert notifications when the status of the entity (or metric) changes to critical and also when the entity becomes unavailable for a period of five minutes.

Note: Alert notifications for a given entity are sent once, at the time that the entity's status becomes critical. An entity that remains in critical status will not continuously cause the generation of alert notifications.

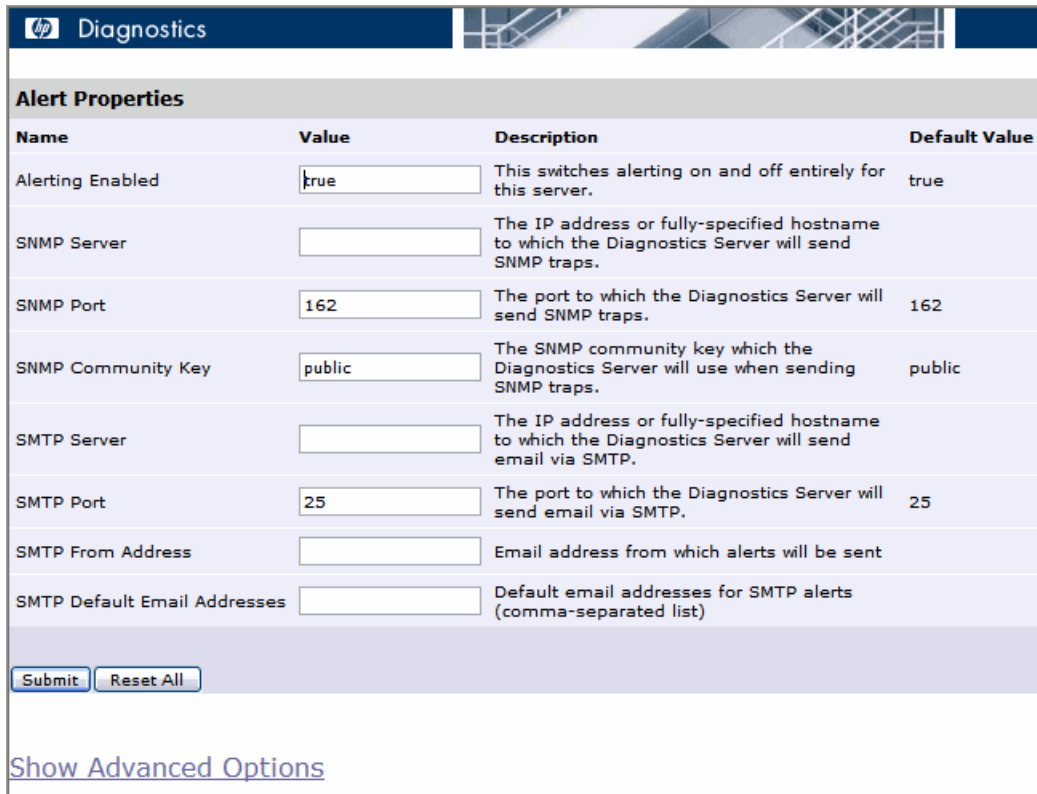
Alert Properties

Before you can set up alert notification rules, you must first set the values of the properties that enable alerting notification. You configure alert notifications from the **Diagnostics Server UI > Configure Diagnostics (Show Advanced Options) > Configuration > Alert Properties** page.

See the following topics for how to access the Alert properties page and enable or disable alert notifications and configure types of notifications:

- ▶ "How to Access the Alert Properties Page" on page 186
- ▶ "How to Enable and Disable Alert Notifications" on page 187
- ▶ "How to Configure SNMP or Email Alert Notifications" on page 188

The Alert Properties page is shown in the following example:



Name	Value	Description	Default Value
Alerting Enabled	<input type="text" value="true"/>	This switches alerting on and off entirely for this server.	true
SNMP Server	<input type="text"/>	The IP address or fully-specified hostname to which the Diagnostics Server will send SNMP traps.	
SNMP Port	<input type="text" value="162"/>	The port to which the Diagnostics Server will send SNMP traps.	162
SNMP Community Key	<input type="text" value="public"/>	The SNMP community key which the Diagnostics Server will use when sending SNMP traps.	public
SMTP Server	<input type="text"/>	The IP address or fully-specified hostname to which the Diagnostics Server will send email via SMTP.	
SMTP Port	<input type="text" value="25"/>	The port to which the Diagnostics Server will send email via SMTP.	25
SMTP From Address	<input type="text"/>	Email address from which alerts will be sent	
SMTP Default Email Addresses	<input type="text"/>	Default email addresses for SMTP alerts (comma-separated list)	

[Show Advanced Options](#)

Caution: You should always update Diagnostics alert notification properties from the Alert Properties page *for each Diagnostics Server in Mediator mode* to ensure that the property values are set correctly.

Changes to the Alert Notification configuration only apply to alert notification rules that you create after you have submitted the configuration change. To reconfigure alert rules that already exist, you must edit each alert rule.

You can turn on or off alert notification messages for all views as described below.



The **Turn On/Off Active Alert Notifications** button in the Diagnostics toolbar lets you control, for all views, whether a message is to be displayed in the Diagnostics Message box each time an external alert notification is sent.

Note: Active Alert Notifications are turned off by default and they will not be displayed unless an external alert notification has actually been sent.

About Alert Notification Rules

To receive an alert notification by SNMP or by e-mail, you must define an Alert Notification Rule for a given entity. Alert notification rules can be created for entities such as applications, probes, probe groups, hosts, and server requests. When an alert notification rule has been created for one of these Diagnostics entities, the rule is used to determine whether a notification is to be sent and how the notification is to be delivered. Alert rules can be based on the status of an entity or the status of a metric collected for an entity.

Understanding the Scope of Alert Rules

The status of lower level entities is propagated up to the higher level entities. For this reason, you do not have to set an alert on each entity where you have set a threshold. A single high-level alert notification rule on a high-level entity can be used to notify you of a variety of conditions in the performance of lower-level entities.

For example, a critical alert notification rule on a probe group is triggered any time a probe or a server request in that probe group enters critical status. This is because the status of the probe group becomes critical when any of its probes or their server requests become critical. Similarly, an alert on a probe can be triggered by the threshold violation of a probe metric; but it can also be triggered when any metric on any of the probe's server requests crosses a threshold. In these cases, the alert notification provides the details for the metric that triggered the alert, even if that metric is actually on a lower-level entity.

Maintaining Alert Notification Rules

You can create, edit, and delete alert notification rules in a number of ways:

- ▶ **Create an alert rule based on entity status.** Right-click the entity in a graph entity table or status view and select one of the alert rule options from the pop-up menu. Or you can also use the Common Tasks pane to maintain alert rules for an entity. The options displayed in the menu vary depending on whether alert rules exist or not. Only one alert notification rule can be created for each entity.

- **Create an alert rule based on a metric status.** Select the Alert Rule indicator icon next to the metric in the Details pane.

Note: You will receive two alert events if both an entity status change and a metric status change rule are set.

You cannot create an enabled alert notification rule until you have configured either the SNMP properties or the e-mail properties as described in "How to Configure SNMP or Email Alert Notifications" on page 188.

Alert Event and Alert Notification Logs

You may view the alert events for all of the Diagnostics Servers in Mediator mode in the Alert Event view. If you want to review the alert events for a single Diagnostics Server in Mediator mode, you can view the server alerting log files for the particular Diagnostics Server:

Event Log

Open the Diagnostics Server in Mediator mode administration page by navigating to the URL http://<diagnostics_server_host>:2006 in your browser, or by selecting **Start > Programs > HP Diagnostics Server > Administration** on the host for the Diagnostics Server in Mediator mode. The port number in the URL, **2006**, is the default port for the Diagnostics Server. If you configured the Diagnostics Server to use an alternative port, use that port number in the URL.

Access the Diagnostics Server Maintenance menu by selecting **Configure Diagnostics**

Select **logging > View Log Files**. Diagnostics displays a list of links to the log files that are available for viewing.

Select the link for <diagnostics_server_install_dir>/log/status_changes.log to view the log messages for all status change events for the Diagnostics Server.

Alert Notification Log

You can also view a log of the alert notifications sent on status change events. Select the link for `<diagnostics_server_install_dir>/log/alerting.log`.

Tasks

How to Access the Alert Properties Page

This task describes access the Alert properties page. You configure alert notifications using the Alert properties page.

Always update Diagnostics alert notification properties from the Alert Properties page *for each Diagnostics Server in Mediator mode* to ensure that the property values are set correctly.

To access the Alert Properties page from the Diagnostics UI:

- 1** From the menu on the top, right hand side of any of the Diagnostics UI Views, click **Maintenance**. Diagnostics displays the Diagnostics Server Maintenance menu in a new browser window.
- 2** Select **Configuration > Alert Properties**. Diagnostics displays the Alert Properties page.

Note: Changing the alert properties in this manner for a given Diagnostics Server impacts only the alert notifications for entities that are using this Diagnostics Server as a Diagnostics Server in Mediator mode.

To access the Alert Properties page from your browser:

- 1** Open the Diagnostics Server in Mediator mode administration page by navigating to the URL http://<diagnostics_server_host>:2006 in your browser, or by selecting **Start > Programs > HP Diagnostics Server > Administration** on the host for the Diagnostics Server in Mediator mode.

The port number in the URL, **2006**, is the default port for the Diagnostics Server. If you configured the Diagnostics Server to use an alternative port, use that port number in the URL.

- 2** Access the Diagnostics Server Maintenance menu by selecting **Configure Diagnostics**

- 3 Select **configuration > Alert Properties**. Diagnostics displays the Alert Properties page.

How to Enable and Disable Alert Notifications

This task describes how to enable and disable alert notifications from the Alert properties page.

Diagnostics enables alert notification by default. This means that if you have created alert notification rules and an alert triggering event has occurred, Diagnostics sends the notifications that you specified in the rules.

When alert notification is disabled, you will still see the alert conditions displayed in the Diagnostics views using the color of the status indicator and the border of the graph entity table and details pane cells that contain the metric that exceeded its threshold. Only the external alert notifications to SNMP or e-mail are disabled.

To disable alert notifications:

- 1 Access the Alert Properties page using one of the methods specified in "How to Access the Alert Properties Page" on page 186.
- 2 Set the value of the **Alerting Enabled** property to false.
- 3 Click **Submit** to disable alert notifications. It can take up to 30 seconds for the new property value to take effect.

To enable alert notifications:

- 1 Access the Alert Properties page using one of the methods specified in "How to Access the Alert Properties Page" on page 186.
- 2 Set the value of the **Alerting Enabled** property to true.
- 3 Click **Submit** to enable alert notifications. It can take up to 30 seconds for the new property values to take effect.

How to Configure SNMP or Email Alert Notifications

This task describes how to configure SNMP alert notifications or SMTP e-mail notifications.

SNMP alert notifications are disabled until you configure the SNMP alert notification properties. When you maintain an alert rule, you will not be able to select the SNMP alert notification option until you have configured these properties.

Notes:

- ▶ All SNMP alert notifications are sent as SNMP v2c traps. To help properly parse these traps on the receiving end, please refer to the `MercuryStatusAlerts.mib` file included with the server installation.
- ▶ You configure Diagnostics alert notification properties from the Alert Properties page for each Diagnostics Server in Mediator mode.

To configure SNMP alert notifications:

- 1 Access the Alert Properties page using one of the methods specified in "How to Access the Alert Properties Page" on page 186.
- 2 Set the value of the **SNMP Server**, **SNMP Port**, and **SNMP Community Key** properties to the appropriate values based on the instructions on the Alert Properties page.
- 3 Click **Submit** to enable SNMP alert notifications. It can take up to 30 seconds for the new property values to take effect.

SMTP alert notifications are disabled until you configure the SNMP alert notification properties. When you maintain an alert rule, you will not be able to select the SNMP alert notification option until you have configured these properties.

To configure SMTP e-mail alert notifications:

- 1 Access the Alert Properties page using one of the methods specified in "How to Access the Alert Properties Page" on page 186.

- 2 Set the value of the **SMTP Server**, **SMTP Port**, **SMTP From Address**, and **SMTP Default E-mail Addresses** properties to the appropriate values based on the instructions on the Alert Properties page. These values will be used to configure each alert rule that you create.
-

Note: You should set the SMTP Default E-mail Address to an alias address or e-mail distribution list that you can easily control from your e-mail server. This will save you from having to update each alert rule to a new e-mail address each time the person to receive the alerts changes.

- 3 Click **Submit** to enable SMTP alert notifications. It can take up to 30 seconds for the new property values to take effect.

How to Create or Maintain Alert Notification Rules for an Entity

This task describes how to create or maintain alert notification rules for an entity.

Note: You cannot create an enabled alert notification rule until you have configured either the SNMP properties or the e-mail properties as described in "How to Configure SNMP or Email Alert Notifications" on page 188.

To create or maintain an alert notification rule for an entity:

- 1 Right-click an entity in the graph entity table of a view with a detail layout. Diagnostics displays the pop-up menu for the selected entity.

- 2 Open the Create Alert Rule dialog box by selecting the **Create Alert Rule** menu option. If an alert rule has already been created for the selected entity, a menu option for **View/Edit Alert Rule** is displayed instead of **Create Alert Rule**. To delete an existing alert rule for an entity select the **Delete Alert Rule** menu option and select Yes in the Delete Alert dialog box.

How to Create or Maintain Alert Notification Rule on a Metric

This task describes how to create or maintain an alert notification rule on a metric.

Note: You cannot create an enabled alert notification rule until you have configured either the SNMP properties or the e-mail properties as described in "How to Configure SNMP or Email Alert Notifications" on page 188.

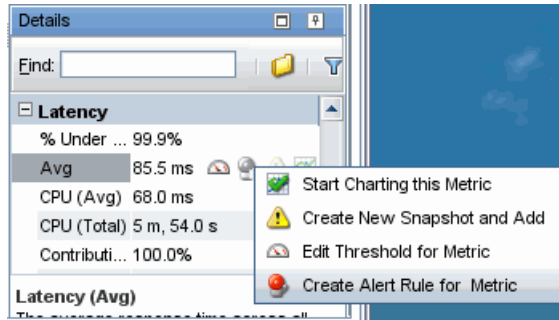
This task includes the following steps:

- "Select the Alert Rule Icon in the Details Pane" on page 190
- "The Create/Edit Alert Rule Dialog Box is Displayed" on page 191
- "Enter a Name and Description" on page 192
- "Select One or More Alert Triggers" on page 193
- "Select One or More Alert Notification Options" on page 193

1 Select the Alert Rule Icon in the Details Pane



In the Details pane select the Alert Rule indicator icon next to the metric you want to use in an alert rule. To edit or delete an alert rule for the metric, right-click the Alert Rule indicator icon and select **Edit Alert Rule for Metric** or **Delete Alert Rule for Metric**.

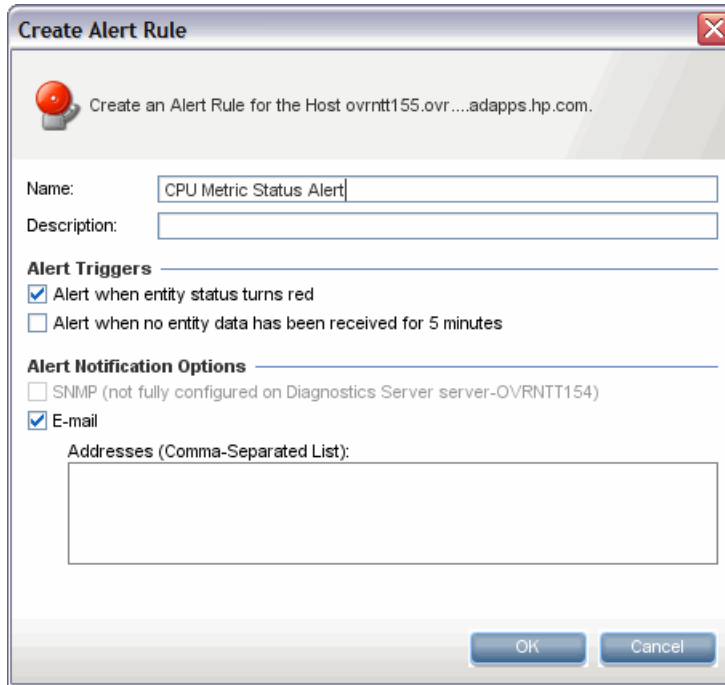
Example:**2 The Create/Edit Alert Rule Dialog Box is Displayed**

The Create Alert Rule dialog box is displayed allowing you to create a new alert rule based on this metric. Only those metrics that can be used in an alert rule will show the Alert Rule indicator icon. If an alert rule has already been created for the metric the Alert Rule indicator icon will be highlighted (red, Active Alert Rule).

Note: For entity status alert rules you can select multiple entities in a graph entity table and then select the menu item to create, replace or delete alert rules. Typically, when selecting multiple entities you should select groups of entities that you want to have the same alert rules.

Example:

The Create Alert Rule dialog box as shown in the following example:



3 Enter a Name and Description

Enter a **Name**. The name is either for the entity (to help you to remember the entity and the reason why you wanted to receive alert notifications) or the metric name.

For example, if the alert rule was for a host named **Prod_010** that had been experiencing CPU usage spikes, you may want to enter a name such as **Prod_010 CPU Check**. However, you may want to use a more generic name for the alert rule if there is more than one metric with a threshold value that could trigger a single alert notification. A specific name that includes the name of the metric could be misleading in this case.

Enter a short **Description** for the rule.

4 Select One or More Alert Triggers

Select one or more **Alert Triggers** that will cause notifications to be sent.

If you select **Alert when entity status turns red**, the notifications are sent whenever the status of the entity becomes critical and the indicator is turned red.

An entity's status becomes critical whenever one of its metric thresholds has been exceeded, or when any of the metric thresholds have been exceeded on its sub-entities.

If you select **Alert when no entity data has been received for 5 minutes**, the notifications are sent whenever the Diagnostics Server has received no data from the entity for 5 minutes.

If you do not select at least one Alert Trigger, the alert notification rule is created but is disabled. So to **disable an alert rule**, de-select both of these checkboxes.

5 Select One or More Alert Notification Options

Select one or more **Alert Notification Options** that determine how the alert notifications are to be delivered.

If you select **SNMP**, the notifications are sent via SNMP traps (v2c). Note that the server installation includes a MIB file to help parse these traps.

If you select **E-mail**, the notifications are sent via e-mail.

If you do not select at least one Alert Notification Option, the alert notification rule is created but is disabled.

Reference

Alert Notifications User Interface













This section includes:


- ▶ Alert Rules View Description on page 194
- ▶ Alert Events View Description on page 197

Alert Rules View Description

Diagnostics provides a view that allows you to review and maintain all of the alert notification rules that you have created in one convenient and powerful view. The **Alert Rules** view displays a list of all of the alert notification rules sorted by default in alphabetical order by the entity name.




The following is an example of the Alert Rules view.

Alert Rules						
Entity Name	Alert On	Alert Name	Type	Last Fired	Info	
L81_WAS60_Portal_ovrib...	Probe VM Heap Free (Avg) metric status	VM Heap Free Metric Status Alert	Metric			
L81_WAS60_Portal_ovrib...	Probe EJB Creates (Avg) metric status	EJB Creates Metric Status Alert	Metric			
L81_WAS60_Portal_ovrib...	Probe status	L81_WAS60_Portal_ovribmt10_AIX...	Entity	09/30/09 16:19:04....		
L91_JavaTrader2.WebClic...	Probe Latency (Avg) metric status	Latency Metric Status Alert	Metric			
L91_WAS_ESB_60_HPS...	Probe Availability (Avg) metric status	Availability Metric Status Alert	Metric			
ovrhp04.rose.hp.com	Host CPU (Avg) metric status	CPU Metric Status Alert	Metric			

<p>To access</p> 	<p>In the View bar, click Alerting to open the Alerting group.</p> <p>Select Alert Rules from the Alerting group.</p>
<p>Important information</p>	<p>Alert Rules cannot be filtered by application so in an application context you will not see the Alert Rules selection in the view bar under Alerting.</p>
<p>Relevant tasks</p>	<ul style="list-style-type: none"> ➤ "How to Create or Maintain Alert Notification Rules for an Entity" on page 189 ➤ "How to Create or Maintain Alert Notification Rule on a Metric" on page 190
<p>See also</p>	<ul style="list-style-type: none"> ➤ "How to Access the Alert Properties Page" on page 186 ➤ "How to Enable and Disable Alert Notifications" on page 187 ➤ "How to Configure SNMP or Email Alert Notifications" on page 188 ➤ "About Metric Thresholds" on page 158

The following user interface elements are included:










UI Elements	Description
<Table>	<p>The Alert Rules table displays a list of all of the alert notification rules sorted by default in alphabetical order by the entity name. From the list, you can select right-click menu options to view, edit, delete the alert notification rules and navigate to the Diagnostics views that are associated with the entities that have alert notification rules. You can select multiple rows in the table to delete multiple alert rules.</p> <p>The columns in the Alert Rules view provide the same information and navigation options as their counterparts in the graph entity table of the views with a detail layout.</p> <p>When you right-click the rows in the Alert Rules view, Diagnostics displays the same pop-up menu that appears for the listed entity in the views with a detail layout.</p> <p>The columns are:</p> <p>Entity Name. The name of the entity listed in the view. When the alert rule is on a metric, this column shows the parent entity name. When you hold your mouse pointer over the values in the column the tooltip that Diagnostics displays is the same tooltip displayed in the detail layout view for the entity.</p> <p>Alert On. What status change the alert rule is based on. For example, Probe status, Host status or Probe VM Heap Used (Avg) metric status.</p> <p>Alert Name. The name of the alert notification rule.</p> <p>Type. The type, either entity or metric, listed in the row.</p> <p>Last Fired. The timestamp for the last time the listed alert notification fired. If an alert notification has been issued for a listed alert rule the pop-up menu for the entity contains the View Threshold Violation option.</p> <p>The Last Fired column contains the last time that the alert event fired recently. There is a limited number of alert notification events that are cached in the server.</p>


UI Elements	Description
<p><Table></p>   	<p>Info. This column indicates that the entity has alert rules or comments. One or more of the following icons can appear in this column:</p> <ul style="list-style-type: none"> ▶ Active Alert Rule. Indicates that an alert notification rule has been created for the selected entity and the rule is active. ▶ Disabled Alert Rule. Indicates that an alert notification rule has been created for the selected entity but the rule is currently disabled. (You can disable an alert rule in the Alert Rule dialog box by de-selecting both checkboxes for Alert Triggers.) ▶ Custom Comments. Indicates that comments have been created for the selected entity.

Alert Events View Description

You can review the events that triggered alert notifications in the log files and in the Alert Events view. The information in the alert notification event can help you know where to begin looking for the cause of the performance problem that triggered the alert.

The following is an example of the Alert Events view.

Alert Events								
Timestamp	Entity Name	Violating Metric	Threshold	Value	Alert On	Alert Name	Type	Status
09/30/09 16:19:04	L81_WAS60_...	SQL Latency (Avg)	100 ms	552.5 ms	Probe status	L81_WAS60...	Entity	
09/30/09 16:02:12	/PlantsByWeb...	Server Request L...	2 m	3 m, 8 s	Server Requ...	Latency Met...	Metric	
09/30/09 15:41:38	L81_WAS60_...	SQL Latency (Avg)	100 ms	505 ms	Probe status	L81_WAS60...	Entity	
09/30/09 15:32:46	ovrntt158.ovrt...	Host CPU (Avg)	25.7 %	25.8 %	Host status	ovrntt158.ov...	Entity	
09/30/09 15:31:06	/PlantsByWeb...	Server Request L...	2 m	2 m, 9 s	Server Requ...	Latency Met...	Metric	
09/30/09 15:10:55	L81_WAS60_...	SQL Latency (Avg)	100 ms	106.2 ms	Probe status	L81_WAS60...	Entity	
09/30/09 14:41:02	ovrntt158.ovrt...	Host CPU (Avg)	25.7 %	25.8 %	Host status	ovrntt158.ov...	Entity	
09/30/09 14:40:31	L81_WAS60_...	SQL Latency (Avg)	100 ms	219 ms	Probe status	L81_WAS60...	Entity	
09/30/09 14:29:39	/PlantsByWeb...	Server Request L...	2 m	2 m, 45 s	Server Requ...	Latency Met...	Metric	

<p>To access</p> 	<p>In the View bar, click Alerting to open the Alerting group.</p> <p>Select Alert Events from the Alerting group.</p>
<p>Important information</p>	<p>If an alert notification rule has fired, you can double-click the row representing the alert event to be taken to a view that displays the threshold violation.</p> <p>Or you can right-click a row in the Alert Events table and select View Threshold Violation.</p> <p>Diagnostics displays the detail layout view that is appropriate for the entity type depicted in the selected row. The detail layout view contains the selected entity in the graph entity table, and the graph contains the charted metrics that triggered the alert. The threshold violation is shown with a red plus sign icon.</p>
<p>Relevant tasks</p>	<ul style="list-style-type: none"> ▶ "How to Enable and Disable Alert Notifications" on page 187 ▶ "How to Configure SNMP or Email Alert Notifications" on page 188
<p>See also</p>	<ul style="list-style-type: none"> ▶ "About Metric Thresholds" on page 158

The following user interface elements are included:

UI Elements	Description
<Table>	<p>Diagnostics lists each of the events that triggered an alert in the Alert Events view. By default, the table of events in this view is sorted with the most recent alert events at the top. This can be especially helpful when the Alert Events view is included in a custom dashboard view.</p> <p>The columns are:</p> <p>Timestamp. The time at which the alert event was triggered.</p> <p>Entity Name. The name of the entity listed in the view. When the alert event is on a metric, this column shows the parent entity name. When you hold your mouse pointer over the values in the column, the tooltip that Diagnostics displays is the same tooltip displayed in the detail layout view for the entity.</p> <p>Violating Metric. This indicates the entity type and metric that caused the event.</p> <p>Threshold. The threshold value for the metric that triggered the alert event. Exceeding the threshold value is one of the possible triggers for an alert event.</p> <p>Value. The value for the metric when the alert was triggered.</p> <p>Alert On. The entity status or metric status change the alert rule is based on. For example, Probe status or Probe VM Heap Used (Avg) metric status.</p> <p>Alert Name. The name of the alert notification rule that was triggered by the alert event.</p> <p>Type. The type, either entity or metric, listed in the row.</p> <p>Status. This indicator shows how the listed entity or any of its child entities is performing relative to the thresholds that you set for their metrics. See "Status Indicators" on page 517 for details on how Status is calculated.</p>

6

Performing Snapshot Analysis

This chapter describes how to use the snapshot analysis feature of Diagnostics.

This chapter includes:

Concepts

- ▶ About Snapshot Analysis on page 202
- ▶ Monitoring Performance Versus Analyzing Snapshots on page 203

Tasks

- ▶ How to Create and Manage Snapshots on page 205
- ▶ How to Add and Delete Metrics or Entities from a Snapshot on page 207

Reference

- ▶ Analyze Snapshots User Interface on page 210

Concepts

About Snapshot Analysis

When you see a performance anomaly, you can use Diagnostics to capture the performance of your application, and then store the information as a snapshot. A snapshot includes all of the performance metrics that you can see in the Diagnostics views, but it is limited to just the time period for which the snapshot was captured. With the information captured in a snapshot, you can investigate threshold violations and other performance issues, using the Analyze Snapshots view and the other Diagnostics views. Snapshots provide you with the metrics for the specific slice of time in which the performance of your application was not as you expected. These metrics help you understand what is causing the performance issue.

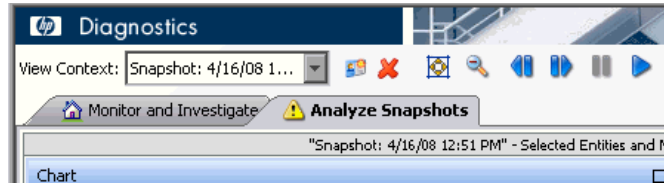
You control when Diagnostics captures a snapshot. When you see a performance anomaly that you want to analyze, you can use the controls in the Diagnostics views to create a snapshot that includes the performance metrics for the specified time period.

Data for the time range of a snapshot are preserved. They are exempt from routine data purging. This time-range preservation enables you to save a snapshot that shows the symptoms of a problem. Later, you can use the snapshot to help troubleshoot similar problems. When you delete a snapshot, the time range is removed from the global list of preserved times. The data is purged.

If you have multiple snapshots in the same time range (or overlapping time ranges), the data is preserved in the overall time range, independent of snapshots. This snapshot-independent preservation enables you to delete a snapshot without removing data that is a part of another snapshot.

Monitoring Performance Versus Analyzing Snapshots

The Monitor and Investigate tab and the Analyze Snapshots tab behave differently, depending on your selection from the **View Context** drop-down menu.



When you select **Monitoring** from the **View Context** drop-down menu, Diagnostics is in Monitoring mode. The views displayed in the Monitor and Investigate tab contain the real-time performance metrics for the time period specified in the **Viewing Time** filter.

When you select a snapshot from the **View Context** drop-down menu, Diagnostics is in Snapshot Analysis mode. As soon as you select a snapshot, Diagnostics opens the **Analyze Snapshots** view in the Analyze Snapshots tab for the selected snapshot. The view displays the entities and metrics that you specified shown in the tables and graph.

You can modify the snapshot to control the entities and metrics that are included in the Analyze Snapshots view. You can then visually compare and correlate the metrics from multiple entities of different types.

For details, see "How to Add and Delete Metrics or Entities from a Snapshot" on page 207.

In general, the controls in Snapshot Analysis mode work the same as the controls in Monitoring mode. However, you should be aware of the following differences in Snapshot Analysis mode:

- The **Viewing Time** filter contains an additional option, **Use Time of Snapshot**. This option enables you to return to the snapshot time after you have changed the viewing time by using the **Viewing Time** filter, by using the **Pan**, **Pause**, and **Play** buttons, or by zooming.

- ▶ When you are in Snapshot Analysis mode, you can use the **Pan**, **Pause**, and **Play** buttons to display performance metrics for periods of time that precede or follow the metrics that are charted in the graph. When you click **Play**, Diagnostics stays in Snapshot Analysis mode. However, it begins to show the current values for the performance metrics charted in the graph and listed in the tables of the views. You can return to the time for the captured snapshot by clicking the **Use Time of Snapshot...** option in the **Viewing Time** filter.
- ▶ The Status view and the Alert Events view always show the most up-to-date information for the performance of your applications, no matter if you are in Snapshot Analysis mode or in Monitoring mode. These views are not impacted by your choices in the **Viewing Time** filter and the **View Context** drop-down list.

Tasks

How to Create and Manage Snapshots

This task describes how to create and maintain a snapshot capture of the data associated with a performance issue for evaluation in the Analyze Snapshots view.


You can create a snapshot capture using the controls of the graph entity table and the details pane. When Diagnostics creates a new snapshot, it creates a default snapshot name, based on the starting date and time of the period for which the snapshot was captured. Diagnostics also sets up the Analyze Snapshot view to display the entities and metrics that are appropriate for the trigger that caused the snapshot to be captured.

To create a new snapshot from the graph entity table:

- 1 From the graph entity table, select the row for the entity that you want to include in the Analyze Snapshots view of the new snapshot. Note that you can select multiple rows in the graph entity table to add to the snapshot.
- 2 Select the **Create New Snapshot and Add** menu option either from the right-click pop-up menu for the entity, or from the Common Tasks menu.

Diagnostics creates a new snapshot for the time period displayed in the view. The new snapshot is selected automatically from the **View Context** drop-down menu as Diagnostics switches to Snapshot Analysis mode. The Analyze Snapshots view opens with the selected entity listed in the graph entity-metric table, and with the same metric charted in the graph as was charted on the view from which you captured the snapshot.

To capture a new snapshot from the details pane:

- 1 From the details pane, select the row for the metric that you want to include in the Analyze Snapshots view of the new snapshot.
- 2  Select the **Create New Snapshot and Add** menu option from the right-click pop-up menu for the metric.

Diagnostics creates a new snapshot for the time period displayed in the view. The new snapshot is automatically selected from the **View Context** drop-down menu as Diagnostics switches to Snapshot Analysis mode. The Analyze Snapshots view opens with the metric you selected charted in the graph, and with the entity to which the metric belongs listed in the graph entity-metric table.

You can rename a snapshot listed in the **View Context** drop-down list to make it easier to locate in the drop-down list, or to make it easier to recognize the performance issue that the snapshot represents.

To rename a snapshot:

- 1 Select the snapshot to be renamed from the **View Context** drop-down list.

Diagnostics switches to Snapshot Analysis mode, and displays the metrics for the selected snapshot in the tab that was open when you made your selection.



- 2 Click **Rename Snapshot** to the right of the **View Context** drop-down list.

Diagnostics displays the Rename Snapshot dialog box with the current name of the selected snapshot.

- 3 Enter the new snapshot name.
- 4 Click **OK** to rename the snapshot.

You can delete a snapshot listed in the **View Context** drop-down list so a snapshot that you have already resolved, or a snapshot in which you are no longer interested, does not clutter the list in the drop-down list.

To delete a snapshot:

- 1 Select the snapshot to be deleted from the **View Context** drop-down list.

Diagnostics switches to Snapshot Analysis mode, and displays the metrics for the selected snapshot in the tab that was open when you made your selection.

- 2 Click **Delete Snapshot** to the right of the **View Context** drop-down list.

Diagnostics deletes the selected snapshot, so it is not longer listed in the **View Context** drop-down list.

How to Add and Delete Metrics or Entities from a Snapshot

This task describes how to add additional metrics or entities to an existing snapshot and how to delete a metric or entity from a snapshot.

As you analyze a snapshot, you discover entities and metrics that help explain the application's behavior. You want to include this information in the Analyze Snapshots view. You also notice that some of the entities and metrics shown in the view are no longer helpful, and should no longer be included in the view. Diagnostics provides a number of ways for you to manipulate the entities and metrics that are included in the Analyze Snapshots view.

You can choose additional metrics to be charted in the graph of the Snapshot Analysis view, either from the views in the Monitoring and Investigate tab, or from the Analyze Snapshots view.

To add a metric to the Analyze Snapshots view:

- 1** Select a snapshot from the **View Context** drop-down menu so Diagnostics is in Snapshot Analysis mode.
- 2** From the details pane in either the Analyze Snapshots view, or in a view displayed in the Monitor and Investigate tab, select the row for the metric that you want to include in the active snapshot.
- 3** Select the **Add to Active Snapshot** menu option from the right-click pop-up menu for the metric, or click the icon for the metric.



Diagnostics displays the Analyze Snapshots view. The selected metric is charted in the graph, and an entry for the selected entity-metric pair is listed in the details pane. In the details pane, the icon for the metric is changed to indicate that the metric is included in the view.

When Diagnostics is in Snapshot Analysis mode, you can choose additional entities to be included in the Snapshot Analysis view. When you include an entity from a Diagnostics view, each of the metrics for the entity that were charted are also charted in the graph of the Snapshot Analysis view. You can select an entity to add to the Snapshot Analysis view from the views displayed on the Monitoring and Investigate tab.

To add an entity to the Analyze Snapshots view:

- 1** Select a snapshot from the **View Context** drop-down menu so Diagnostics is in Snapshot Analysis mode.
- 2** In the Monitor and Investigate tab, navigate to the view that contains the target entity.
- 3** From the graph entity table, select the row for the entity that you want to include in the Analyze Snapshots view for the active snapshot.
- 4** Select the **Add to Active Snapshot** menu option, either from the right-click pop-up menu for the entity, or from the Common Tasks pane.

Diagnostics displays the Analyze Snapshots view. The selected entity is included in the view, so each of the metrics that were charted for the entity in the Monitor and Investigate tab are also charted in the graph, and an entry for each entity-metric pair is listed in the graph entity table. In the details pane, the icon for each of the charted metrics is changed to indicate that the metric is included in the view.



When Diagnostics is in Snapshot Analysis mode, you can remove metrics from the Snapshot Analysis view so they are no longer charted in the graph, or listed as an entity-metric pair in the entity-metric table. You can delete a metric from the views in the Monitoring and Investigate tab or from the Analyze Snapshots view.

Note: When you delete a metric from a view, the entity is no longer included in the view after you delete the last metric for the entity.

To delete a metric from the Analyze Snapshots view:

- 1** Select a snapshot from the **View Context** drop-down menu so Diagnostics is in Snapshot Analysis mode.
- 2** From the details pane either in the Analyze Snapshots view, or in a view displayed in the Monitor and Investigate tab, select the row for the metric that you want to remove from the Analyze Snapshots view of the active snapshot.



- 3 Click the icon for the metric on the selected row, or select the **Remove from Active Snapshot** menu option from the right-click pop-up menu for the metric.

Diagnostics continues to display the view with which you were working. In the Analyze Snapshots view, the selected metric is no longer charted in the graph, and the entry for the selected entity-metric pair is deleted from the entity-metric table. In the details pane, the icon for the metric is changed to indicate that the metric is no longer in the view.



When Diagnostics is in Snapshot Analysis mode, you can remove entities so they are no longer included in the Snapshot Analysis view. When you remove an entity from a Diagnostics view, each of the metrics for the entity that were charted are also removed from the Snapshot Analysis view. You can select an entity to remove from the Snapshot Analysis view from the views displayed on the Monitoring and Investigate tab, and from the Snapshot Analysis view.

To delete an entity from the Analyze Snapshots view:

- 1 Select a snapshot from the **View Context** drop-down menu so Diagnostics is in Snapshot Analysis mode.
- 2 In the Monitor and Investigate tab, navigate to the view that contains the target entity. You do not need to do this if you are in the Analyze Snapshots view.
- 3 From the graph entity table, select the row for the entity that you want to remove from the Analyze Snapshots view for the active snapshot.
- 4 Select the **Remove From Active Snapshot** menu option, either from the right-click pop-up menu for the entity, or from the Common Tasks pane.

Diagnostics displays the Analyze Snapshots view. The selected entity is removed from the view along with each of the metrics that were charted for the view.

Reference

Analyze Snapshots User Interface

This section includes:

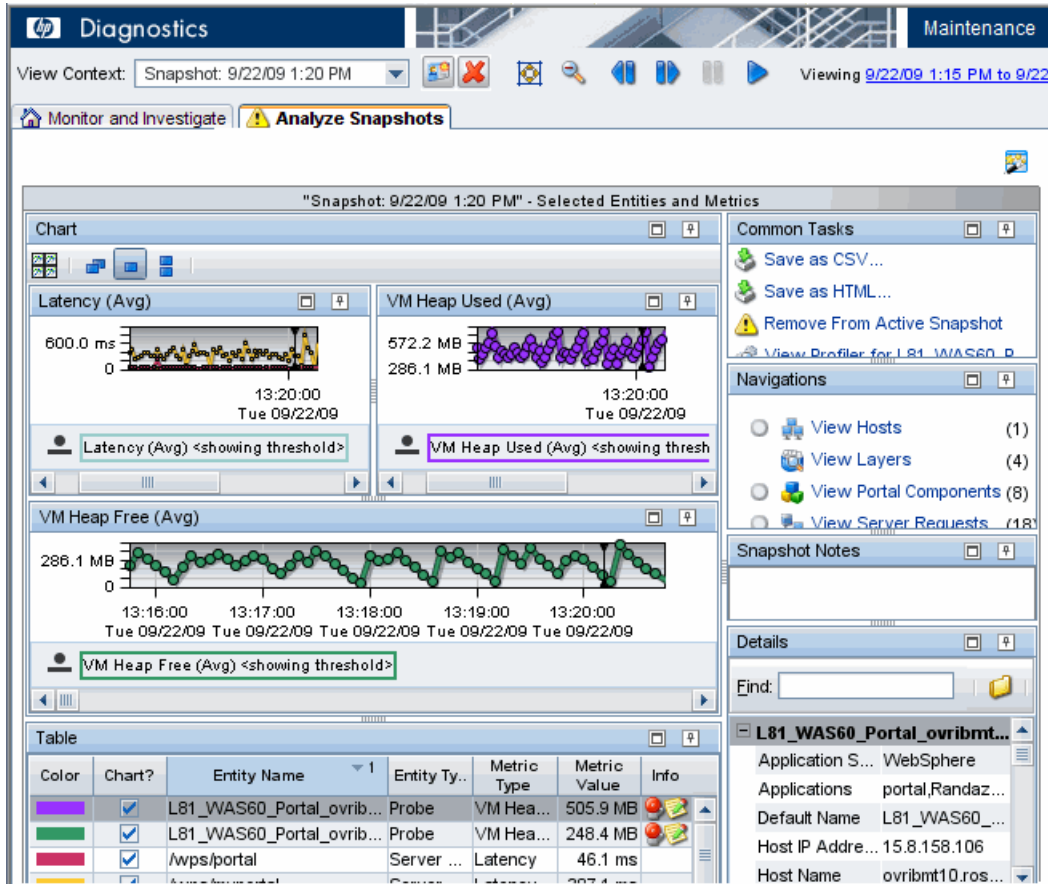
- ▶ Analyze Snapshots View Description on page 210

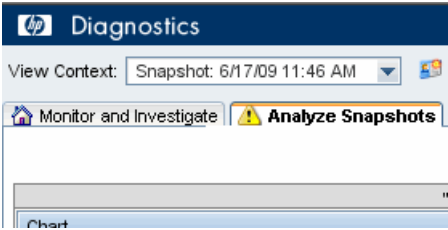
Analyze Snapshots View Description

The Analyze Snapshot View on the Analyze Snapshot tab displays the entities and metrics from a captured snapshot that you have selected to be included in this view. This view provides a way for you to compare, contrast, and correlate entities and metrics of different types that would normally only be shown in separate views in the graph and graph entity table of one view.

Note: The Analyze Snapshots view is displayed in the Analyze Snapshots tab only when Diagnostics is in Snapshot Analysis mode. When Diagnostics is in Monitoring mode, the Analyze Snapshots tab contains instructions for creating a new snapshot.

The following is an example of the Analyze Snapshot view.



<p>To access</p>	<p>To access the Analyze Snapshots view, Diagnostics must be in Snapshot Analysis mode. Diagnostics is in Snapshot Analysis mode when you select a snapshot from the View Context drop-down menu, or when you create a new snapshot from a view while in Monitoring mode.</p> <p>The Analyze Snapshots view displays on the Analyze Snapshots tab when Diagnostics is in Snapshot Analysis mode.</p>  <p>If you open the Analyze Snapshots tab when Diagnostics is in Monitoring mode, the Analyze Snapshots view is not displayed. Instead, a text message appears with instructions for creating a new snapshot.</p>
<p>Important information</p>	<ul style="list-style-type: none"> ▶ If you were looking at the views in the Monitor and Investigate tab, Diagnostics continues to display the same view in that tab. However, the time range displayed in the tab changes to that of the selected snapshot. You can click the Analyze Snapshots tab at any time to see the Analyze Snapshots view for the selected snapshot. ▶ If you were looking at the Analyze Snapshots view, Diagnostics continues to display the same view. However, the performance metrics displayed in the tab change to those that were selected for the view for the selected snapshot. And the time range changes to that of the selected snapshot.

Relevant tasks	<ul style="list-style-type: none">➤ "How to Create and Manage Snapshots" on page 205➤ "How to Add and Delete Metrics or Entities from a Snapshot" on page 207
See also	For more information on creating a new snapshot see "Monitoring Performance Versus Analyzing Snapshots" on page 203.

The following user interface elements are included:

UI Elements	Description
Chart	By default, the charts (graph) in the Analyze Snapshots view contain the charted metrics for the entity-metric pairs listed in the graph entity table. These metrics are the metrics that you have included in the snapshot.
Table	The graph entity table lists each entity-metric pair that you selected to be included in the Analyze Snapshots view. Each entity included in the view is listed once for each metric that has also been included in the view.
Common Tasks and Navigations	Common tasks and actions relevant to the entities displayed in this view.
Snapshot Notes	<p>The Snapshot Notes area enables you to record the reason you created the snapshot, the steps you took, and the progress you made in understanding the cause of the performance problem captured in the snapshot.</p> <p>Any text that you enter here is saved when you leave the view. The text is redisplayed the next time you work with the snapshot in the Analyze Snapshots view.</p>
Details Pane	<p>The details pane in the Analyze Snapshots view lists the metrics for the selected row of the graph entity table.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

7

Working with Custom Views

This chapter describes how to work with custom views in Diagnostics.

This chapter includes:

Concepts

- ▶ About Custom Views on page 216
- ▶ Retention of Customization on page 216
- ▶ Share Custom Views on page 217
- ▶ Upgrade Custom Views from Previous Diagnostics Version on page 219
- ▶ Custom Views for Use with Different Instances (Open in Custom View) on page 219

Tasks

- ▶ How to Create a Custom View Group on page 221
- ▶ How to Hide or Open a Custom View Group on page 222
- ▶ How to Create a Custom View on page 223
- ▶ How to Save a Custom View on page 227
- ▶ How to Rename a Custom View on page 229
- ▶ How to Delete a Custom View on page 230

Reference

- ▶ My Views User Interface on page 231

Concepts

About Custom Views

Diagnostics lets you create your own custom views and save the changes so that you can reuse the custom view when you use Diagnostics in the future. The controls in the View bar enable you to save, retrieve, and revise your customized views. Instructions for using the View bar controls to maintain your custom views are provided in the following sections.

The **My Views** custom view group (or **Shared Views** in the application context) is installed with the product. Out-of-the-Box, two custom views are provided in the My Views view group (these are read-only view):

- ▶ **Probe Details** - shows key probe metrics, hosts CPU usage and top server requests by latency.
- ▶ **Java Probes Memory Usage** - shows key memory metrics for Java probes including data on collection leaks.

If you accessed the Diagnostics UI as the *Mercury System* customer you will see **Server - Basic Metrics** as the Out-of-the-Box custom view. The view shows key metrics for our internal monitors systems.

When you create or edit custom views they must be saved in a custom view group, such as My Views, or in a custom view group that you create.

Retention of Customization

When you make modifications to the way information is displayed in a Diagnostics view, Diagnostics retains the customizations in different ways depending on whether the view was a custom view or a standard view and depending on how you navigated to the view.

Customization of Default Views

When you customize a view that you selected from a view group that was installed with the product, the changes are retained and displayed each time you access the view as long as you do not shut down Diagnostics completely. Once you leave Diagnostics or exit back to the Application Overview screen, the view is displayed in its default configuration the next time you access it.

The out-of-the-box custom views Probe Details and Java Probe Memory Usage are read only views. To save changes you should use the save as function to create a new view that can be changed and persisted.

Customization of Drill Down Views

When you customize a view that you accessed by drilling down from another view, your customizations are retained only as long as you navigate within the breadcrumbs. If you drill down on the same path again, the customizations will be retained as long as you did not navigate to another path in the meantime.

Customization of Custom Views

When you customize a view that you selected from a view group that you created, the customizations are retained when you access the view again, regardless of whether the Diagnostics has been shut down. Customizations to your custom views are automatically saved.

Share Custom Views

Diagnostics provides two ways for you to share the custom views that you have created, with other users.

- ▶ When you create a custom view and you are not in the context of an application (Entire Enterprise views selected in the Applications window), the view is stored on the Diagnostics Server in Commander mode in the directory `<diagnostics_server_install_dir>/storage/userdata/<customer name>/<user name>`. Views are stored as XML files, and are named based on the view group and the view name. For example, the **Employee Benefits Overview** view, in the **Employee Application** view group, is stored as **Employee Application - Employee Benefits Overview.xml**.

To share a view with another user, copy the XML file to that user's directory.

Note: The two views are completely independent, and changes to one will not be reflected in the other.

- ▶ When you create a custom view in the context of an application, the XML file for that view is stored in `<diagnostics_server_install_dir>/storage/userdata/<customer name>/<application identifier>`.

In the Diagnostics configuration UI use the query page Advanced Options and enter the following path to execute this query:

```
/groupby[name='Default Client']/index[equals(name,'apps')]/app
```

to find the application name based on application identifier shown in the subpath column.

Any user who logs into Diagnostics and has permission to view that application will share this view. Note, however, that with the new application permissions, only users with the Modify Screens permission for the application will actually have their changes saved.

This does, however, present a potential for multiple users to be viewing the same custom view and make changes. In this case, only the first user's changes will be saved. Other users (or even the same person logged in multiple times) will see an error message that their changes could not be saved, and they should save the view as a new view if they want to preserve the changes.

Upgrade Custom Views from Previous Diagnostics Version

When you open the custom views that were created in earlier versions of Diagnostics for the first time in the latest version of Diagnostics, Diagnostics upgrades the view for any changes that are necessary because of changes to the functionality of Diagnostics. When Diagnostics changes your custom views it issues a message to let you know that your custom view has been modified.

For instructions on upgrading Diagnostics data refer to the *HP Diagnostics Installation and Configuration Guide* appendix on upgrades.

For instructions on using individual custom views from a backup or from another user see "Share Custom Views" on page 217.

Custom Views for Use with Different Instances (Open in Custom View)

You can create custom views that can be used with different instances of an entity. From a standard view you can select an entity (such as a probe) and choose **Open in Custom View** from the pop-up menu for the entity. The custom view you select is displayed in the context of the entity you selected.

Only those custom views that apply to the type of entity selected are shown as available in the Open in Custom View dialog box.

Out-of-the-Box two custom views are provided in the My Views group accessed from the View bar:

- ▶ Probe Details
- ▶ Java Probes Memory Usage

Both these views can be opened (Open in Custom View) within the context of a selected probe.

If you accessed the Diagnostics UI as the *Mercury System* customer you will see **Server - Basic Metrics** as the Out-of-the-Box custom view. The view shows key metrics for our internal monitors systems.

Tasks

How to Create a Custom View Group

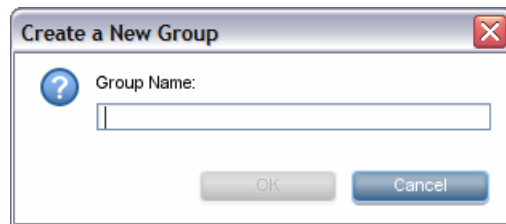
Custom views must be saved in a custom view group, such as **My Views**, which is installed with the product, or in a custom view group that you create. Custom views cannot be stored in any of the view groups installed with the product except for **My Views**.

Note: Custom view groups created within the context of an application can only be viewed within that context. And these custom view groups are shared across all users of that application.

To create a view group:

- 1 Right-click the View bar inside any view group, and select **Create a View Group** from the pop-up menu. Make sure to hold the pointer over an open space on the View bar, and not over the icon for a view.

The Create a View Group dialog box opens.



- 2 Type a name for the view group, and click **OK**.

Diagnostics creates the new view group and opens it in the View bar.

How to Hide or Open a Custom View Group

If there are View Groups listed in the View bar that you do not wish to see in the list any longer you may close them so that Diagnostics will not display them in the View bar. You may open View groups that have been hidden so that they are once more available in the View bar.

To close a view group:

- 1 Select the View group that you want to hide from the View bar.

Note: In custom view groups, you may only hide groups that do not contain any views. To hide a custom view group with views, first delete the views.

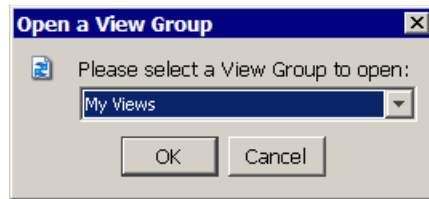
- 2 Right-click the View bar inside of the selected view group, and select **Close this View Group** from the pop-up menu. Make sure to hold the pointer over an open space on the View bar, and not over the icon for a view.

Diagnostics hides the view group so that it is no longer visible in the view group displayed in the View bar and opens the View group that preceded the view group that you hid.

To open a view group:

- 1 Right-click the View bar inside of any view group, and select **Open a View Group** from the pop-up menu. Make sure to hold the pointer over an open space on the View bar, and not over the icon for a view.

Diagnostics displays the Open a View Group dialog box as shown in the following image:



- 2 Select the view group that you want to open and click **OK**

Diagnostics unhides the selected view group and displays it at the end of the list of view groups in the view bar.

How to Create a Custom View

You may assemble various combinations of one or more existing views to create a completely new view. By creating a new view, you can put the standard views and custom views that you use regularly together in one view, allowing you to see the performance metrics side-by-side and to navigate to the underlying metrics just as you can in any other view in Diagnostics.

Note: Custom views created within the context of an application can only be viewed within that context. And these custom views are shared across all users of that application.

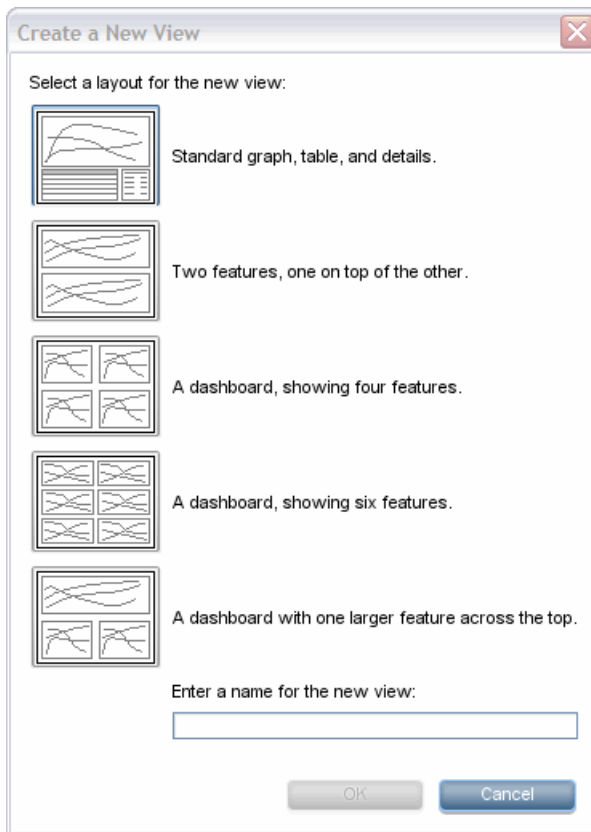
To create a view:

- 1 Select the custom view group where you want to store the new view from the view bar.

Note: Custom views can only be stored in the My Views view group, or in a view group that you have created.

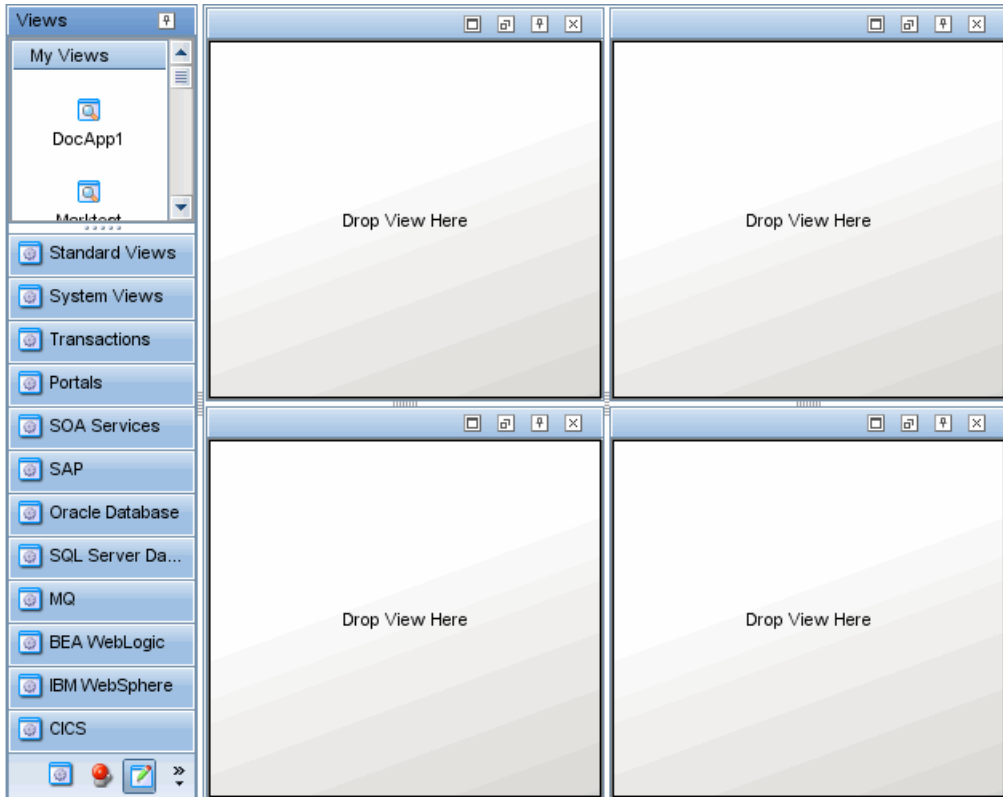
- 2 Right-click the View bar inside the custom view group, and select **Create a New View** from the pop-up menu.

Diagnostics opens the **Create a New View** dialog box.



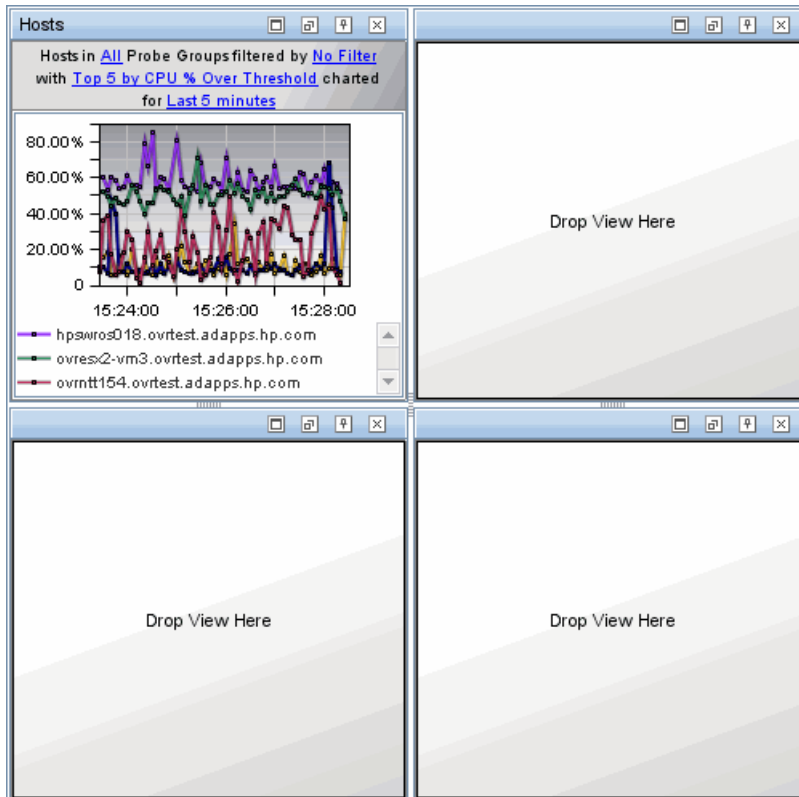
- 3 Select the icon for the layout of the new view. The selected icon is highlighted. Enter a name for the new view, and click **OK**.

An icon for the new view is added to the view group, and the selected view layout is displayed on your screen with **Drop View Here** placeholders for each of the views that make up the new view. The example below shows the view layout for the four-feature dashboard.



- 4 Select the views that you want to appear in the new custom view.
 - To add a view, select it from the view groups in the View bar, and drag it into the **Drop View Here** placeholders.
 - To remove a view, click the **Remove From View** button in the upper right corner of the view, as shown below.

The view is deleted, and the empty **Drop View Here** placeholder is displayed once more.



The changes that you make to the new view are automatically saved as you make them.

Adding any summary view to a new custom summary screen will add each of the summaries' features to the new custom summary screen. For example if you add the Server Summary view to a new four way custom view then it will add all three of the Server Summary features. You would then have one remaining area of the four for another view.

Adding a summary view, for instance the Server Summary view, to a new two way custom view will result in a view with more than two features. Since the Server Summary view has three features each feature is added.

How to Save a Custom View

As you work with a view, you may find that your customization to the view has been particularly useful in helping you to understand an aspect of your application's performance.

You can save the customized view exactly as it is displayed into the My Views view group or into a view group that you have created.

You can save customized versions of views that you opened from the Standard Views group, and you can save new versions of customized views that you previously stored in one of your view groups.

The out-of-the-box custom views Probe Details and Java Probe Memory Usage are **read-only** views. To save changes you should use the save as function to create a new view that can be changed and persisted.

When you are working with a custom view that you have saved, any changes you make to the view are automatically saved to the custom view. The next time you open the custom view, the changes you made the last time you opened the view are used to display the performance metrics.

Note: If you are using Diagnostics with either Business Service Management or Performance Center, the customized views are saved for the Business Service Management or Performance Center user who created the view.

If you are using Diagnostics with LoadRunner, the views are stored on the Diagnostics Server for user **admin**.

There are two ways to save a custom view once you have configured the view in the desired manner: using the pop-up menu in a custom view group and using the Save this View button in the toolbar.

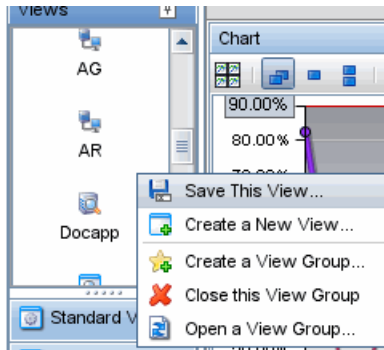
To save a view in a custom view group:

- 1 Open the custom view group where you want to store the saved view.

Note: Custom views can only be saved in the view group, My Views, or in a view group that you have created.

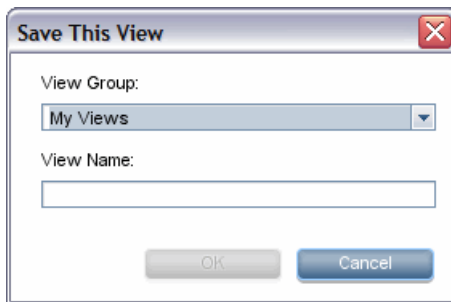


- 2 Right-click the View bar inside the custom view group, and select **Save This View**, or click the **Save This View** button in the view toolbar.



Make sure to hold the pointer over an empty area in the View bar, and not over the icon for a view.

Diagnostics displays the **Save This View** dialog box.



- 3 Select a view group from the list of view groups that can contain custom views.
- 4 Type a name for the **View Name**, and click **OK**.

The view that is currently displayed is saved as a custom view, and an icon for the new view is displayed in the custom view group.

How to Rename a Custom View

You can rename a view that you have saved in a custom view group.

Note: You can only rename views in the view group My Views or in a view group that you have created.

To rename a view:

- 1 Select the view group in the View bar that contains the view that you want to rename.
- 2 Right-click the icon for the view that you want to rename and select **Rename View** from the pop-up menu.

The Rename View dialog box opens.



- 3 Enter a new name for the view, and click **OK**.

Diagnostics renames the view and updates the view bar to display the new name.

How to Delete a Custom View

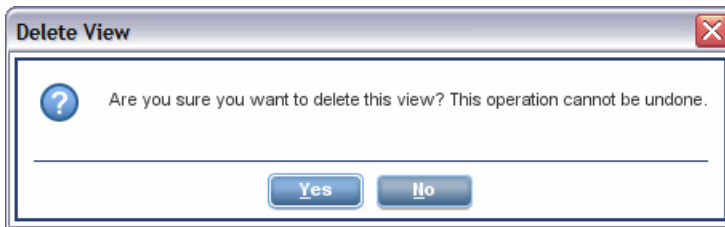
You can delete a view that you have saved in a custom view group.

Note: You can only delete views from the view group My Views or in a view group that you have created.

To delete a view:

- 1 Select the view group in the View bar that contains the view that you want to delete.
- 2 Right-click the icon for the view that you want to delete, and select **Delete View** from the pop-up menu.

Diagnostics displays the Delete View verification dialog box.



- 3 Click **Yes** to delete the view.

The selected view is deleted and no longer appears in the view group.

If the view that you deleted was displayed on the active Diagnostics screen, then the view is replaced with the default view that is displayed when Diagnostics opens.

Reference

My Views User Interface

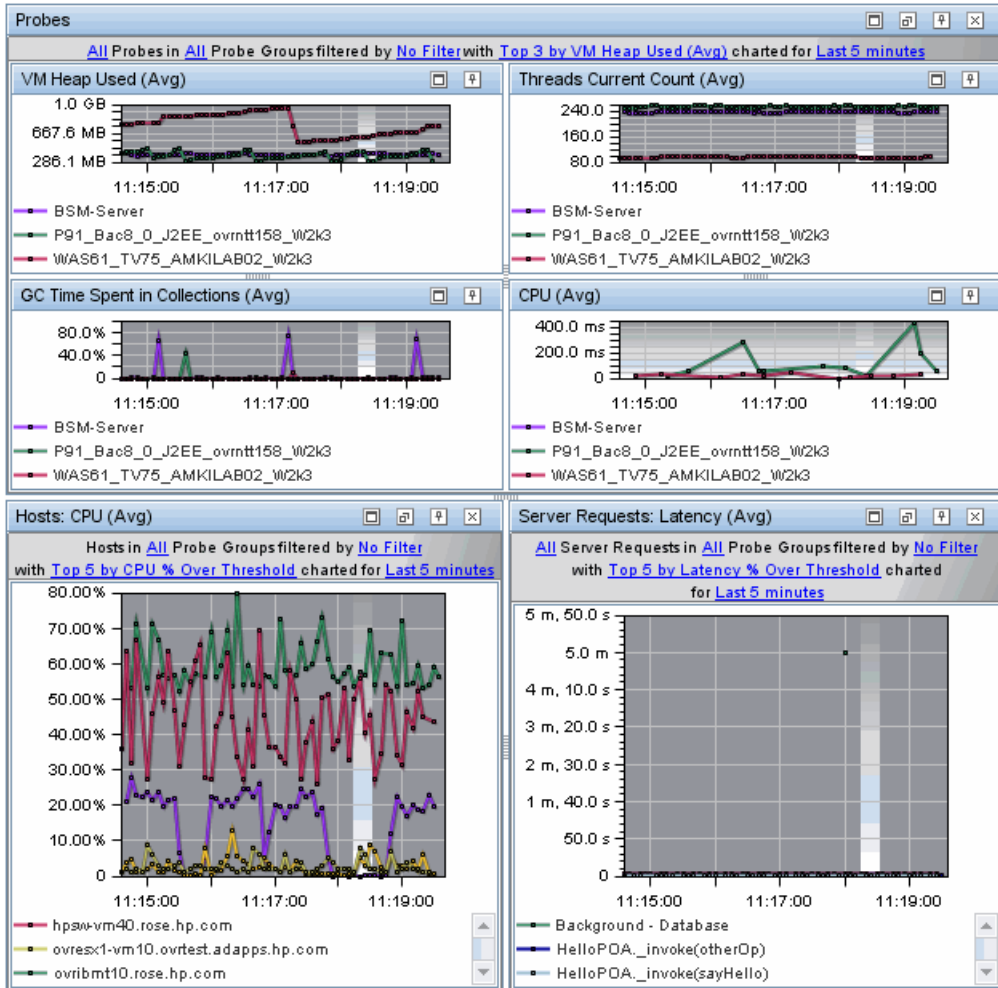
This section includes:

- ▶ [Probe Details \(in My Views\) Description on page 232](#)
- ▶ [Java Probes Memory Usage \(in My Views\) Description on page 237](#)

🔍 Probe Details (in My Views) Description

This page provides an out of the box custom view for Probe Details. This view is accessed from My Views in the View bar. The Probe Details view shows key probe metrics, server request latency and shows CPU usage on the host.

The following is an example of the Probe Details view.



To access	In the View bar, select My Views to open the My Views view group and access custom views. Two custom views are provided out of the box: Probe Details and Java Probes Memory Usage.
Important information	Both the Probe Details and the Java Probes Memory Usage custom views can be opened (Open in Custom View) within the context of a selected probe. The out-of-the-box custom views Probe Details and Java Probe Memory Usage are read-only views. To save changes you should use the save as function to create a new view that can be changed and persisted.
See also	"Open in Custom View" on page 120 "Probes View Description" on page 440 "Hosts View Description" on page 373 "Server Requests View Description" on page 389

User interface elements are described below:

UI Elements	Description
VM Heap Used graph	<p>By default, the graph displays the top three probes with the highest average VM Heap Used during the previous five minutes.</p> <p>By default, the graph is based on all probes in all probe groups. You can filter the data by probe or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average VM heap used.</p> <p>The Probe Details tooltip for the trend line in the graph displays the following:</p> <p>Probe. The name of the probe.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe, for example, Java or .NET.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click on the graph the drill down workflow is to the Probes view with each of the relevant metrics charted on separate graphs (VM Heap Used, Threads Current Count, GC Time Spent in Collections, CPU).</p>
Threads Current Count graph	<p>By default, the graph displays the top three probes with the highest average Threads Current Count during the previous five minutes.</p> <p>By default, the graph is based on all probes in all probe groups. You can filter the data by probe or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average thread current count.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the VM Heap Used graph.</p> <p>If you double-click on the graph the drill down workflow is to the Probes view with each of the relevant metrics charted on separate graphs (VM Heap Used, Threads Current Count, GC Time Spent in Collections, CPU)</p>

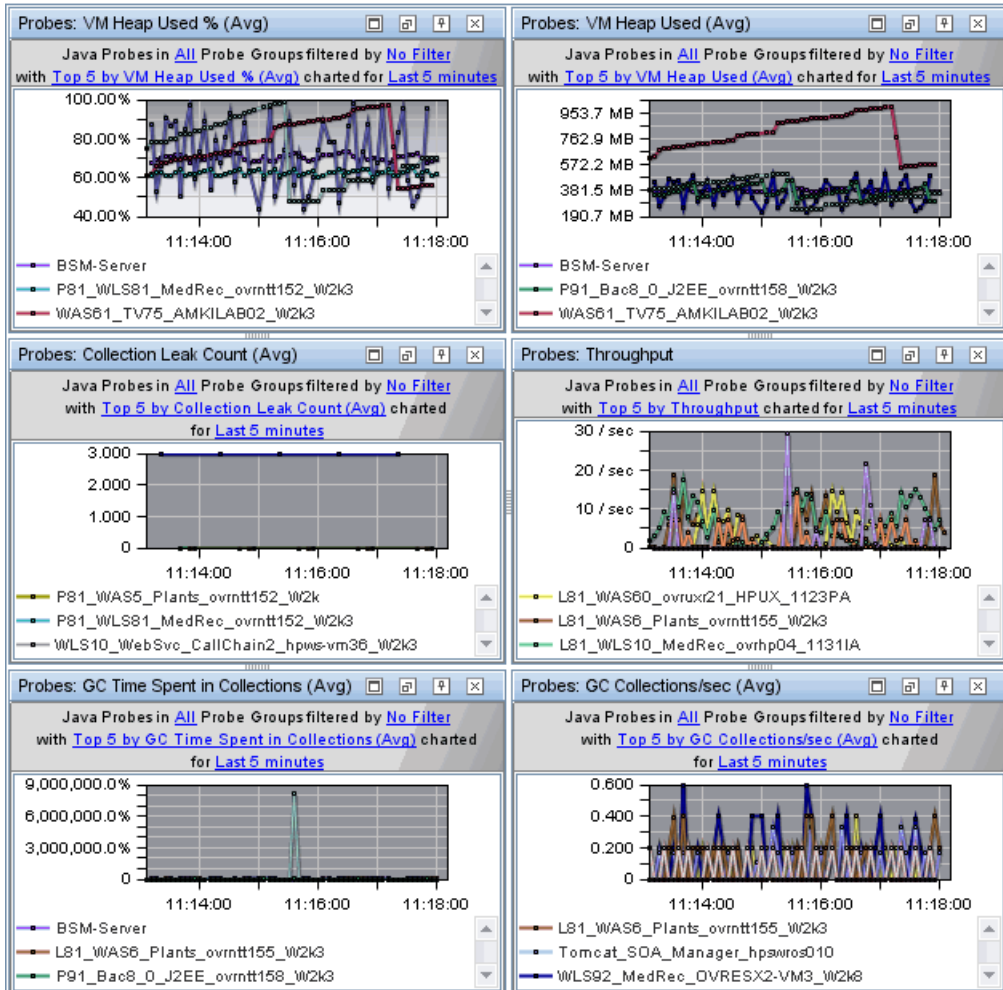
UI Elements	Description
<p>GC Time Spent in Collections graph</p>	<p>By default, the graph displays the top three probes with the highest average GC Time Spent in Collections during the previous five minutes.</p> <p>By default, the graph is based on all probes in all probe groups. You can filter the data by probe or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average GC time spent in collections.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the VM Heap Used graph.</p> <p>If you double-click on the graph the drill down workflow is to the Probes view with each of the relevant metrics charted on separate graphs (VM Heap Used, Threads Current Count, GC Time Spent in Collections, CPU)</p>
<p>CPU graph</p>	<p>By default, the graph displays the top three probes with the highest average CPU during the previous five minutes.</p> <p>By default, the graph is based on all probes in all probe groups. You can filter the data by probe or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average CPU.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the VM Heap Used graph.</p> <p>If you double-click on the graph the drill down workflow is to the Probes view with each of the relevant metrics charted on separate graphs (VM Heap Used, Threads Current Count, GC Time Spent in Collections, CPU)</p>

UI Elements	Description
Hosts CPU graph	<p>By default, the graph displays the top three probes with the highest average hosts CPU during the previous five minutes.</p> <p>By default, the graph is based on all probes in all probe groups. You can filter the data by probe or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average host CPU.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the VM Heap Used graph.</p> <p>If you double-click on the graph the drill down workflow is to the Hosts view with the relevant metric charted.</p>
Server Requests Latency graph	<p>By default, the graph displays the top three probes with the highest average server requests latency during the previous five minutes.</p> <p>By default, the graph is based on all probes in all probe groups. You can filter the data by probe or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average server requests latency.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the VM Heap Used graph.</p> <p>If you double-click on the graph the drill down workflow is to the Server Requests view with the relevant metric charted.</p>

Java Probes Memory Usage (in My Views) Description

- This page provides an out of the box custom view for Java Probes Memory Usage. This view is accessed from My Views in the View bar. The Java Probes Memory Usage view shows key memory metrics for Java probes including data on collection leaks and time spent in garbage collection.

The following is an example of the Java Probes Memory Usage view.



To access	In the View bar, select My Views to open the My Views view group and access custom views. Two custom views are provided out of the box: Probe Details and Java Probes Memory Usage.
Important information	<p>Both the Probe Details and the Java Probes Memory Usage custom views can be opened within the context of a selected probe.</p> <p>The out-of-the-box custom views Probe Details and Java Probe Memory Usage are read only views. To save changes you should use the save as function to create a new view that can be changed and persisted.</p> <p>If you double-click on any of the graphs the drill down workflow is to the Java Probes view with the relevant metric charted.</p>
See also	<p>"Open in Custom View" on page 120</p> <p>"Java Probes View Description" on page 445</p>

User interface elements are described below:

UI Elements	Description
<p>Percent VM Heap Used graph</p>	<p>By default, the graph displays the top five Java probes with the highest percentage VM Heap Used during the previous five minutes.</p> <p>By default, the graph is based on Java probes in all probe groups. You can filter the data by probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the percentage VM heap used.</p> <p>The Probe Details tooltip for the trend line in the graph displays the following:</p> <p>Probe. The name of the probe.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe, for example, Java or .NET.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click on any of the graphs the drill down workflow is to the Java Probes view with the relevant metric charted.</p>
<p>VM Heap Used graph</p>	<p>By default, the graph displays the top five Java probes with the highest average VM Heap Used during the previous five minutes.</p> <p>By default, the graph is based on Java probes in all probe groups. You can filter the data by probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average VM heap used.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the Percent VM Heap Used graph.</p> <p>If you double-click on any of the graphs the drill down workflow is to the Java Probes view with the relevant metric charted.</p>

UI Elements	Description
Collection Leak Count graph	<p>By default, the graph displays the top five Java probes with the highest average collection leak count during the previous five minutes.</p> <p>By default, the graph is based on Java probes in all probe groups. You can filter the data by probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average collection leak count.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the Percent VM Heap Used graph.</p> <p>If you double-click on any of the graphs the drill down workflow is to the Java Probes view with the relevant metric charted.</p>
Throughput graph	<p>By default, the graph displays the top five Java probes with the highest Throughput during the previous five minutes.</p> <p>By default, the graph is based on Java probes in all probe groups. You can filter the data by probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the throughput.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the Percent VM Heap Used graph.</p> <p>If you double-click on any of the graphs the drill down workflow is to the Java Probes view with the relevant metric charted.</p>

UI Elements	Description
<p>GC Time Spent in Collections graph</p>	<p>By default, the graph displays the top five Java probes with the highest average GC Time Spent in Collections during the previous five minutes.</p> <p>By default, the graph is based on Java probes in all probe groups. You can filter the data by probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average GC time spent in collections.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the Percent VM Heap Used graph.</p>
<p>GC Collections/second graph</p>	<p>By default, the graph displays the top five Java probes with the highest average GC Collections per second during the previous five minutes.</p> <p>By default, the graph is based on Java probes in all probe groups. You can filter the data by probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average GC collections per second.</p> <p>The Probe Details tooltip for the trend line in the graph is described above with the Percent VM Heap Used graph.</p>

8

Working with Topologies

This chapter describes how to work with topologies in Diagnostics. There are a number of topology views that provide you with a visual representation of the interactions and relationships between key application components. Understanding the common layout and controls in the topologies makes it easier to customize the diagram and select the way you want data displayed.

This chapter includes:

Concepts

- ▶ About Diagnostics Topologies on page 244
- ▶ Description of the Topology Layout on page 245
- ▶ The Topology Diagram on page 247
- ▶ The Topology Toolbar on page 249
- ▶ Save the Topology to an HTML Web Page on page 256

Concepts

About Diagnostics Topologies

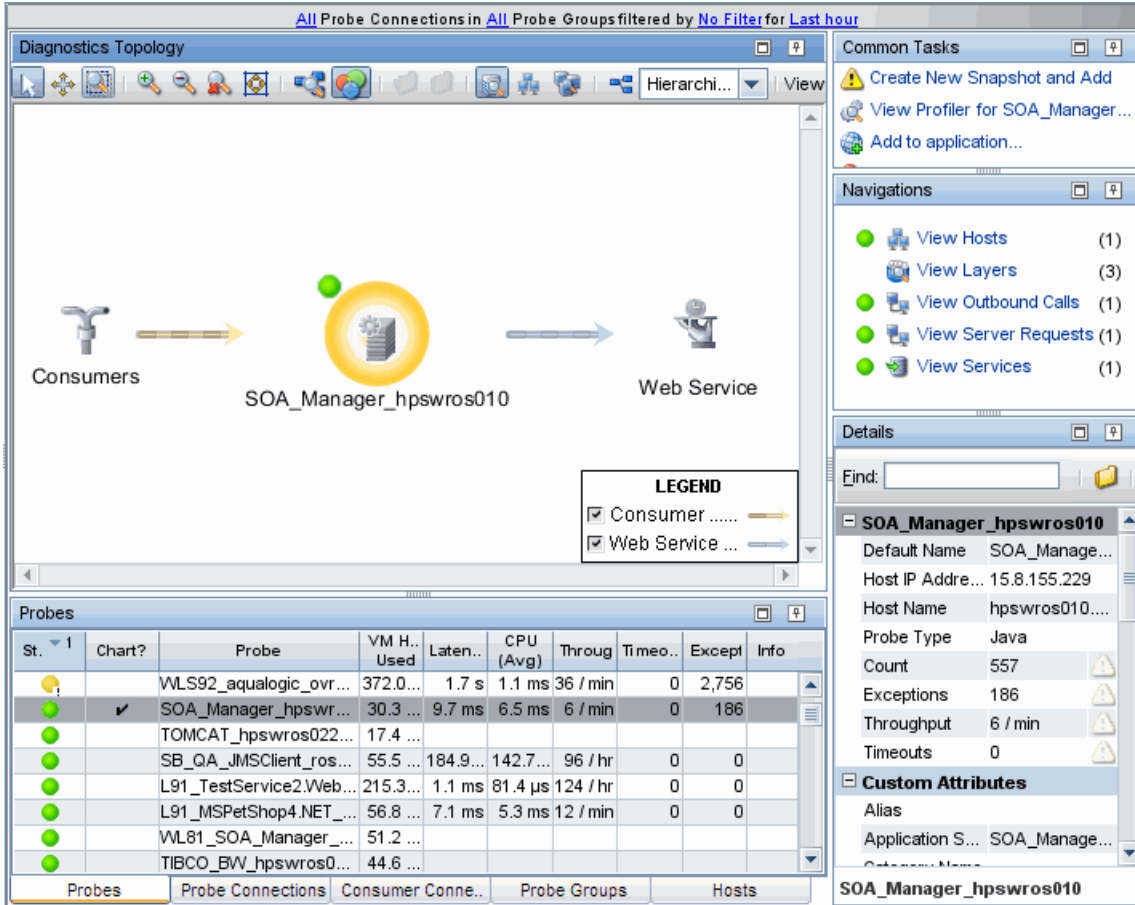
The Topology layout provides a graphical representation of the way your applications or transactions are working. It generates a diagram that shows interactions and relationships between key components, such as server requests, application servers, and databases.

A Topology diagram is displayed in the following views:

- ▶ Standard Views has a **Topology view** for probe connections, probes, probe groups, consumer connections and hosts
- ▶ For server requests there is a **Server Request Topology** view
- ▶ **Application Explorer view** has a topology layout
- ▶ Transactions specialized view group has a **Path Topology** view for business transactions
- ▶ SOA Services specialized view group has a **Service Topology view**

Description of the Topology Layout

The Topology layout has the appearance and controls of a detail layout except the topology diagram is displayed instead of a graph. The following figure shows an example of the Topology layout.



The screenshot displays the 'Diagnostics Topology' window. The main area shows a topology diagram with three nodes: 'Consumers', 'SOA_Manager_hpswros010', and 'Web Service'. Arrows indicate connections from Consumers to SOA_Manager_hpswros010 and from SOA_Manager_hpswros010 to Web Service. A legend below the diagram identifies the arrow types for 'Consumer' and 'Web Service ...'. The window includes a toolbar with various navigation and view controls, and a 'View' dropdown menu.

On the right side, there are several panels: 'Common Tasks' with options like 'Create New Snapshot and Add', 'View Profiler for SOA_Manager...', and 'Add to application...'; 'Navigations' with links to 'View Hosts (1)', 'View Layers (3)', 'View Outbound Calls (1)', 'View Server Requests (1)', and 'View Services (1)'; and 'Details' for the selected host 'SOA_Manager_hpswros010', showing fields like 'Default Name', 'Host IP Address', 'Host Name', 'Probe Type', 'Count', 'Exceptions', 'Throughput', and 'Timeouts'.

At the bottom, a 'Probes' table is visible, listing various probes with their status, chart availability, and performance metrics.

St.	Chart?	Probe	VM H. Used	Laten..	CPU (Avg)	Throug	Timeo..	Except	Info
		WLS92_aqualogic_ovr...	372.0...	1.7 s	1.1 ms	36 / min	0	2,756	
	<input checked="" type="checkbox"/>	SOA_Manager_hpswr...	30.3 ...	9.7 ms	6.5 ms	6 / min	0	186	
		TOMCAT_hpswros022...	17.4 ...						
		SB_QA_JMSClient_ros...	55.5 ...	184.9...	142.7...	96 / hr	0	0	
		L91_TestService2.Web...	215.3...	1.1 ms	81.4 μs	124 / hr	0	0	
		L91_MSPetShop4.NET...	56.8 ...	7.1 ms	5.3 ms	12 / min	0	0	
		WL81_SOA_Manager_...	51.2 ...						
		TIBCO_BW_hpswros0...	44.6 ...						

The topology diagram includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by grouping entities, changing diagram type, or changing connection (link) information.

Note: The Application Explorer includes a topology display that is similar, but some functions are different. Please see Chapter 11, "Application Explorer View" for details.

Topology Diagram

The topology is a mapping of entities in an application and the connections between entities. The topology diagram provides you with a visualization of your application or transactions. See "The Topology Diagram" on page 247 for details.

When you first access the Standard Topology view, you see a message in the place of the topology diagram instructing you to select an entity in the table to generate a topology.

Entity Table

The entity table in a topology layout is used like the table in any detail layout. You select an entity in the table to generate a topology for that entity. All other entities with connections to the entity you select in the table are marked with a check mark in the chart column and are also included in the topology. You can filter the data by **Probe Connections**, **Probe Group** and **Time Range**.

In the Application Explorer topology view there is no entity table. Instead you will see a status and navigation pane and a metrics summary pane.

A series of tabs is available at the bottom of the view. In the Standard Topology view you can select the following tabs: Probe Connections, Probes, Probe Groups, Hosts, Consumer Connections.

Common Tasks and Navigations

The common tasks and navigations pane in the Topology view and the Service Topology view displays the links that are available for the entity you select in the entity table. The relevant tasks and navigations will vary depending on the type of entity you select.

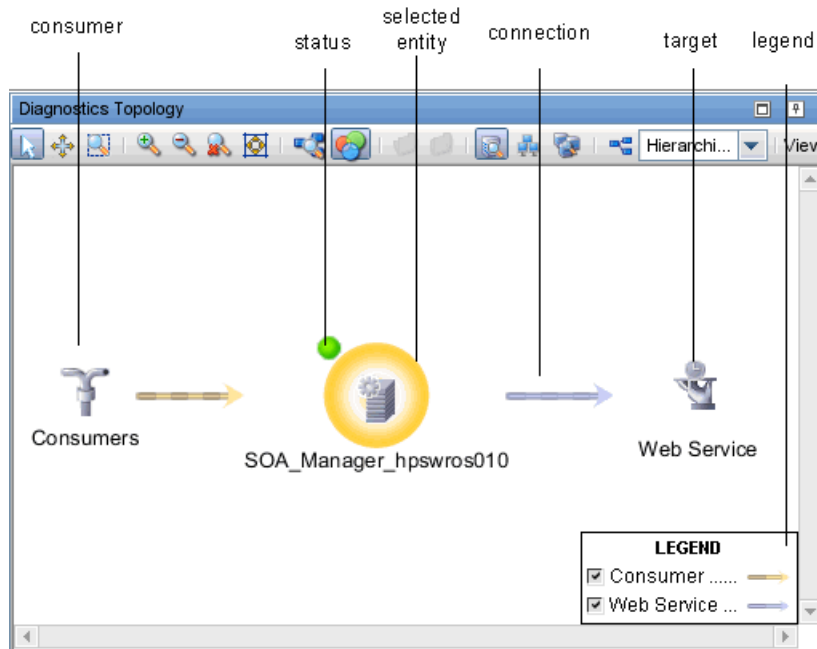
In the Application Explorer topology view the relevant navigations are shown in the status and navigations pane.

Details

The Details pane in the Topology view and the Service Topology view lists the metrics for the selected row in the entity table.

The Topology Diagram

The objects that appear in the topology diagram are shown in the following figure, and are described in the section that follows. See "The Topology Toolbar" on page 249 for a description of the items in the toolbar.



Selected Entity. Entities with a blue 'halo' are selectable in the topology. When an entity is selected, it is displayed with a yellow halo. Metrics for a selected entity are displayed in the details pane. You can right-click entities on the diagram to see a tooltip containing information on that entity.

Status. The colored dot next to an entity in the topology indicates the status of the entity. Status is based on a variety of factors, including metric thresholds. Status is only available when you view time ranges that are an hour or less. Status is shown as gray when no status is available, or when the time range exceeds an hour. See "Status Indicators" on page 517 for more information on how status is calculated.

Connection. The connection arrows represent aggregated calls to applications within your environment. Connections can be shown between various entities such as probes, probe groups, Web services and consumers, or targets. The arrow on the connection indicates the direction of the communication.

The color of the connection arrow can indicate the type of call or performance status and load, depending on the selection from the View Link drop-down menu in the topology toolbar. When you click a connection in the topology, the line is emphasized with a black stripe (indicating that you have selected it).

Consumer. The consumer icon represents calls to a probe (or other entity) coming from a non-instrumented source (not monitored by a probe). These calls are grouped together in the topology under the consumer node. You can't select the consumer icon but if you want to see which consumers are making the calls represented in the topology you can select the consumer connections arrow coming from the consumer icon. The Consumer Connections tab then opens to display details about all the different consumers making calls to a particular entity.

You can configure consumer IDs to identify specific consumers. The default consumer ID is the IP address but you can configure consumers in a number of ways. "About Consumer IDs" on page 663.

Target. The target icon represents calls to a non-instrumented source (not monitored by a probe) or when multiple connections have the same origin or destination. Targets, as determined through the instrumentation of your probes, represent the origin of inbound calls or the destination of outbound calls. A variety of icons are used to represent the different types of targets that Diagnostics recognizes. You cannot select targets in the diagram since targets do not contain any metric information.

For example, you could see the following entities in a topology, each represented by different icons.

- Probes
- Probe Groups
- Hosts
- Web Services
- Consumers
- Connections
- Targets.

The Topology Toolbar

The topology toolbar appears at the top of the topology display, below the view title. The toolbar allows you to control the appearance of the topology by scrolling and zooming, and by toggling the appearance of useful aids, such as the topology overview box and the legend.



Panning

To reposition the diagram so that the entities that you want to see are visible in the diagram, you can scroll the diagram using the scroll bars or you can pan by putting the pointer in panning mode.

When the diagram is not completely displayed in the view, the scroll bars are activated, so you can use them to scroll to the location in the diagram that you want to see. When the vertical scroll bar is active, you can use the mouse wheel, instead of the scroll bar, to scroll vertically within the diagram.



To pan to a specific selected area of the diagram, click **Pan** to put the pointer into pan mode. When in pan mode, the pointer is shaped like a hand. Drag the pointer to reposition your view of the diagram. When you release the mouse button, the diagram continues to be displayed in the position where you released the mouse button. You may continue to pan as long as the pointer remains in pan mode.



To return the pointer to selection mode, click **Pan** again, or click **Select Mode**.

Zooming

Diagnostics provides several controls that allow you to zoom in the diagram to get a closer look at parts of a displayed topology diagram, or to zoom out to see the bigger picture. There is a limit to how far you can zoom the diagrams in or out. When you have zoomed to one of these limits, using the zoom controls no longer affects the appearance of the diagram.

Zooming In



To zoom in a selected area of the diagram, click **Zoom Mode** to put the pointer into zoom mode. When in zoom mode, the pointer is shaped like a magnifying glass. Drag the pointer to define the location and size of the area that you would like to zoom. When you release the mouse button, Diagnostics zooms in to show the selected area at maximum magnification. To return the pointer to selection mode, click **Zoom Mode** again, or click **Select Mode**.



To zoom in the entire diagram to get a closer look at an area of the diagram, click **Zoom In**. Diagnostics zooms in the Diagram, keeping the same entities in the center of the diagram as it provides a closer look. You may need to use the scroll bar or panning to reposition the center of the view as you zoom in.

You can also zoom in using the mouse wheel. To use the mouse wheel to zoom in, hold down the CTRL key while rolling the wheel forward.

Zooming Out



To zoom out to get a more complete view of the diagram, click **Zoom Out**. Diagnostics zooms out the Diagram, keeping the same entities in the center of the diagram as it provides a broader view. You may need to use the scroll bar or panning to reposition the center of the view as you zoom out.

You can also zoom out using the mouse wheel. To use the mouse wheel to zoom out, hold down the **Ctrl** key while rolling the wheel backwards.

Fit Contents to View



To zoom to fit the topology diagram into the view, click **Zoom Contents to Fit View**. Diagnostics zooms so that the diagram fits the view, keeping the same entities in the center of the diagram as it provides a broader view. You may need to use the scroll bar or panning to reposition the center of the view as you zoom out.

Restoring the Diagram to Primary Zoom



Diagnostics sizes the diagram for the best fit to your browser window when it first displays a diagram. You can choose to zoom the diagram in or out. Then you can restore the diagram to this primary zooming resolution by clicking **Reset Zoom**. The custom layout changes that you made to the position of probes and probe connections are not impacted.

Diagram Overview



The Topology diagram overview provides you with a view of the entire diagram. The diagram overview helps you to scroll and zoom the diagram, so the information that you need is visible in the Topology view. The diagram overview is especially helpful when you have zoomed in on the diagram.

You can also use the diagram overview to pan and zoom the view of the diagram. The diagram overview is outlined by a black view frame, which is initially not visible beyond the edges of the diagram overview. You can pan the primary diagram by dragging this view frame to reposition it within the diagram overview. You can zoom the primary diagram by resizing the view frame at the handles, and then dragging it to the desired zoom level.

To display the diagram overview, click the **Diagram Overview** button in the diagram toolbar. Diagnostics displays the overview box in the lower-right corner of the diagram. Clicking the **Diagram Overview** button again removes the overview box. If the legend was displayed, it is removed when the diagram overview box is displayed.

Diagram Legend



The graph legend lists each of the probe connection types that have been included in the currently displayed diagram. It shows the specific color used to draw each probe connection type.

To display the diagram legend, click the **Diagram Legend** button in the diagram toolbar. Diagnostics displays the legend in the lower-right corner of the diagram. Clicking the **Diagram Legend** button again removes the legend. If the diagram overview box was displayed, it is removed when the legend is displayed.

You can use the check box that precedes each probe connection type entry in the legend to indicate which probe connection types or load are to be diagrammed. When a probe connection type is not checked, all of the probe connections for that type are removed from the diagram. This can be useful in a complicated diagram, in which the number of probe connections can make it difficult to understand.

Note: Filtering the diagrammed probe connections in the legend does not impact the probe connection metrics listed in the graph entity table.

Grouping Probes

You can change the way the diagram is drawn by selecting to group probes by hosts or by probe groups. Grouping probes can simplify the appearance of the diagram.

Grouping Probes by Hosts



To group probes by the hosts, click **Group Probes by Host** in the toolbar. The probes in the diagram are replaced with probe hosts that contain the probes installed on the host.

Grouping Probes by Probe Group



To group probes by probe groups, click **Group Probes by Probe Group** in the toolbar. The probes in the diagram are replaced with probe groups that contain the probes added to the probe group.

Ungroup Connected Probes



To cause the probes to again be displayed as individual probes in the diagram, click **Don't Group Connected Probes** in the toolbar.

Working With Grouped Probes



When probes are grouped in the diagram, a small toggle icon is shown above the icon for the group. If you click this icon, or right-click the group icon, the group is opened so you can see each of the probes in the group, along with the probe connections to the targets.

With a large topology it may be necessary to click the Layout button after ungrouping probes to properly display overlapping connections.



You can collapse the expanded probe group by clicking the icon in the group header, or by right clicking anywhere in the group box.



You can expand all of the groups in the diagram by clicking **Expand all Nodes** in the toolbar. Diagnostics opens all of the groups in the diagram.



You can collapse all of the groups in the diagram by clicking **Collapse all Nodes** in the toolbar. Diagnostics collapses all of the groups in the diagram.

Adjusting the Layout

The topology diagram can be drawn in several different layouts. You can customize the layouts by dragging the targets and probes to reposition them in a way that makes the probe connections easier to see.

Selecting the Diagram Layout

Diagnostics can display the diagram using several different layout formats. The default layout is hierarchical. You may find that one of the other layouts makes your diagram easier to understand.

To change to an alternate layout, select the layout from the **Layout** drop-down menu. Diagnostics redraws the diagram using the selected layout.

The drop-down menu for Layout allows you to select layout types including:

- ▶ **Hierarchical.** Organizes nodes without overlaps in horizontal or vertical levels. Arranges the graph such that the majority of links are short and flow uniformly in the same direction (from left to right, from top to bottom, and so on). Reduces the number of link crossings. Most of the time, produces drawings with no crossings or only a small number of crossings.
- ▶ **Tree.** Takes into account the size of the nodes so that no overlapping occurs. Optionally, reshapes the links to give them an orthogonal form (alternating horizontal and vertical line segments).
- ▶ **Uniform.** Often provides a drawing without any or with only a few link crossings and with approximately equal length links for small- and medium-size graphs having a small number of cycles. The maximum number of nodes for which you can use the algorithm depends on the connectivity of the graph and is difficult to predict.
- ▶ **Spring.** Often provides a drawing with no link crossings or few crossings for small- and medium-size graphs. The maximum number of nodes for which you can use this layout depends on the connectivity of the graph: hundreds for trees, dozens for cyclic graphs.

Customizing the Diagram Layout

You can reposition the probes and the targets in the topology diagram so the probe connections become more clear. To reposition a probe or probe connection, drag the node you want to move to the intended location. Diagnostics redraws the probe connections to and from the relocated node.

Note: Diagnostics redraws the diagram when an entity has been detected that must be added to the diagram, or when an entity must be removed from the diagram. If the view is zoomed in, the addition of new nodes to the diagram does not cause the diagram to zoom to fit the contents.

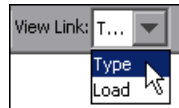
Restoring the Layout



When you have customized the diagram layout by repositioning probes and targets, you can have the diagram redrawn clicking **Layout all Nodes** in the diagram toolbar. Diagnostics redraws the diagram, keeping your zoom and pan settings as you set them.

View Link (Type or Load)

The View Link drop-down menu shows information about probe connections in the current Topology diagram.



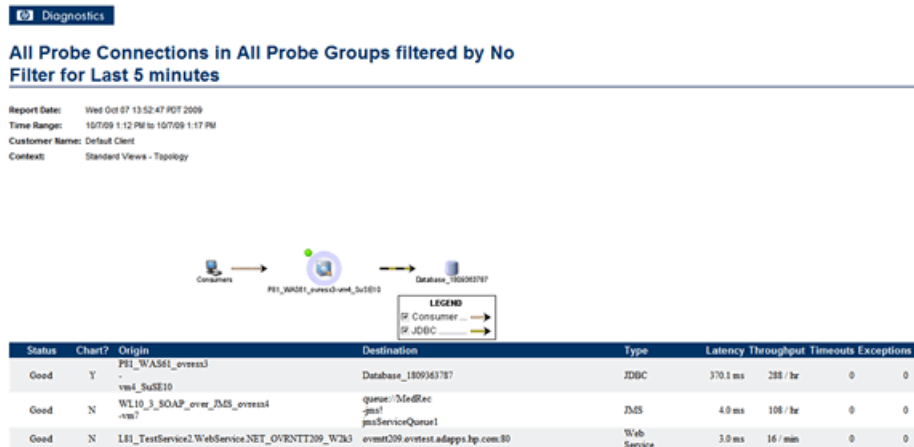
By default, the connections are initially colored according to the Type of call (for example, one color for ADO, another for JDBC, and so on).

You can use the View Link drop-down menu to select **Load**, which uses line thickness and color to indicate the connection's frequency of use and latency time combined (thick as heavy, thin as light) as well as its performance level status (green as good, yellow as warning, and red as critical problem).

Save the Topology to an HTML Web Page



You can save the current Topology as an HTML web page. When you have the view you would like to capture, click the **Saves this view as an HTML web page** button in the View toolbar (upper-right side of the view). The exported view shows the information in the current display as a web page that is also saved to an HTML file you name. The following screen is an example of a Topology HTML web page.



Note: Whatever diagram display resolution appears at the time of your export is duplicated in your report.

For the Standard Topology view, the Probe Connections tab of the data entity table is what is saved to your HTML file.

9

Working with Instance Trees

This chapter describes instance tree data collected by Diagnostics and how you can drill down to the associated call profiles for the most interesting instance trees.

This chapter includes:

Concepts

- ▶ About Instance Trees on page 258
- ▶ Solve Performance Problems Using Instance Tree Data on page 261
- ▶ Cross VM Trees on page 265
- ▶ Exception Data on page 266
- ▶ SOAP Fault Data on page 269
- ▶ Types of SOAP Fault Markers on page 271
- ▶ When Instance Trees Are Not Sufficient on page 271
- ▶ Aggregate Trees on page 272

Concepts

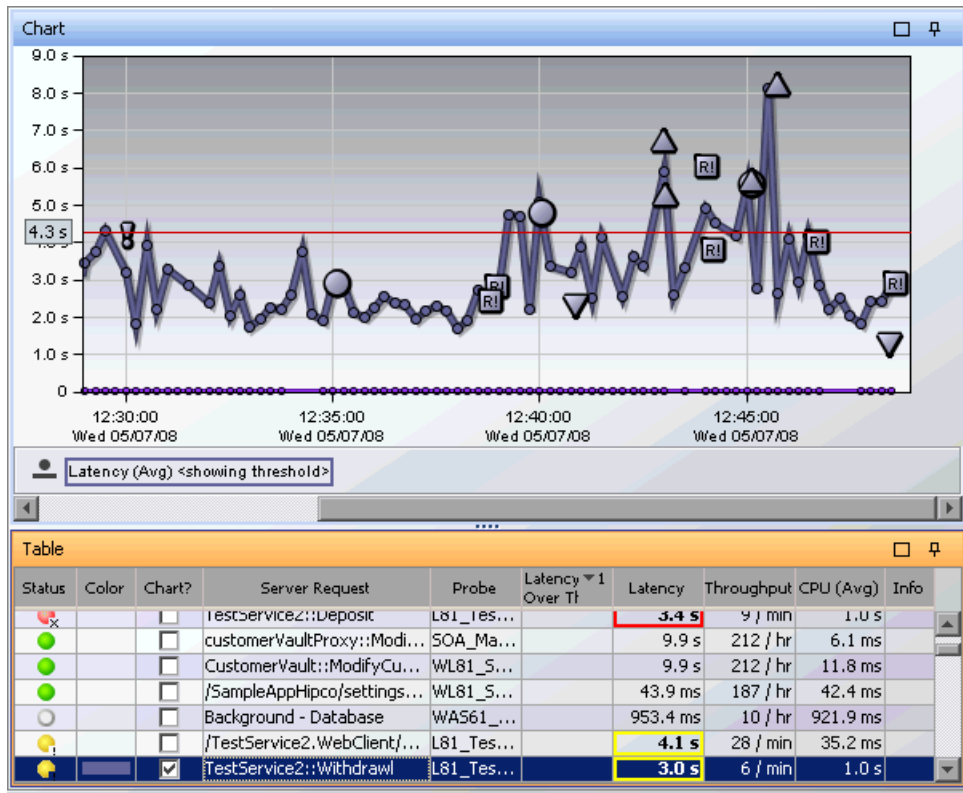
About Instance Trees

Diagnostic probes work by adding instrumentation before and after important methods in your application to provide a simplified program trace that includes elapsed times for performance optimization and problem diagnosis.

Diagnostics does not record every captured trace because of storage considerations. Instead, at five-minute intervals, Diagnostics uses an algorithm to select the most interesting instance trees for each server request. The selected instance trees are stored to disk and the trees that were not selected are discarded.

For each server request on each probe, Diagnostics saves an instance tree for the fastest, slowest, second slowest and median or average invocation of that server request. Diagnostics also saves a randomly selected end-to-end complete trace of the entire cross-VM tree. When the server request makes external calls (across two or more virtual machines) such as RMI, the cross-VM tree includes the trees of the remote system if the remote system was also instrumented. And if methods throw exceptions or have SOAP faults, this information is saved and details provided in exceptions instance trees.

An example of a Server Requests view showing some of the instance tree markers is shown below. The markers are placed on the graph to indicate the total latency and the end time of the server request for which the instance tree was created.



Four instance tree marker icons are used to represent the four different types of instance trees that are saved:



- **Maximum Instance Tree.** The maximum instance tree for a server request represents an instance tree with the largest latency for the time period. This tree depicts one of the worst-performing invocations of the server request and resulting root method. This is one of the primary tools used for diagnosing the root cause of poor application performance.



► **Minimum Instance Tree.** The minimum instance tree for a server request represents an instance tree with the smallest latency for the time period. This tree depicts one of the best-performing invocations of the selected server request and resulting root method. It can be useful for comparison to instances that perform poorly.



► **Average Instance Tree.** The average instance tree for a server request represents an instance tree that represents the average latency for the time period. This tree depicts a typical invocation of the selected server request and resulting root method call. It can be useful for comparison to instances that perform poorly.



► **Cross VM Instance Tree.** The cross VM instance tree for a server request represents an instance tree that includes a call to another server request by using a technology such as RMI or Web services.

The following icons are used to represent exceptions and faults. Additional information is collected by the probe when exceptions or SOAP faults occur. You can drill down to the call profile graph and view exception or fault details for a method, including the stack trace.



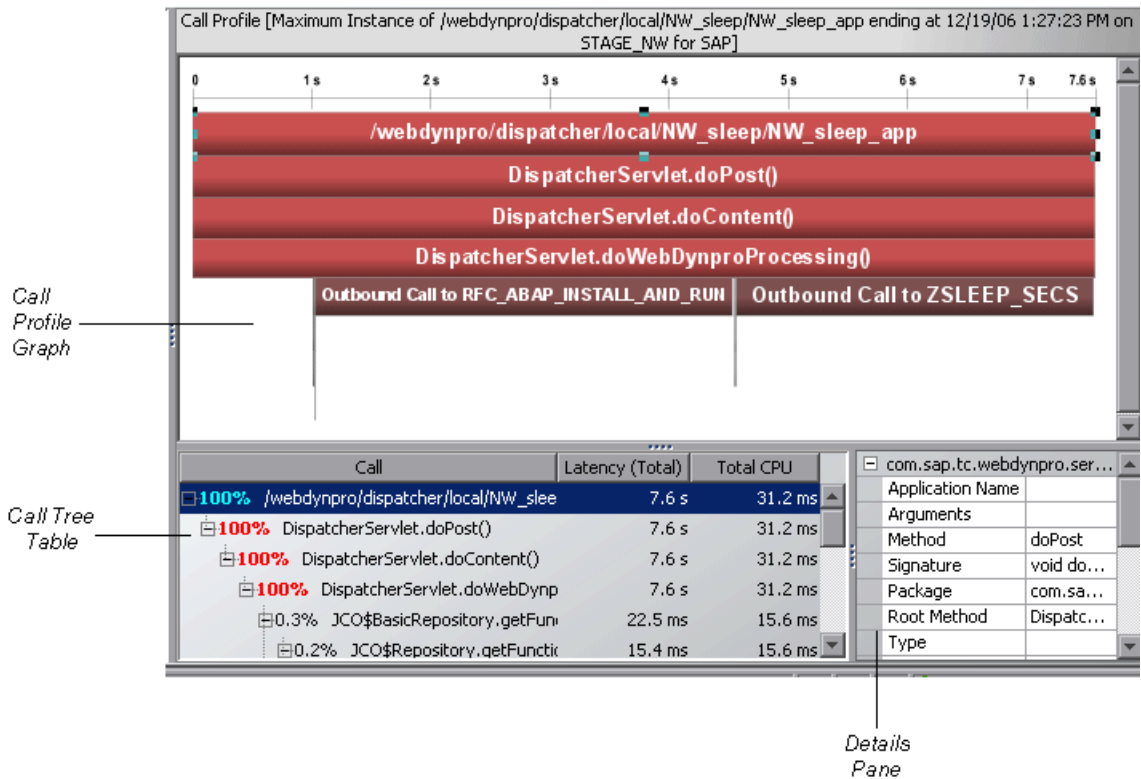
► **Exception.** The exclamation mark indicates where an exception occurred. When you click the icon, a call profile opens. This call profile has the methods that threw the exceptions marked with a yellow dashed outline and an **Exceptions** tab providing details on the exception. A stack trace for the exception is included with the details.



► **SOAP Faults.** The icon indicates where an soap fault occurred. See "Types of SOAP Fault Markers" on page 271 for details on the icons used to represent different types of faults. When you click the icon, a call profile opens. This call profile has a SOAP Faults tab providing details on the fault, including the stack trace.

When you select an instance tree icon, the program trace is presented in a call profile visualization.

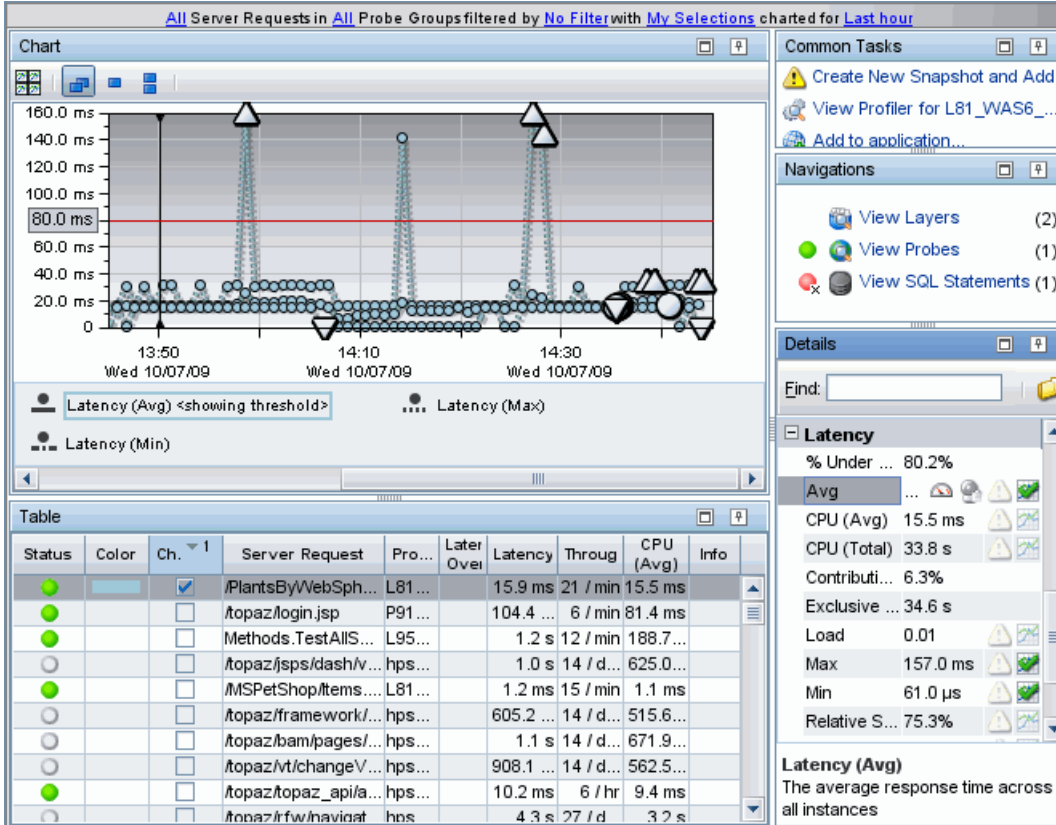
An example of a call profile visualization of an instance tree is shown below:



Solve Performance Problems Using Instance Tree Data

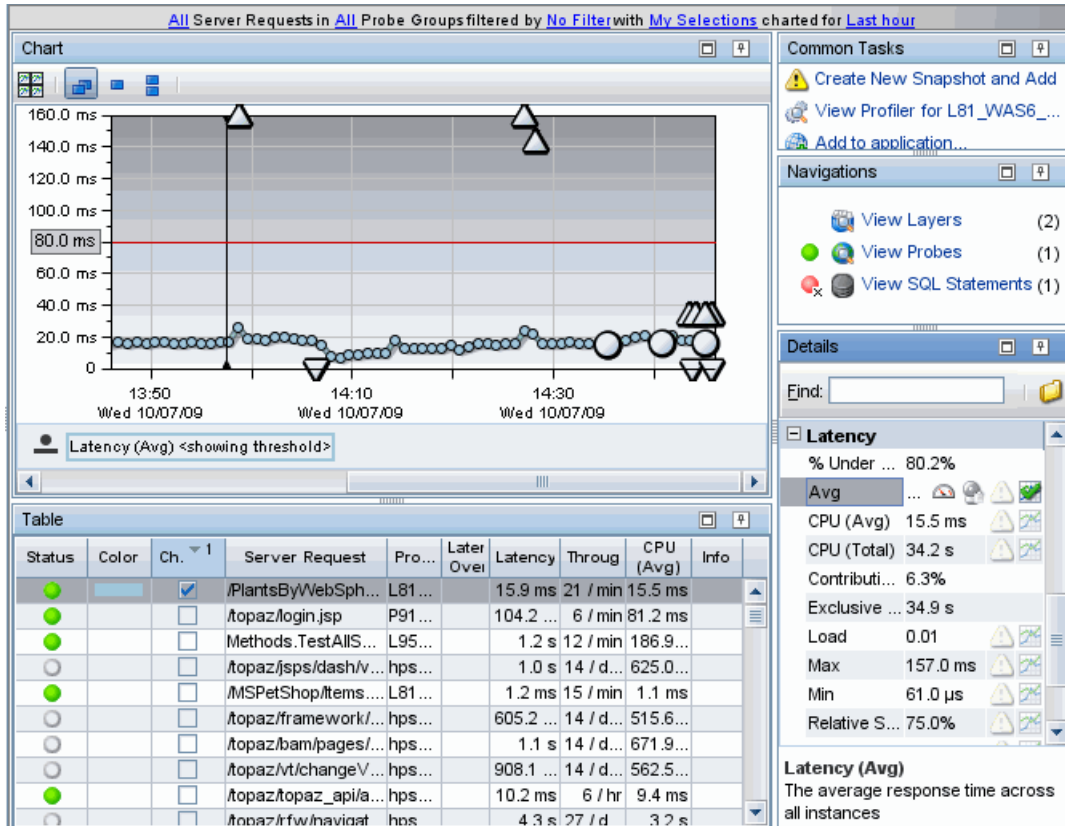
To allow detailed analysis of your application's performance, you can display the average, minimum, maximum, and standard deviation response times on the same graph.

The following example shows how to use instance tree data to analyze a performance problem. Look at the Server Requests view below.



The graph shows that the response time for the `/PlantsByWebSphere/ servlet/ShoppingServlet` server request generally varies between about 10 ms and 20 ms, and sometimes spikes above 160 ms.

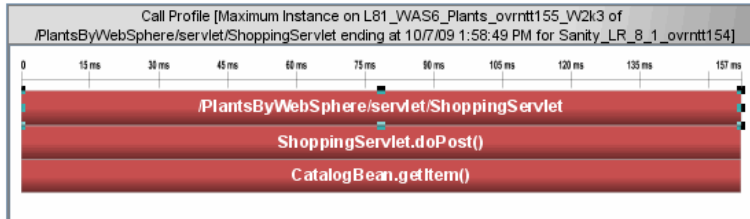
The following image is less noisy. It shows the charted trend for the Average Latency without trend lines for the minimum and maximum latency:



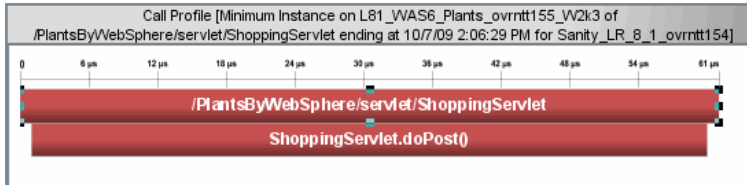
The trend line for this metric includes instance tree icons to indicate where the fastest, slowest, and median instance trees were recorded.

To understand why `/PlantsByWebSphere/servertlet/ShoppingServlet` is sometimes four times slower than average (or five times slower than the best case), compare the captured instance trees against one another to see what is different about each case:

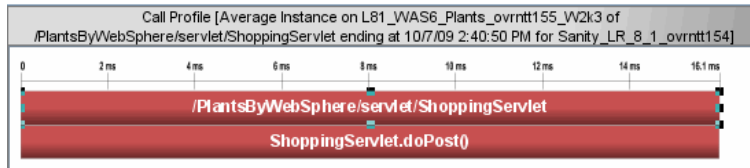
The maximum (slowest) instance tree call profile looks as follows:



The minimum (or fastest) instance tree call profile looks as follows:



The average (or median) instance tree call profile looks as follows:



Comparing these three call profiles, it becomes apparent that the `getItem` method (**CatalogBean.getItem**) is slowing down the server request in the slowest instance. Note that there is not much difference between the fastest and average calls in terms of application behavior.

Of course, it is not always true that the average profile will look the same as the minimum profile. For example, consider a server request that queries information out of an EJB. There are at least three basic execution time profiles for this operation:

- ▶ The EJB is cached in the application server memory and the retrieval is nearly instantaneous. This would be the minimum call profile.
- ▶ The EJB must be retrieved from the database. Depending upon usage statistics, this could be the average call profile.

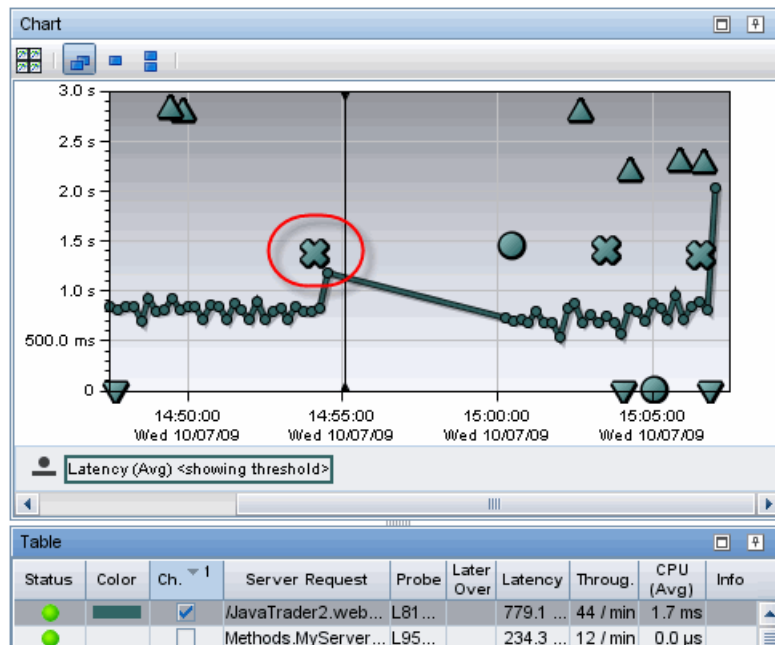
- The server request must wait a significant amount of time for a pooled database connection prior to requesting the EJB data. This could be the maximum call profile.

Cross VM Trees

Cross-VM trees are collected to help you quickly understand average application behavior when multiple virtual machines are involved.

The goal is to help you triage the problem to a particular virtual machine so that you can examine the situation in more detail. This can be done, for example, by pulling up application JMX metrics, or looking at system metrics such as CPU utilization or disk I/O on the problematic machine.

The following image is an example of a trend for a server request that makes cross-VM calls:



The x-icon indicates where a cross-VM call tree was recorded. When you click the icon, the following cross-system call profile opens:



The red nodes in the profile are the processing that took place on the origination system, and the blue nodes are the processing that took place on the child system, **ovrntt209**.

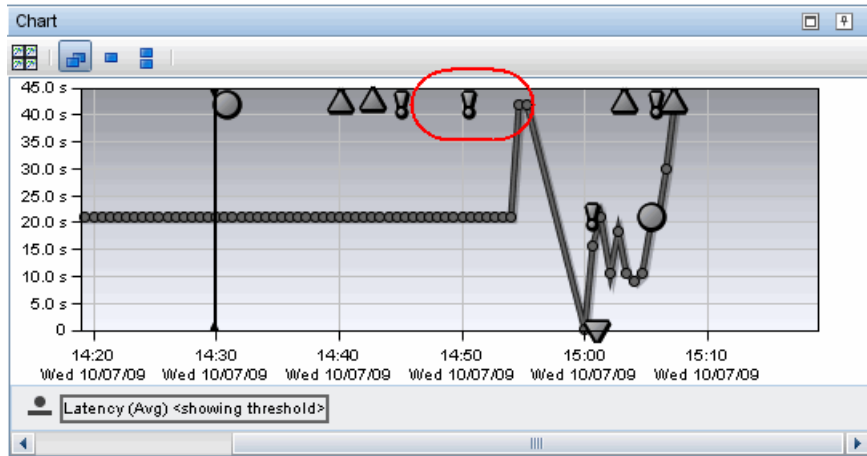
This call tree indicates that over half of the elapsed time for this server request was caused by processing in the child system.

Exception Data

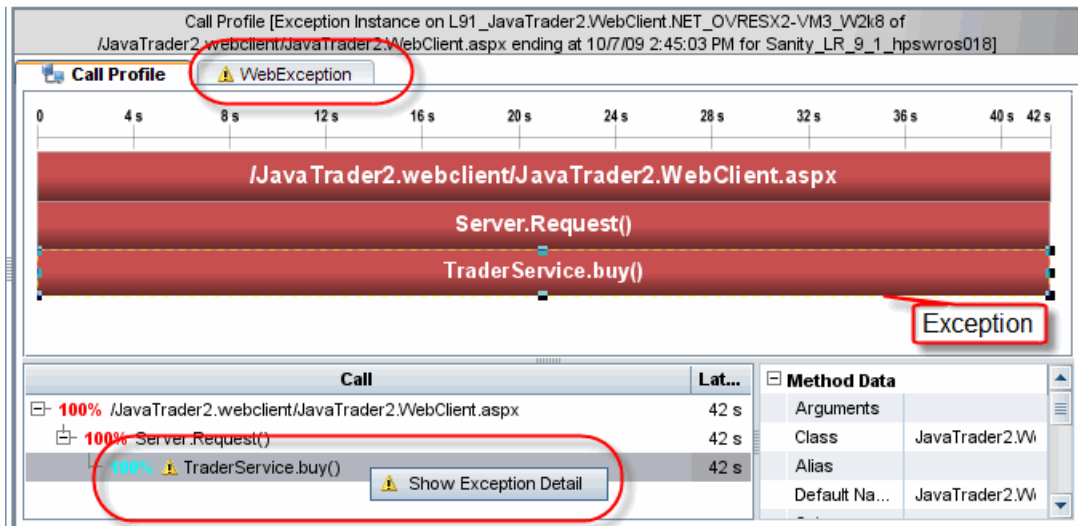
Information on exceptions is also collected by the probe. An example of the Probes view graph entity table is shown below where you can see the number of exceptions thrown for each probe.

Status	Color	Ch. 1	Probe	VM Heap Used	Latency	CPU (Avg)	Through..	Timeo...	Exceptions	Info
		<input checked="" type="checkbox"/>	L91_JavaTrader2.WebCli...	187.8 MB	19.9 s	629.2 µs	8 / min	0	152	
		<input type="checkbox"/>	L81_MSPetShop.NET_OV...	153.4 MB	1.6 ms	1.5 ms	37 / min	0	0	
		<input type="checkbox"/>	WLS10_WebSvc_CallCh...	188.5 MB	80.2 ms	5.1 ms	11 / min	0	0	

If you drill down to the server requests for this probe collection you will see the charted trend with exceptions indicated by an icon.

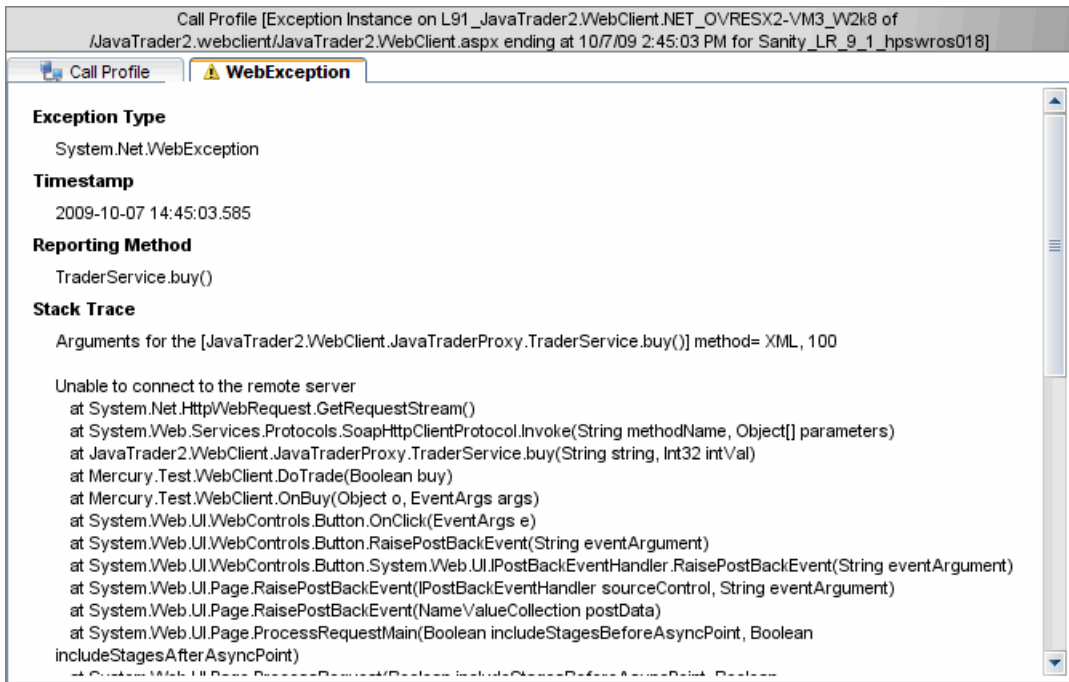


The exclamation mark indicates where an exception occurred. When you click the exception instance icon, a call profile opens.



The call profile has the method that threw the exception marked with a yellow dashed outline. In the call tree the method throwing the exception is marked by a small exclamation mark icon.

You can select the Exceptions tab to get details on the exception including the stack trace. Or you can right-click the exception in the call tree and select Show Exception Detail. This is useful when there are several tabs with detail data. The Exception tab is labeled with the exception name (not including the package and class information).



You can copy and paste the stack trace listing in this view to send to others for use in diagnosing a problem.

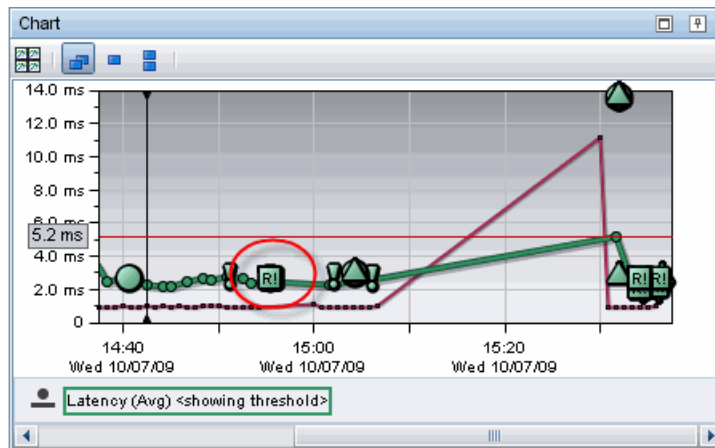
Configuring Collection and Display of Exception Data

By default, every exception that occurs in the monitored application is a candidate for the exception instance tree. Collecting all exception information is usually undesirable however, because exceptions that are not of interest overload the display as well as the data collection and transfer operations. You can therefore limit the exception types for which data is collected by the probe.

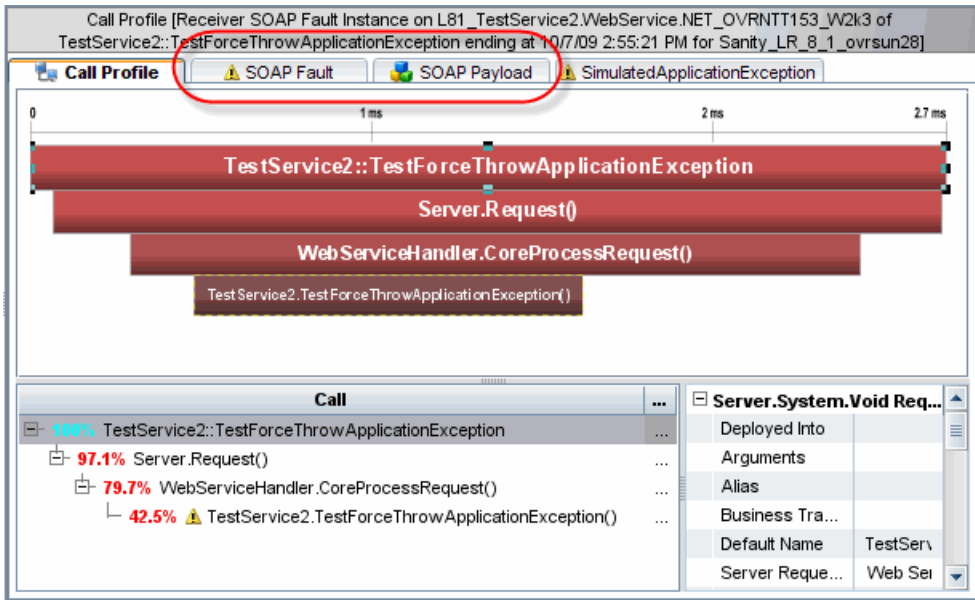
See the *HP Diagnostics Installation and Configuration Guide* for details on limiting exception tree data.

SOAP Fault Data

If methods throw SOAP faults, these are marked in the Server Requests view with an icon.



This is an example of the type of icon used to indicate where an soap fault occurred. See "Types of SOAP Fault Markers" on page 271 for details on the icons used to represent different types of faults. When you click the icon, a call profile opens.








You can select the **SOAP Fault** tab to get details on the fault including the stack trace. You can select the **SOAP Payload** tab to get details on the payload.

For details, see "Configure Collection of SOAP Faults and Payloads" on page 409. See also the *HP Diagnostics Installation and Configuration Guide* for additional information on configuring the probes to capture SOAP fault data.

Note: The Java probe requires the Diagnostics SOAP message handler to collect payload on SOAP faults. See the *HP Diagnostics Installation and Configuration Guide* for details on configuring the SOAP message handler for Java probes.

Types of SOAP Fault Markers

The following table describes each type of SOAP fault and indicates the icon used to represent it in Diagnostics views:

Icon	SOAP Fault	Description
	VersionMismatch	The processing party found an invalid namespace for the SOAP Envelope element.
	MustUnderstand	An immediate child element of the SOAP Header was not understood.
	DataEncodingUnknown	SOAP header block or SOAP body child element uses unsupported data encoding.
	Sender	The message was incorrectly formed or did not contain the appropriate information required. This includes the Invalid Operation SOAP faults. SOAP 1.1 Client faults are displayed in Diagnostics as Sender faults.
	Receiver	The message could not be processed for reasons attributable to the processing of the message rather than to the contents of the message itself. SOAP 1.1 Server faults are displayed in Diagnostics as Receiver faults.

When Instance Trees Are Not Sufficient

For some applications, three types of instance trees do not provide sufficient information to make it possible to distinguish between possible application behaviors. For example, a common enterprise pattern is to have a single point of entry into a Java application (a "controller" servlet) that dispatches between completely different commands each with completely different performance behaviors.

A URL for such a server request may appear inside Diagnostics as follows:

```
/controller
```

where the actual URLs used by customers to access the system may look like this:

```
/controller?action=checkout&userId=43a893c43&itemId=889fe  
/controller?action=browseItem&userId=43a893c43&itemId=889fe  
/controller?action=listCategory&userId=43a893c43&categoryId=438
```

In this case, maximum, minimum, and average trees for **/controller** are insufficient to diagnose application problems—the call profile for each action processed could be significantly different.

For cases like this you should customize the probe instrumentation for the **/controller** servlet slightly so that the probe can pick up the equivalent of the 'action' parameter as part of the Diagnostics server request. This requires a minor change to the instrumentation that HP Software Customer Support can help you make.

With this customized instrumentation, you would be able to see three distinct server requests inside Diagnostics:

```
/controller?action=checkout  
/controller?action=browseItem  
/controller?action=listCategory
```

Each of these three server requests would have minimum, maximum, and average instance trees to assist in your problem diagnosis.

Aggregate Trees

The Diagnostics Profiler supports both instance trees and aggregate trees. The following are the benefits of instance trees over aggregate trees.

- ▶ An instance tree call profile is a visual representation of what your application actually did in response to a particular request. Instance trees have exact start times (such as 5:09:33 PM on Tuesday), as do the methods within them. To handle the large number of instance trees that can be generated for a heavily used server request, Diagnostics selects the most interesting instance trees to keep.
- ▶ An aggregate tree is an amalgamation of all of the instance trees that have occurred during that five-minute period. This ensures that no data is lost. As nothing is ever thrown away (it is all averaged together) this overcomes the potential for losing some interesting data.

Resolving Problems Using Instance and Aggregate Trees

Aggregate trees amalgamate data, but the visualized trees do not represent what actually occurred in the system. For example, an aggregate tree could show a cache being both hit and missed when in reality, for any particular invocation, only one of the two could have occurred. This can be confusing and even lead to misunderstandings when reports are passed on to those who do not understand Diagnostics.

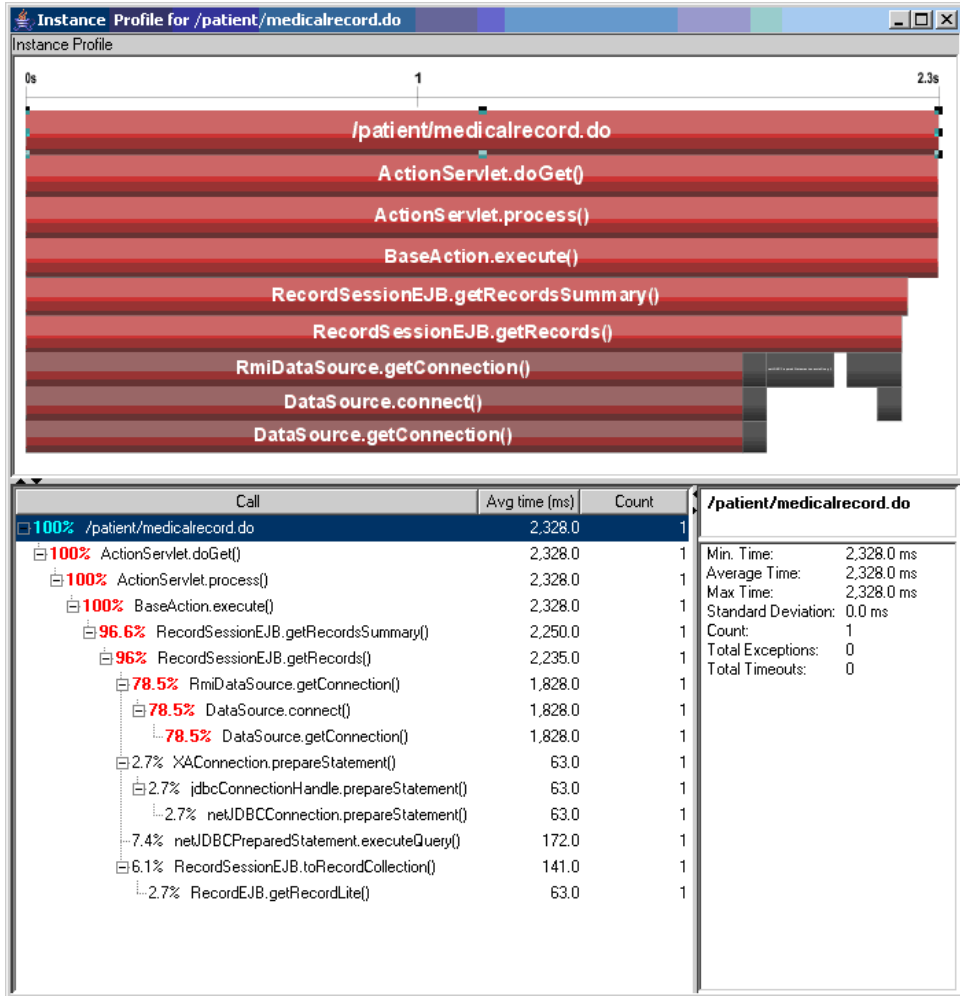
Another disadvantage is that while aggregate trees do well for general performance tuning (such as what occurs during a load test), they tend to hide information about relatively infrequent slow cases. Unfortunately, it is often these cases that cause problems for the customer.

The following examples from the Diagnostics Profiler which has both instance and aggregate trees demonstrate this problem:

First, consider the following two 'slowest' instance trees recorded on the medical records Java application.

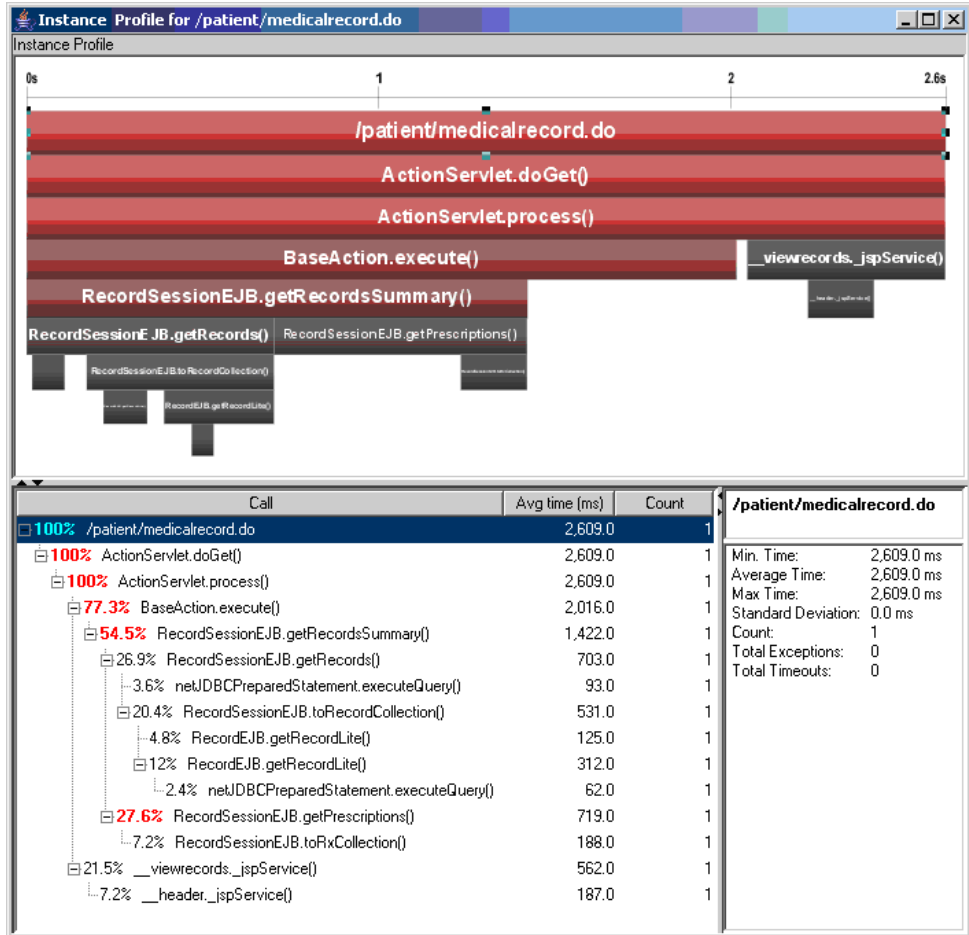
Example Instance Tree #1

The following slowest instance tree quite clearly shows that the majority of the 2.3 seconds spent responding to the `/patient/medicalrecord.do` server request was spent waiting for a JDBC connection to the database. (Is the application server's database connection pool too small for this workload?)



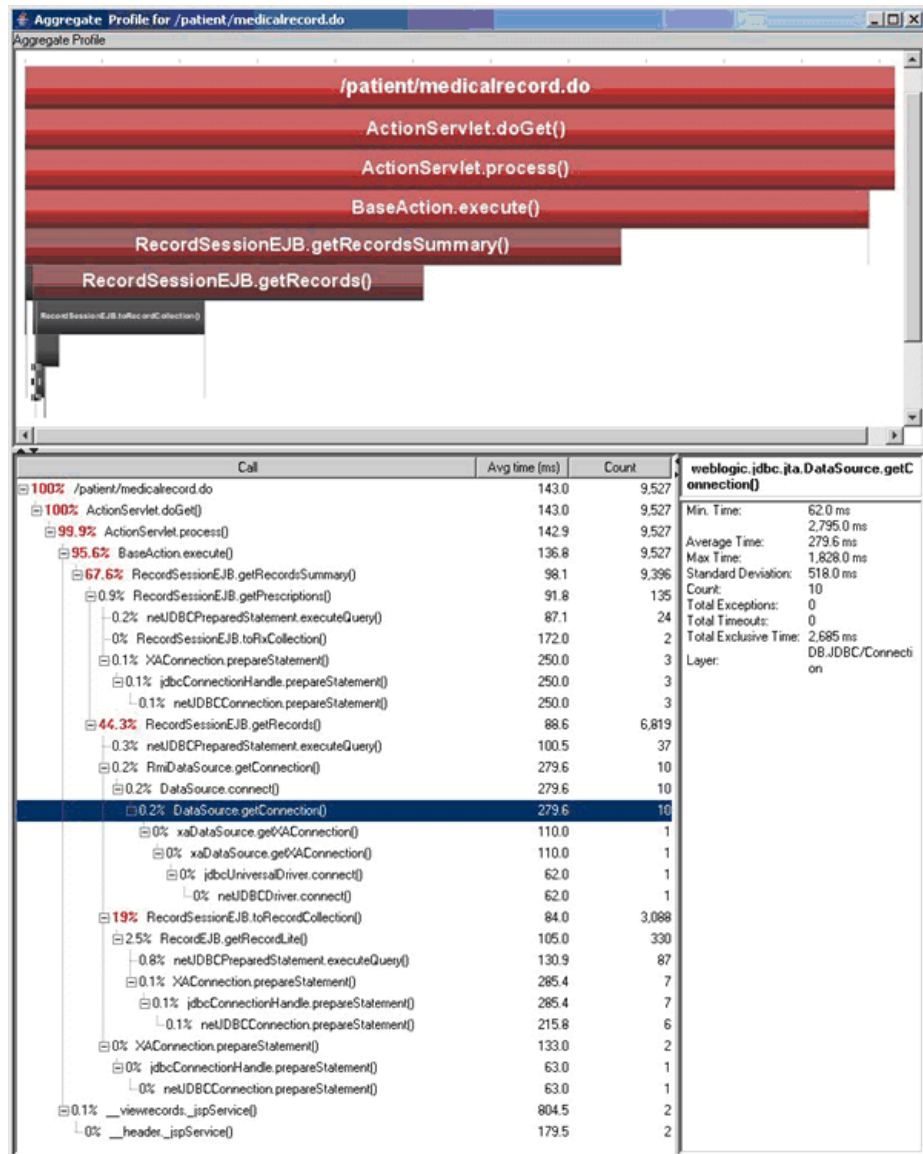
Example Instance Tree #2

This second slow instance tree from the same application shows a different call path: time spent reading records and prescriptions from the database. (Is the database overloaded? Are the SQL queries run by these methods optimal?)



Aggregate Tree Example

The instance trees in the previous two examples show distinctly different problems for the `/patient/medicalrecord.do` server request. The following aggregate tree was produced for the same server request:



Several things can be deduced from this example:

- The aggregate tree is much larger. As it combines multiple possible execution paths into a single tree, it is more difficult to understand what the application is doing.
- If you want to improve the overall performance of your application, the aggregate tree does indicate that you should concentrate on the common case (that the database connection was quickly retrieved from the pool).
- The aggregate tree does not identify that occasionally this server request stalls because it cannot obtain a database connection from the pool. **DataSource.getConnection()** contributes only 0.2%, approximately the same as other methods that have never caused a problem (such as **RecordEJB.getRecordLite**).

Part III

Diagnostics Standard Views

10

Server Summary View

This chapter describes the **Server Summary** view.

This chapter includes:

Concepts

- ▶ Server Summary Overview on page 282

Tasks

- ▶ How to Use the Server Summary View on page 285

Reference

- ▶ Server Summary User Interface on page 289

Concepts

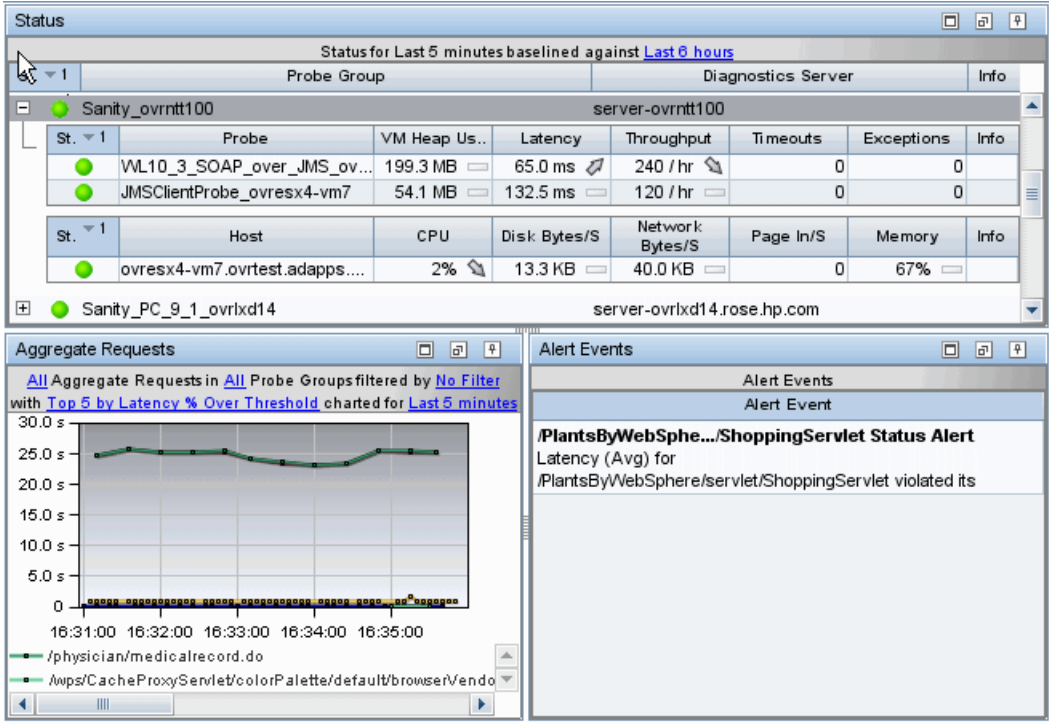
Server Summary Overview

The **Server Summary** view provides a high-level overview of your enterprise and enables you to easily check the status.

The Server Summary view is the default view for the Entire Enterprise application when Diagnostics is run by itself or is integrated with Business Service Management. If you selected an application, the Server Summary view gives a high-level overview of that application.

For details on the user interface, see "Server Summary View Description" on page 289.

The Server Summary view is a composite of the Status view, the Aggregate Requests graph, and the Alert Events list.



From the Server Summary **Status** pane you can see, at a glance, the overall performance across your enterprise. The Status table contains the probe groups that you defined when you installed the probes. Expanding the probe group displays the Probe table and Host table. If you've configured an integration with SiteScope (or external monitors) to forward data to Diagnostics you will also see a Monitors table.

By default the first column in the Status pane shows the status for each probe group, probe, host, monitor. The status reflects how the performance of the item compares with the threshold setting. The Status pane also shows key metrics and attributes for each item.

From the **Aggregated Requests graph**, you can see the trends of the aggregated server requests across your enterprise (or application).

The **Alert Events table** lists recent alerts that have been triggered. By default, the table of events in this view is sorted with the most recent alert events at the top. The list can contain a maximum of 20 events. For a complete list go to the Alerting view.

Tasks

How to Use the Server Summary View

If the metrics displayed in the Server Summary view raise any concerns, you can drill down to find out more information from a different view that depicts performance metrics at a lower level. For example, from a probe in the status table, you can drill down to server requests for that probe.

You can also take action to further analyze a performance problem. For example create a snapshot for later analysis.

You select drill down navigations and actions by right-clicking an item or in some cases you can double-click an item to navigate based on a pre-defined workflow. Diagnostics updates the available drill downs to allow only navigations that are appropriate for the selected entity.

This task includes the following steps:

- "Drilling Down to Detail Data from the Status Pane" on page 285
- "Displaying Tooltip Details From the Status Table" on page 287
- "Drilling Down to Detail Data From the Aggregate Requests Graph" on page 288
- "Displaying Server Request Details From the Charted Metric" on page 288
- "Drilling Down to Detail Data From An Alert Event" on page 288

Drilling Down to Detail Data from the Status Pane

From the Status table you can drill down into the details for the probe groups, probes and hosts that are behind the status that you see in the view. And you can take actions that are relevant to the entity selected.

Right-clicking a row in the Status table displays a pop-up menu of suggested actions you can take to further analyze a performance problem and diagnose the cause. The available actions and drill downs vary according to the type of probe group, probe or host.

When you double-click a row in the probe table, Diagnostics displays the Probes view for the selected probe.

When you double-click a row in the host table, Diagnostics displays the Hosts view for the selected host.

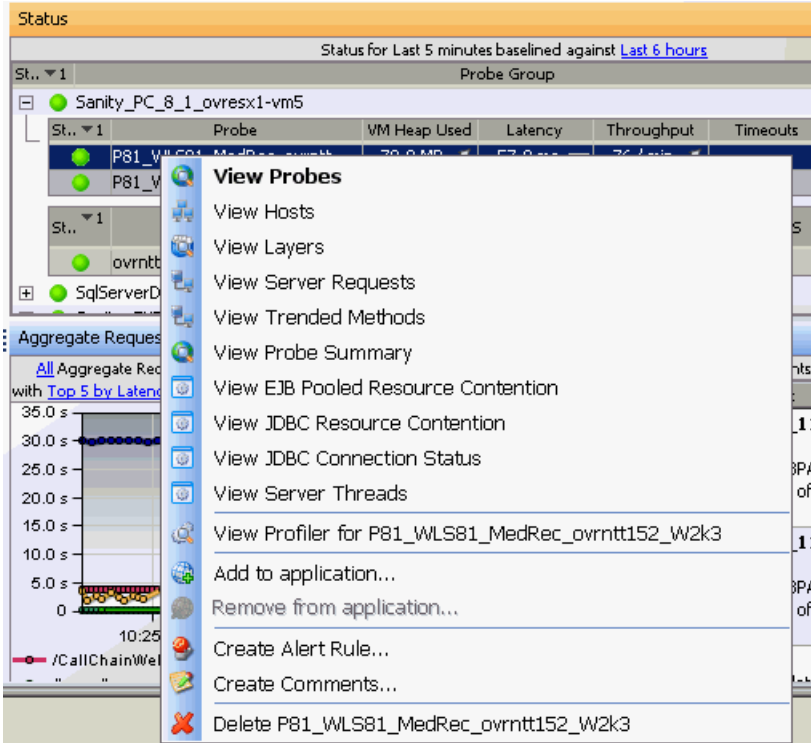
When you double-click a row in the monitors tables, the pre-defined workflow displays the Monitors view for the selected monitor.

The Probe Group Summary view and Probe Summary view are available navigations by right-click menu. These views are not available from the Standard Views, only by navigating from a status or summary view.

For details on the Probe Group Summary view or Probe Summary view, see "Server Summary User Interface" on page 289.

Example:

An example of the relevant navigations and actions for a probe is shown below.



Displaying Tooltip Details From the Status Table

You can view details for a probe group, probe or host listed in the Status table by holding your mouse pointer over that entry until the tooltip appears.

Drilling Down to Detail Data From the Aggregate Requests Graph

From the Aggregated Requests graph you can drill down to the Aggregate Requests view by double-clicking the graph or its legend. The Aggregate Requests view gives the full-featured view for the graph, rather than the summary.

Displaying Server Request Details From the Charted Metric

You can get additional details about a server request whose metric is charted in the graph by viewing the tooltip displayed in the Aggregate Requests graph.

Hold your mouse pointer over a point on the charted metric trend line that is between two of the round nodes on the line and a Server Request Details tooltip is displayed.

Hold your mouse pointer over a node on the line and additional information about the data used to point the point on the trend line is displayed in the tooltip.

Drilling Down to Detail Data From An Alert Event

From an alert event listed in the alert events table you can drill down to the Alerts view by double-clicking an event. If a threshold on a probe is violated, the drill down is to the Probes view.

Reference

Server Summary User Interface

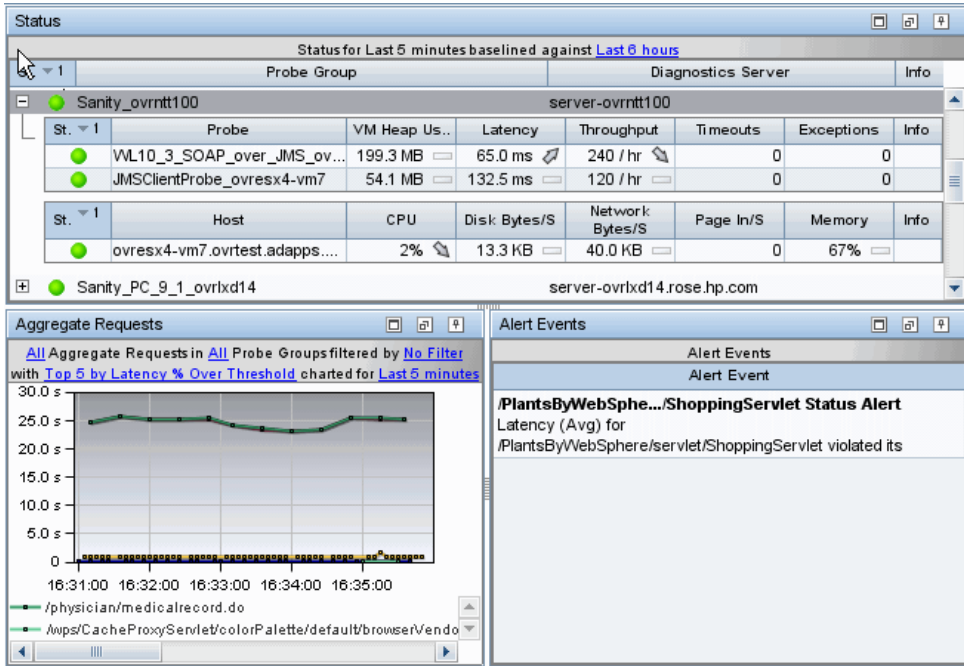
This section describes:


- ▶ [Server Summary View Description on page 289](#)
- ▶ [Scenario Summary View Description on page 297](#)
- ▶ [Business Process Summary View Description on page 300](#)
- ▶ [Probe Group Summary View Description on page 303](#)
- ▶ [Probe Summary View Description on page 309](#)

Server Summary View Description

The Server Summary view provides a high-level overview of your enterprise and allows you to easily check the status.

The following is an example of the Server Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views window opens.</p> <p>In the View bar select the Standard Views view group and click Server Summary.</p> <p>You can also access this view directly from the Diagnostics Applications window. To do this, select Entire Enterprise in the navigation pane. And then select the application summary link.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the Server Summary view. For more information about the ways you can control how performance metrics are presented in this view, see "Common Controls in the Diagnostics UI" on page 71.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Server Summary View" on page 285 for information on how to use data in this view to analyze performance.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ See "Status View" on page 515 ➤ See "Status Indicators" on page 517 ➤ See "Aggregate Request Overview" on page 530 ➤ See Chapter 5, "Working with Alerts"

Status Pane

The following user interface elements are included in the Status pane:

UI Elements	Description
<Title and metric trend indicators>	The title of the status table contains the baseline control. You use this control to set the time that Diagnostics uses as the baseline performance level with which to compare the current performance level. The result of the comparison of the baseline performance to the current performance is shown in the probe and host sub-tables by using the metric trend indicators (arrows that are displayed following the metric values).
Status	By default, the first column in the Status table presents the status for each probe group, probe, and host. The status reflects how the performance of the items compares with their performance thresholds.
<Probe Group Table>	<p>Lists the probe groups that you defined when you installed the agents and configured probes. Also shows the name of the Diagnostics server the probes in the probe group communicate with.</p> <p>Probe Group Details tooltip displays the probe group name.</p> <p>Within each probe group, there are several sub-tables for probes, hosts and monitors. You can expand and collapse these sub-tables.</p>

UI Elements	Description
<Probe Table>	<p>Lists all of the probes in the probe group along with the status.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>Probe Details tooltip for the probe column displays:</p> <ul style="list-style-type: none"> ➤ Probe. The name of the probe. ➤ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ➤ Host. The host on which the probe is running. ➤ Probe Type. Java, .NET, Oracle, SAP, MQ, SQL Server. <p>By default, the following metrics are displayed for each probe (the displayed columns are customizable):</p> <p>VM Heap Used. The average amount of heap used by the applications being monitored by the probe.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the application threw an exception.</p> <p>Info. This column indicates if the entity has an alert rule or custom comment defined.</p> <p>The result of the comparison of the baseline performance to the current performance is shown in the metric trend indicator arrows that are displayed following the metric values. A tooltip for each indicator gives a description of the current status. See "Status Indicators" on page 517 for a description of the status indicators.</p>

UI Elements	Description
<Hosts Table>	<p>Host Table. Lists the hosts for each of the probes in the probe group.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>The Host Details tooltip for the host column displays the fully qualified host name.</p> <p>By default, the following performance information is displayed for each host:</p> <p>CPU. Average CPU utilization (system metric)</p> <p>Disk Bytes/Second. The average number of bytes read and written per second.</p> <p>Network Bytes/Second. The average number of bytes sent and received per second.</p> <p>PageIn/Second. The average number of virtual memory pages swapped in per second.</p> <p>Memory. The average amount of memory in use.</p> <p>Info. This column indicates if the entity has an alert rule or custom comment defined.</p> <p>The result of the comparison of the baseline performance to the current performance is shown in the metric trend indicator arrows that are displayed following the metric values. A tooltip for each indicator gives a description of the current status. See "Status Indicators" on page 517 for a description of the status indicators.</p>

UI Elements	Description
<Monitor Table>	<p>Monitors Table. If you have configured an integration with SiteScope to forward data to Diagnostics, summary data for these external monitors is displayed in the monitor table. Lists the SiteScope monitors in the probe group. By default, the following performance information is displayed for each monitor:</p> <p>Type. Type of monitor.</p> <p>Collector. Collector responsible for this monitor (for example SiteScope).</p> <p>Info. This column indicates if the entity has an alert rule or custom comment defined.</p>

Aggregate Requests Graph

The following user interface elements are included in the Aggregate Requests Graph:

UI Elements	Description
<Title>	<p>The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.</p>
Aggregate Requests Graph	<p>Displays a graph of the average latency for aggregate requests as a trend line. By default, the five aggregated server requests with the highest latency percent over threshold during the previous five minutes are charted. You can use the view filters in the Title bar to filter the data displayed.</p> <p>The x-axis of the graph shows actual chronological time. The y-axis of the graph shows latency.</p> <p>Server Request Details tooltip displays information about the Server Request selected. Other tooltips are displayed for different types of server requests (such as Web Service Operation Details tooltip). Some of the information in the tooltip may include:</p> <ul style="list-style-type: none"> ▶ Server Request. The name of the selected server request. ▶ Probe Group. The name of the probe group to which the probe was assigned. ▶ Root Method. The method that originated from the server request. This method may be a portion of a URL. In the case of an RMI call, it may be a class and method representing the name of the server request itself. ▶ Package. The name of the package that contains the class from which the method was called. ▶ Method Signature. The signature of the root method. <p>The tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Double-clicking the graph displays the Aggregate Requests view.</p>

Alert Events List

The following user interface elements are included in the Alert Events List:

UI Elements	Description
Alert Events	The Alert Events table lists recent alerts that have been triggered. By default, the table of events in this view is sorted with the most recent alert events at the top. The list can contain a maximum of 20 events. For a complete list go to the Alerting view.

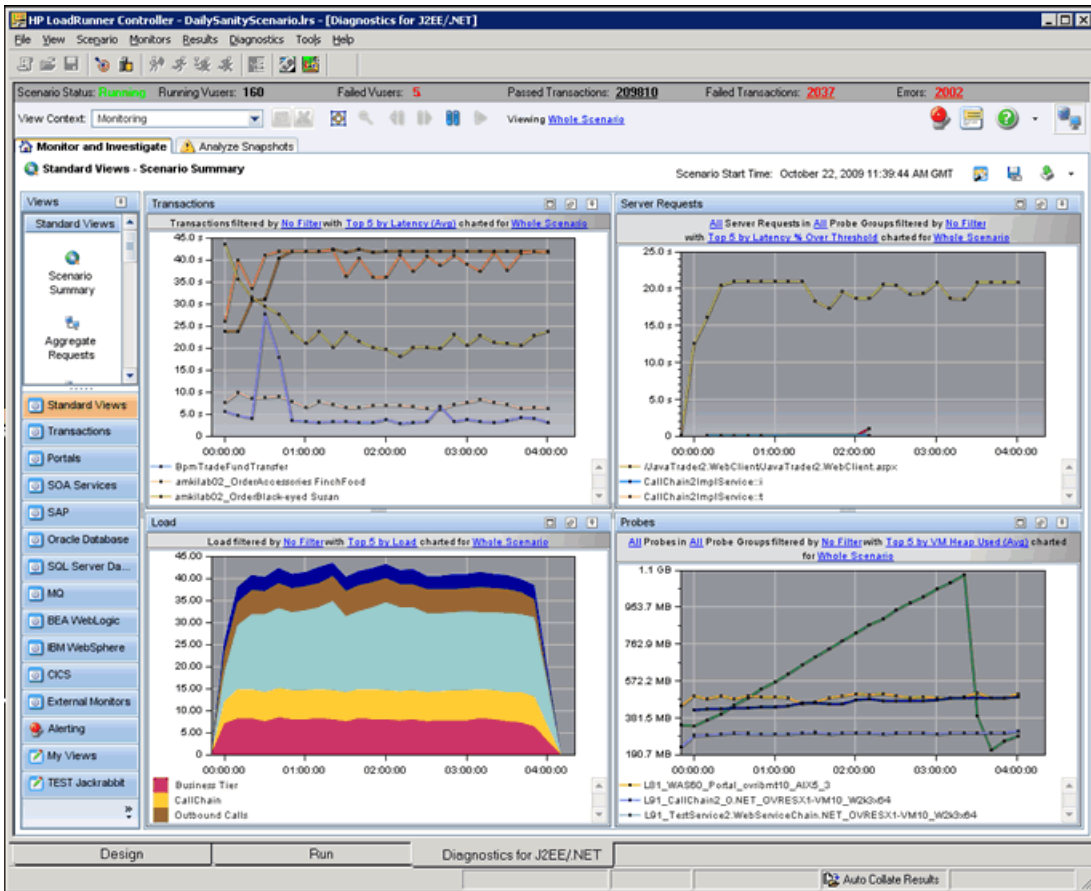
Scenario Summary View Description

This Scenario Summary view is displayed when you drill down for Diagnostics details, while running a scenario in HP LoadRunner or HP Performance Center. You will see the Scenario Summary view in the Diagnostics View bar instead of the Server Summary view when drilling down from LoadRunner or Performance Center.

The Scenario Summary view is a dashboard view that contains a summary of Diagnostics detail data for the scenario you are running. Monitoring versions (charts/graphs) of the following views are displayed:

- ▶ Transactions
- ▶ Server Requests
- ▶ Load
- ▶ Probes

The following is an example of the Scenario Summary view.



<p>To access</p>	<p>You access the Scenario Summary view when drilling down from HP LoadRunner or HP Performance Center.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Integration Between Diagnostics and LoadRunner" on page 1122 ➤ "Integration Between Diagnostics and Performance Center" on page 1124

User interface elements are described below:

UI Elements	Description
Transactions Graph	<p>A monitoring version of the Synthetic Transactions view.</p> <p>By default, the graph displays the top five transactions with the highest average latency, charted for the whole scenario.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Synthetic Transactions view.</p> <p>For information on the Synthetic Transactions view see "Synthetic Transactions View Description" on page 607.</p>
Server Requests Graph	<p>A monitoring version of the Server Requests view.</p> <p>By default, the graph displays the top five server requests with the highest latency percent over threshold, charted for the whole scenario. You can filter the data by sever request and probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Requests view.</p> <p>For information on the Server Requests view see "Server Requests View Description" on page 389.</p>

UI Elements	Description
Load Graph	<p>A monitoring version of the Load view.</p> <p>By default, the graph displays the load for the top five layers, charted for the whole scenario. You can filter the data by Layer Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average load for a layer.</p> <p>If you double-click the graph the pre-defined workflow displays the Load view.</p> <p>For information on the Server Requests view see "Load View Description" on page 511.</p>
Probes Graph	<p>A monitoring version of the Probes view.</p> <p>By default, the graph displays the top five probes with the highest average VM Heap Used, charted for the whole scenario. You can filter the data by Layer Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average VM Heap Used.</p> <p>If you double-click the graph the pre-defined workflow displays the Probes view.</p> <p>For information on the Probes view see "Probes View Description" on page 440.</p>

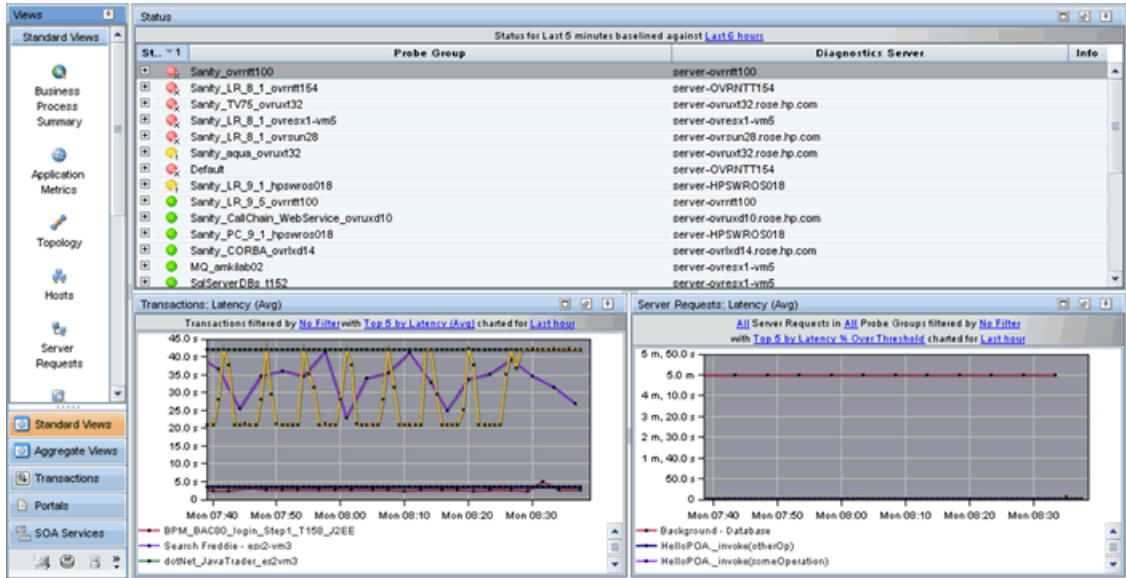
Business Process Summary View Description

This Business Process Summary view is displayed when you drill down for Diagnostics details from Business Process Monitor. You will see the Business Process Summary view in the Diagnostics View bar instead of the Server Summary view when drilling down from Business Process Monitor.

The Business Process Summary view is a dashboard view that contains a summary of Diagnostics detail data for the business process you selected in Business Process Monitor. Monitoring versions (charts/graphs) of the following views are displayed:

- Status view
- Transactions
- Server Requests

The following is an example of the Business Process Summary view.



To access	You access the Business Process Summary view when drilling down from Business Process Monitor.
-----------	--

User interface elements are described below:

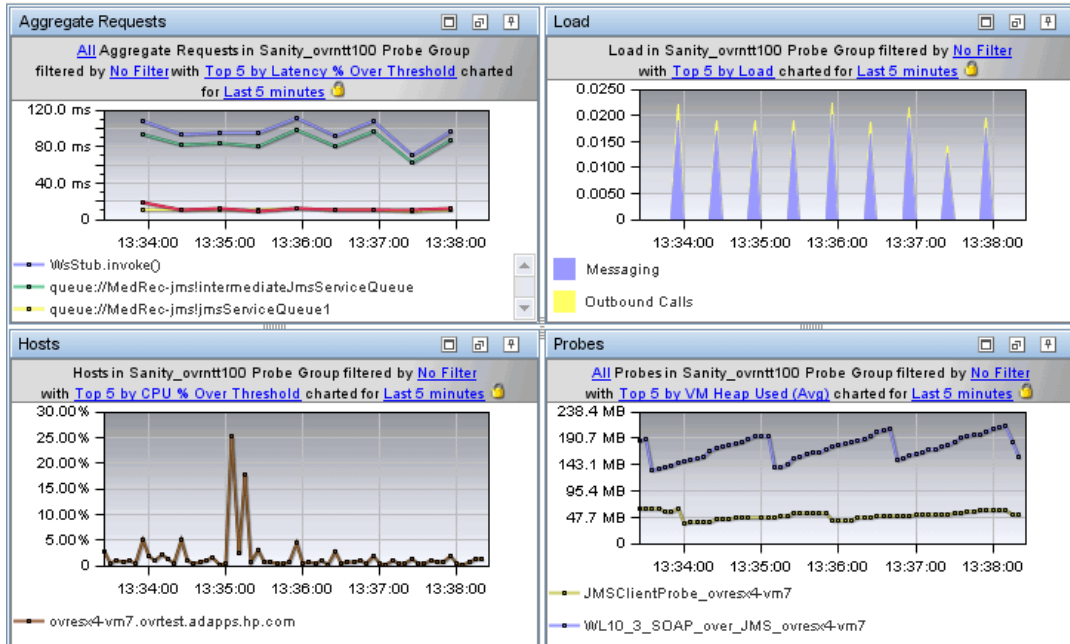
UI Elements	Description
Status view	<p>A monitoring version of the standard Diagnostics Status view.</p> <p>For more information about the Status view, see "Status View Description" on page 520.</p>
Transactions Graph	<p>A monitoring version of the Synthetic Transactions view.</p> <p>By default, the graph displays the top five transactions with the highest average latency, charted for the whole scenario. You can filter the data by Transaction Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Synthetic Transactions view.</p> <p>For information on the Synthetic Transactions view see "Synthetic Transactions View Description" on page 607.</p>
Server Requests Graph	<p>A monitoring version of the Server Requests view.</p> <p>By default, the graph displays the top five server requests with the highest latency percent over threshold, charted for the whole scenario. You can filter the data by Application Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Requests view.</p> <p>For information on the Server Requests view see "Server Requests View Description" on page 389.</p>

Probe Group Summary View Description

The Probe Group Summary view displays four panes with summary performance data for the selected probe group:

- Aggregate Requests graph
- Load graph
- Hosts graph
- Probes graph

The following is an example of the Probe Group Summary view.



<p>To access</p>	<p>The Probe Group Summary view cannot be accessed directly from the View bar. You can access the Probe Group Summary view by drilling down from a probe group in a Status table in any of the status or summary views.</p>
<p>Important information</p>	<p>The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Aggregate Request Overview" on page 530 ➤ "Load View" on page 507 ➤ "Hosts View" on page 369 ➤ "Probes View" on page 433

The following user interface elements are included:

UI Elements	Description
Aggregate Requests	<p>Displays a graph of average latency of aggregate requests, using a trend line. By default, the five aggregated server requests with the highest average latency percent over threshold for the probe group during the selected time period are charted.</p> <p>The x-axis of the graph shows actual chronological time. The y-axis of the graph shows latency.</p> <p>Server Request Details tooltip displays information about the Server Request selected. Other tooltips are displayed for different types of server requests (such as Web Service Operation Details tooltip). Some of the information in the tooltip may include:</p> <ul style="list-style-type: none"> ➤ Server Request. The name of the request. ➤ Probe Group. The name of the probe group. ➤ Root Method. The method that originated from the server request. This method may be a portion of a URL. In the case of an RMI call, it may be a class and method representing the name of the server request itself. ➤ Package. The name of the package that contains the class from which the method was called. ➤ Method Signature. The signature of the root method. <p>The tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Double-clicking the graph displays the Aggregate Requests view.</p>

UI Elements	Description
Load	<p>Displays a graph of the load for each layer where processing has taken place in the applications monitored under the selected probe group.</p> <p>The breakdown of the load across layers is shown in a stacked area graph. By default, the five layers that have the highest load during the previous five minutes are charted.</p> <p>The x-axis of the graph shows actual chronological time in hours, minutes, and seconds (hh:mm:ss). The y-axis of the graph shows the scale for the calculated load values.</p> <p>Layer Details tooltip for the area graph displays the layer name and probe group.</p> <p>Double-clicking the area graph displays the Load view.</p>

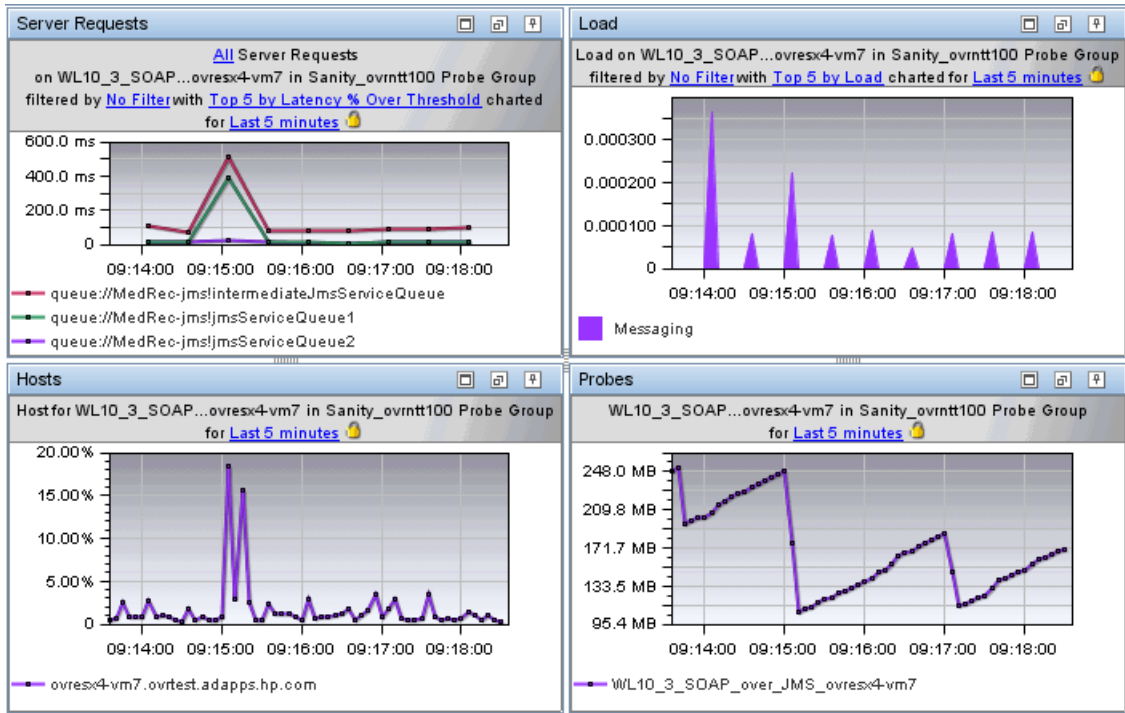
UI Elements	Description
<p>Hosts</p>	<p>Displays a graph of the average CPU for hosts, as a trend line. By default, the five hosts with the highest CPU percent over threshold during the previous five minutes are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds.</p> <p>Host Details tooltip for the trend line in the graph displays the fully qualified name of the host.</p> <p>If you hover the mouse over a point on the trend line the tooltip displays additional information used to plot the selected node such as sample start and end times.</p> <p>Double-clicking the graph displays the Hosts view.</p>
<p>Probes</p>	<p>Displays a graph of the average VM heap used, as a trend line. By default, the five probes that have the highest average heap usage during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the heap usage amount in megabytes (mb) or other related units, if more appropriate.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <ul style="list-style-type: none"> ➤ Probe. The name of the probe collecting the metrics represented. ➤ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ➤ Host. The host on which the probe is running. ➤ Probe Type. Java, .NET, Oracle, SAP, MQ, SQL Server. <p>The Probe Details tooltip for nodes on the trend line display additional information used to plot the selected node such as sample start and end times.</p> <p>Double-clicking the graph displays the Probes view.</p>

Probe Summary View Description

Displays four panes with summary performance data for the selected probe:

- Server Requests graph
- Load graph
- Hosts graph
- Probes graph

The following is an example of the Probe Summary view.



<p>To access</p>	<p>The Probe Summary view cannot be accessed directly from the View bar. You can access the Probe Summary view by drilling down from a probe selected in a status table in any of the status or summary views</p>
<p>Important information</p>	<p>The view titles in the different panes describe the view using the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Server Requests View" on page 379 ➤ "Load View" on page 507 ➤ "Hosts View" on page 369 ➤ "Probes View" on page 433

The following user interface elements are included:

UI Elements	Description
Server Requests	<p>Displays a graph of average latency, as a trend line. By default, the five server requests that have the highest average latency during the selected time range are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>Server Request Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself.</p> <p>Type. The Server Request type. Examples of server request types include: ADO, CICS, Corba, HTTP, JDBC, JMS, MSMQ, RMI, Remoting (.NET), SAP ABAP types, Web Services. In addition, you may see server request types named Pseudo and RootRename.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Deployed Into. Tells where the web service was deployed into (usually an EAR file).</p> <p>Method Signature. The signature of the root method.</p> <p>Double-clicking the graph displays the Server Requests view.</p>

UI Elements	Description
Load	<p>Displays a graph of the load for each layer where processing has taken place in the applications monitored by the selected probe.</p> <p>The breakdown of the load across layers is shown in a stacked area graph. By default, the five layers that have the highest load during the previous five minutes are charted.</p> <p>The x-axis of the graph shows actual chronological time in hours, minutes, and seconds (hh:mm:ss). The y-axis of the graph shows the scale for the calculated load values.</p> <p>Layer Details tooltip for the area graph displays the layer name, probe and probe group.</p> <p>Double-clicking on the area graph displays the Load view.</p>

UI Elements	Description
<p>Hosts</p>	<p>Displays a graph of the average CPU for hosts, as a trend line. By default, the five hosts with the highest CPU percent over threshold during the previous five minutes are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds.</p> <p>Host Details tooltip for the trend line in the graph displays the fully qualified name of the host.</p> <p>If you hover the mouse over a point on the trend line the tooltip displays additional information used to plot the selected node such as sample start and end times.</p> <p>Double-clicking the graph displays the Hosts view.</p>
<p>Probes</p>	<p>Displays a graph of the average VM heap used, as a trend line. By default, the five probes that have the highest average heap usage during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the heap usage amount in megabytes (mb) or other related units, if more appropriate.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <ul style="list-style-type: none"> ➤ Probe. The name of the probe collecting the metrics represented. ➤ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ➤ Host. The host on which the probe is running. ➤ Probe Type. Java, .NET, Oracle, SAP, MQ, SQL Server. <p>The Probe Details tooltip for nodes on the trend line display additional information used to plot the selected node such as sample start and end times.</p> <p>Double-clicking the graph displays the Probes view.</p>

11

Application Explorer View

This chapter describes the Application Explorer view.

This chapter includes:

Concepts

- ▶ Application Explorer Overview on page 316

Tasks

- ▶ How to Use the Application Explorer on page 319

Reference

- ▶ Application Explorer Available Metrics by Entity Type on page 324
- ▶ Application Explorer User Interface on page 326

Concepts

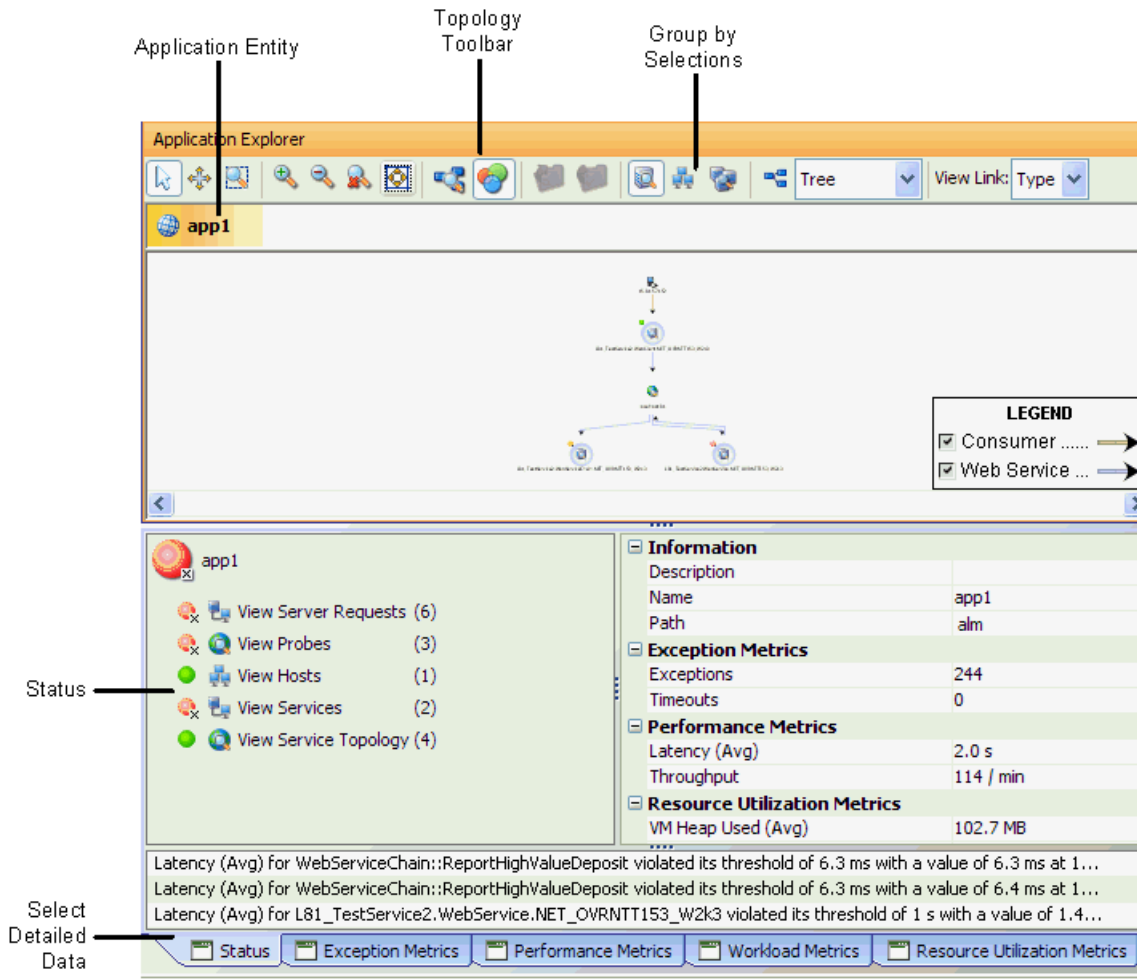
Application Explorer Overview

The Application Explorer gives you a topological view of your application, and a set of tabs showing data for various aspects of the selected entity's performance. As you select different entities in the topology, the data is updated to reflect the current selection.

Note: If there are no probes or collectors associated with an application, then in the Application Explorer, you won't see a topology diagram. Instead you will see a message about the topology view. An application must have a probe in it in order to draw a topology.

For details on the user interface, see "Application Explorer View Description" on page 326.

An example of the Application Explorer view is shown below:



Application Entity. When you enter the Application Explorer, the application entity is selected by default in the topology (the application is shown at the top, above the topology). The entity selected in the topology determines what is displayed in the lower panes.

For more information about how applications entities are defined in Diagnostics, see Chapter 2, "Working with Applications".

Topology. The topology provides a visual representation of your application. You can use the topology toolbar to control the appearance of the topology by scrolling and zooming, and by toggling the appearance.

For information on topology diagrams, see Chapter 8, "Working with Topologies".

The types of entities that may be selected in the topology are as follows:

- Application
- Probe group
- Host
- Probe (Java, .NET, SAP NetWeaver-ABAP collector)
- Collector (Oracle, SQL Server, MQ)
- Connections

Other entities (like consumer) are shown in the topology but cannot be selected.

Status. You can also see the status for an entity in the topology, represented by a green, yellow, red or grey icon.

For a description of the status indicators, see "About Status Indicators" on page 110.

Select Detail Data. You can use the tabs at the bottom of the display to access performance information for the entity selected. The tabs available will vary slightly based on the entity you select in the topology.

Tasks

How to Use the Application Explorer

The Application Explorer gives you a topological view of your application, and a set of tabs showing data for various aspects of the selected entity's performance. As you select different entities in the topology, the data is updated to reflect the current selection.

You can use the Application Explorer as the starting point to determine how your application is performing. For example, if some aspect of an application is taking longer to complete than expected, you can start with the Application Explorer and examine each entity in the application to locate the performance issues. If you find that a server request has unusually high latency then you can drill into more details, for that request.

You can navigate to all other relevant views from within the Application Explorer or you can select another view from the View bar.

This task includes the following steps:

- "Drilling Down from the Entity Status List" on page 319
- "Drilling Down from the Status Summary" on page 320
- "Drilling Down from the Different Metrics Tabs" on page 322

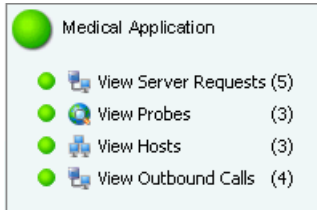
Drilling Down from the Entity Status List

When you select the Status tab in the Application Explorer, the pane on the lower left shows the status of the entity selected in the topology. And under this you'll see the entities associated with the selected entity and their status.

The relevant navigations for each of these entities is available as a link that allows you to navigate to a detail view (for example, View Hosts or View Layers).

Example:

For example you may see the following when an application entity is selected. This indicates that for the Medical Application you should see three probes, three hosts, five server requests and four outbound calls.



Drilling Down from the Status Summary

When you select the Status tab in the Application Explorer, the pane on the lower right shows summary information for the entity selected in the topology. Threshold violations for a metric (such as Latency (Avg)) are shown in red or yellow in the summary table. This is designed to give you a high level view of the performance metrics for the entity selected.

Each of the summary groups (Exception Metrics, Performance Metrics) is associated with a different tab at the bottom of the screen. You can select the tab associated with the summary group to drill down to more details.

Example:

For example the Performance Metrics summary group shows Latency and Throughput for the selected application. Selecting the Performance Metrics tab at the bottom of the display gives you performance metrics trend graphs for the entity selected.

The screenshot displays the Application Explorer View for a 'Medical Application'. The left sidebar shows a tree view with the following items:

- Medical Application
- View Server Requests (5)
- View Probes (3)
- View Hosts (3)
- View Outbound Calls (4)

The main content area shows a summary table with the following data:

Information	
Description	
Name	Medical Application
Path	mycorp
Exception Metrics	
Exceptions	0
Timeouts	0
Performance Metrics	
Latency (Avg)	52.1 ms
Throughput	109 / min
Resource Utilization Metrics	
VM Heap Free (Avg)	18.3 MB

Below the table, it states 'No Alert Events available.' At the bottom, there is a tabbed interface with the following tabs: Status, Exception Metrics, Performance Metrics, Workload Metrics, and Resource Utilization... The 'Performance Metrics' tab is currently selected and highlighted with a red box.

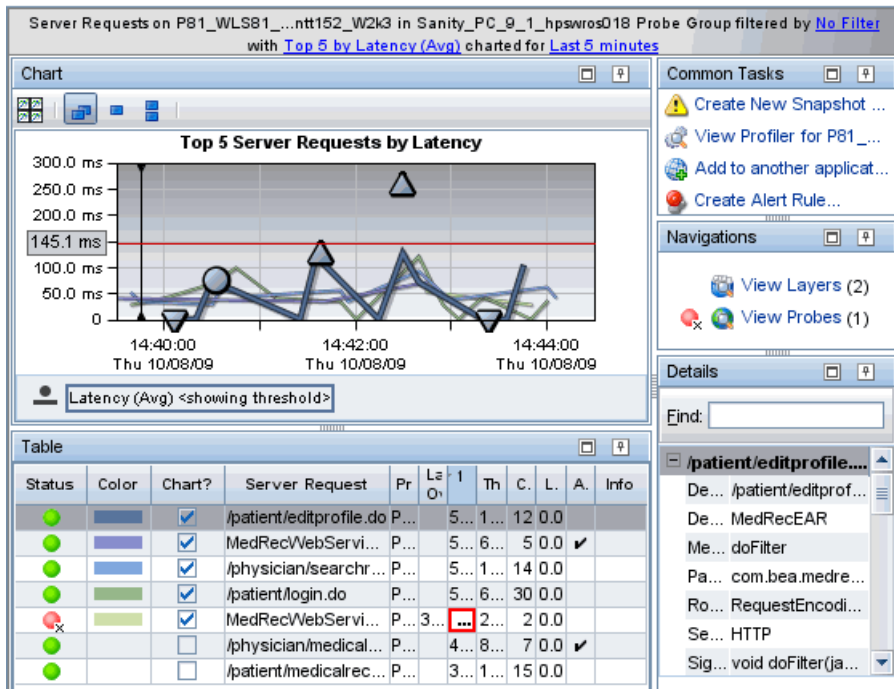
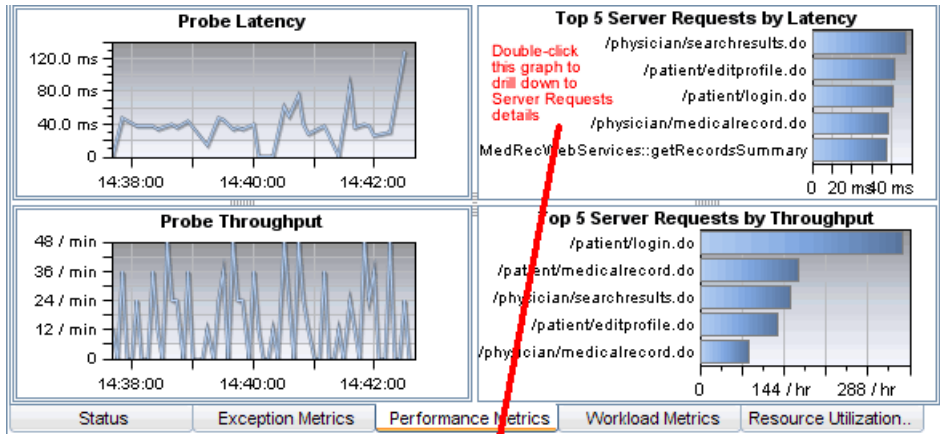
Drilling Down from the Different Metrics Tabs

When you double-click any of the graphs in the Performance Metrics, Workload Metrics or Resource Utilization tabs, a detail view is displayed filtered on the entity you selected in the topology.

For a drill down from probe to server request details, you will see all server requests active in the probe. They may or may not be assigned to the application you had selected. The App? column in the drilled to table will be checked to indicate if the server request is part of the application.

Example:

Select a probe in the topology and then select the Performance Metrics tab. Double-click the Top 5 Server Requests by Latency graph. This drill down displays server request details for the selected probe, with the top 5 server requests charted.



Reference

Application Explorer Available Metrics by Entity Type

The following table shows the metrics available for each entity type in the Performance Metrics tab, Workload Metrics tab and Resource Utilization tab.

Entity Type	Performance Metrics	Workload Metrics	Resource Utilization Metrics
Application entity:	Throughput Top N Server Requests by Throughput Latency by layers Top N Server Requests by Latency	Count Top N Server Requests by Count Load Top N Server Requests by Load	CPU (Avg) Heap Total/Used Disk bytes/sec Network bytes/sec
Probe (Java and .NET) entity:	Throughput Top N Server Requests by Throughput Latency Top N Server Requests by Latency	Count Top N Server Requests by Count Load Top N Server Requests by Load	CPU (Avg) Heap Total/Used
Probe (ABAP) entity:	Throughput Top N Server Requests by Throughput Latency Top N Server Requests by Latency	Count Top N Server Requests by Count Load Top N Server Requests by Load	CPU (Avg) Extended Memory Private Memory

Entity Type	Performance Metrics	Workload Metrics	Resource Utilization Metrics
Host entity:	N/A	N/A	CPU (Avg) Memory, Virtual Memory Disk bytes/sec Network bytes/sec
Probe Group entity:	Throughput Top N Aggregated Server Requests by Throughput Latency Top N Aggregated Server Requests by Latency	Count Top N Aggregated Server Requests by Count Load Top N Aggregated Server Requests by Load	N/A
Probe Connection entity (when connection has an outbound call):	Throughput Top N Outbound Calls by Throughput Latency Top N Outbound Call by Latency	Count Top N Outbound Calls by Count Load Top N Outbound Calls by Load	N/A
Probe connection entity (when connection is only an inbound probe call):	Throughput Top N Server Requests by Throughput Latency Top N Server Requests by Latency	Count Top N Server Requests by Count Load Top N Server Requests by Load	N/A
Oracle collector entity:	User Transaction Per Sec Executions Per Sec	Load by layers Load by wait time	DB wait time ratio DB CPU time ratio CPU usage per sec CPU usage per txn

Entity Type	Performance Metrics	Workload Metrics	Resource Utilization Metrics
SQL Server collector entity:	Database:Transaction s/sec	Load by wait time	CPU Busy Time (Avg) IO Busy Time (Avg) Idle Time (Avg)
MQ collector entity:	N/A	Messages Transferred/sec (Avg) Top N Channels by Messages Transferred/sec Depth (Avg) Top N Channels by Depth	N/A

Application Explorer User Interface

This section describes:

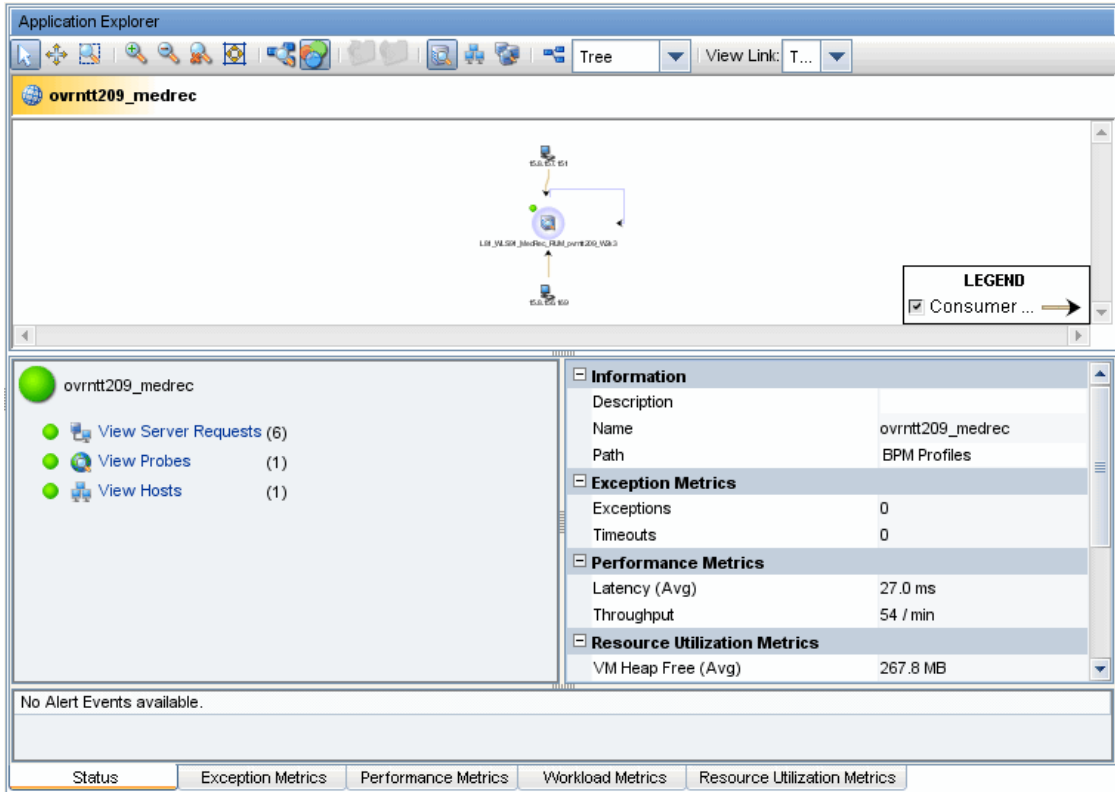
- [Application Explorer View Description on page 326](#)


Application Explorer View Description

The Application Explorer gives you a topological view of your application, and a set of tabs showing data for various aspects of the selected entity's performance. As you select different entities in the topology, the data is updated to reflect the current selection.

You can use the tabs at the bottom of the display to access performance information for the entity selected. The tabs available will vary slightly based on the entity you select in the topology.

The following is an example of the Application Explorer view.



<p>To access</p> 	<ul style="list-style-type: none"> ➤ In the navigation pane of the Diagnostics Applications window, double-click an application name from the Applications groups. The Diagnostics views open with the Application Explorer displayed. This is the default view for an application. ➤ Alternately, from the Applications window you can select an application and then select Open application explorer and this will display the Application Explorer. ➤ Or, from the View bar, click Standard Views to open the Standard Views view group. In the Standard Views view group, click Application Explorer.
<p>Important information</p>	<p>The Application Explorer view is not available if you select the Entire Enterprise context in the Applications window.</p> <p>When you select the application entity in the topology, the time period shown is always five minutes. For other entities you can select a time period.</p> <p>The metrics shown in the Application Explorer tables will vary depending on what type of entity you've selected in the topology. See "Application Explorer Available Metrics by Entity Type" on page 324 for details.</p>
<p>Relevant tasks</p>	<p>You can navigate to all other relevant views from within the Application Explorer or you can select another view from the View Bar.</p> <p>See "How to Use the Application Explorer" on page 319 for information on how to analyze performance using this view.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Working with Applications" on page 45 ➤ Chapter 8, "Working with Topologies" ➤ "About Status Indicators" on page 110

When you enter the Application Explorer, the application entity is selected by default in the topology (the application is shown at the top, above the topology). The entity selected in the topology determines what is displayed in the lower panes.

The topology provides a visual representation of your application. The following user interface elements are included:

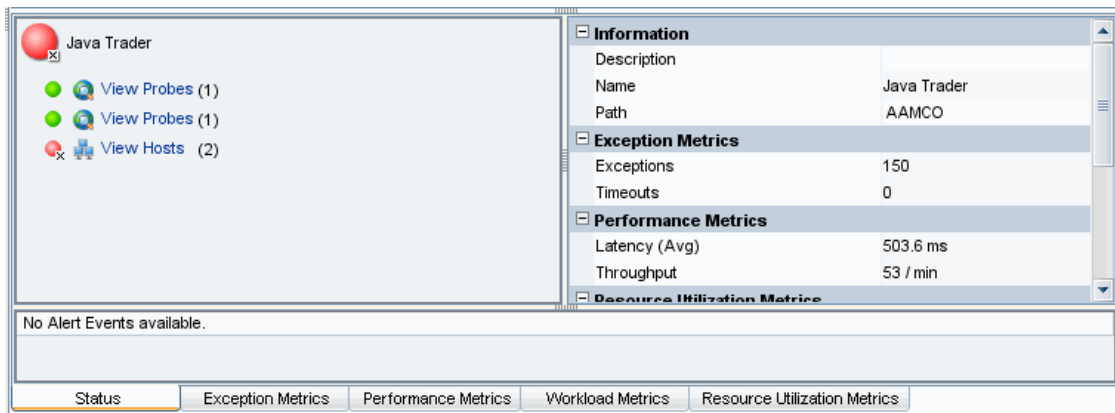
UI Elements	Description
<topology toolbar>	You can use the topology toolbar to control the appearance of the topology by scrolling and zooming, and by toggling the appearance.
<application>	When you enter the Application Explorer, the application entity is selected by default in the topology (the application is shown at the top, above the topology). The entity selected in the topology determines what is displayed in the lower panes.
<topology>	<p>The topology provides a visual representation of your application.</p> <p>Entities with a blue 'halo' are selectable in the topology. When an entity is selected, it is displayed with a yellow halo.</p> <p>The types of entities that may be selected in the topology are as follows:</p> <ul style="list-style-type: none"> ➤ Application ➤ Probe group ➤ Host ➤ Probe (Java, .NET, SAP NetWeaver-ABAP collector) ➤ Collector (Oracle, SQL Server, MQ) ➤ Connections <p>Other entities (like consumer) are shown in the topology but cannot be selected.</p> <p>The colored dot next to an entity in the topology indicates the status, which is based on a variety of factors, including metric thresholds.</p> <p>The connection arrows represent aggregated calls to applications within your environment. Connections can be shown between various entities such as probes, probe groups, Web services and consumers, or targets. The arrow on the connection indicates the direction of the communication.</p>

UI Elements	Description
<legend>	The legend gives you details on the type of connections shown in the topology diagram. You can show/hide the legend with the Diagram Legend button on the toolbar.
<tabs>	You can use the tabs at the bottom of the display to access performance information for the entity selected. The tabs available will vary slightly based on the entity you select in the topology.

Status Tab

The Status tab is the default displayed when you select the sApplication Explorer view.

The following is an example of the Application Explorer Status tab.



The following user interface elements are included:

UI Elements	Description
<status bulb>	<p>The status bulb next to an entity indicates the status, which is based on a variety of factors, including metric thresholds.</p> <p>Status is only available when you view time ranges that are an hour or less. Status is shown as gray when no status is available, or when the time range exceeds an hour.</p> <p>See "About Status Indicators" on page 110 for an explanation of how status is calculated.</p>
<associated entities>	<p>The pane on the lower left shows the status of the entity selected in the topology. And under this you'll see the entities associated with the selected entity and the relevant navigations for each.</p>
<summary metrics>	<p>The pane on the lower right shows summary information for the entity selected in the topology. You can expand and collapse each summary group. Threshold violations for a metric (such as Latency (Avg)) are shown in red or yellow in the summary.</p> <p>Each of the summary groups is associated with a different tab at the bottom of the screen. Select the relevant tab (for example Performance Metrics) to get more details.</p> <p>The metrics shown will vary depending on what type of entity you've selected in the topology. See "Application Explorer Available Metrics by Entity Type" on page 324 for details.</p>
<alert events>	<p>Alerts for the selected entity are shown at the bottom of the display. Only the last three alerts for the entity selected are shown. The alerts are not cumulative for all related entity, just for the selected entity.</p>

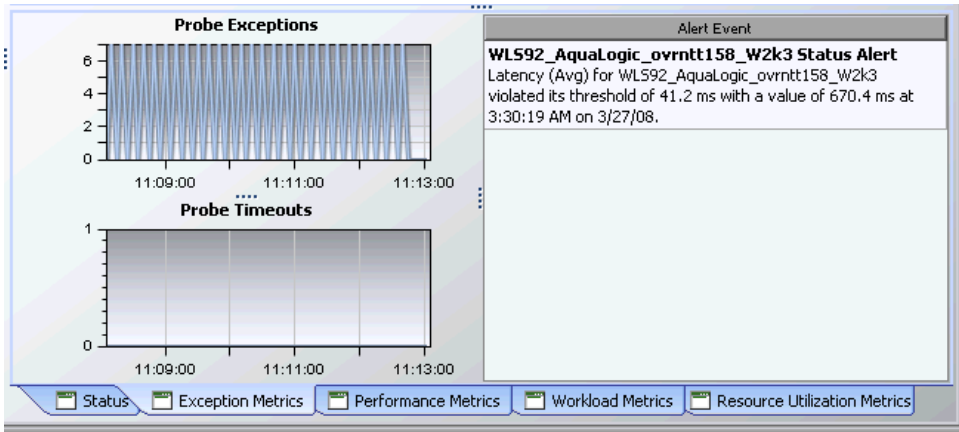
Exception Metrics Tab

To get more detailed information on exceptions you can select the Exception Metrics tab.

The Exceptions Metrics tab is available when you select the following types of entities: applications, probes (Java and .NET), probe (ABAP), probe group and probe connection. No Exceptions Metrics tab is available when a host, or collector entity (Oracle, SQL Server, MQ) is selected in the topology.

If there are no exceptions, the graphs on this tab will show a zero trend line.

The following is an example of the Application Explorer Status tab.



The following user interface elements are included:

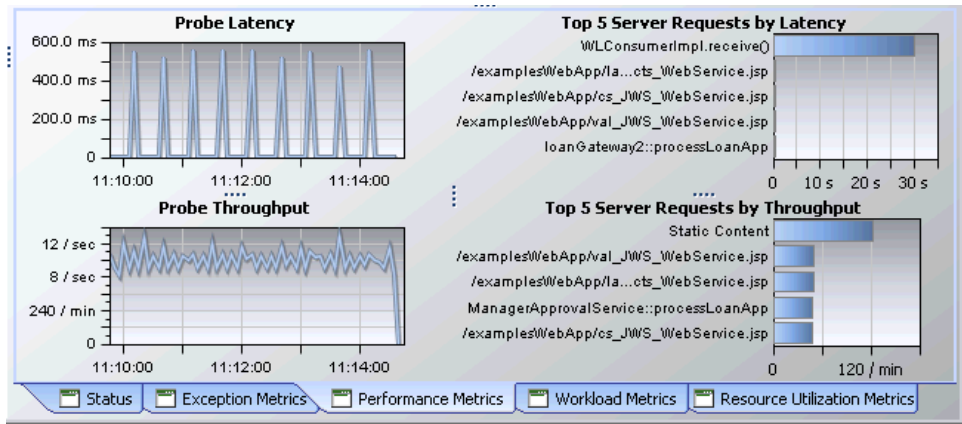
UI Elements	Description
<exceptions graph>	Exceptions graph shows a trend line of the number of exceptions for the selected entity over time. Double-clicking the graph allows you to drill down to more detail.
<timeouts graph>	Timeouts graph shows a trend line of the number of timeouts for the selected entity over time. Double-clicking the graph allows you to drill down to more detail.
<list of alerts>	Lists any alerts for the selected entity.

Performance Metrics Tab

To get more detailed information on performance you can select the Performance Metrics tab. The lower pane then displays additional data.

The specific data varies depending on the type of entity selected. See "Application Explorer Available Metrics by Entity Type" on page 324.

The following is an example of the Application Explorer Status tab.



The following user interface elements are included:

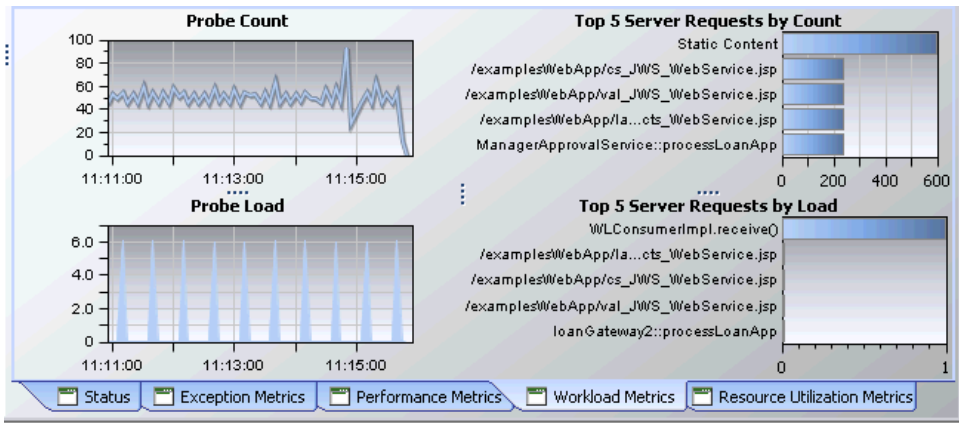
UI Elements	Description
<trend graphs>	<p>Trend graphs show data such as latency or throughput over time for a selected entity.</p> <p>Double-clicking a chart allows you to drill down to more detail.</p>
<top N bar charts>	<p>Top 'N' bar charts show, for example, the five server requests with the highest latency or throughput.</p> <p>If you select a connection in the Application Explorer topology, the bar graphs show Top 5 Calls by <metric> allowing display of both outbound and inbound calls.</p> <p>Double-clicking a chart allows you to drill down to more detail.</p>

Workload Metrics Tab

To get more detailed information on workload you can select the Workload Metrics tab. The lower pane then displays additional data.

The specific data varies depending on the type of entity selected. See "Application Explorer Available Metrics by Entity Type" on page 324.

The following is an example of the Application Explorer Status tab.



The following user interface elements are included:

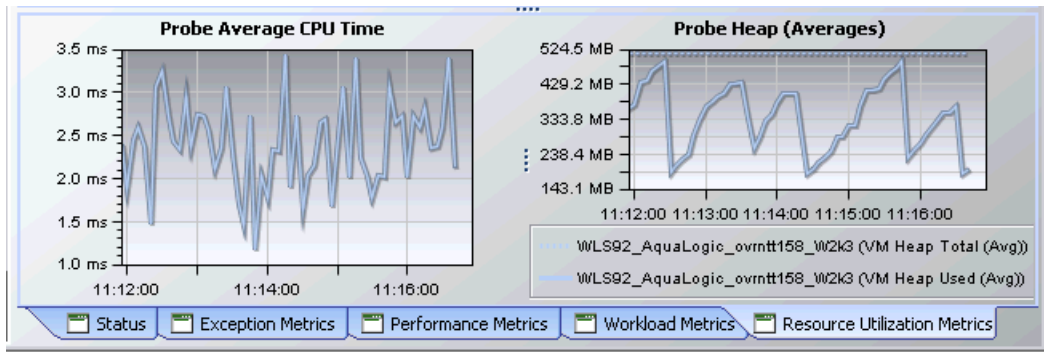
UI Elements	Description
<trend graphs>	Trend graphs show data such as count or load over time for a selected entity. Double-clicking a graph allows you to drill down to more detail.
<top N bar charts>	Top 'N' bar charts show, for example, the five server requests with the highest count or load. Double-clicking a chart allows you to drill down to more detail.

Resource Utilization Metrics Tab

To get more detailed information on resource utilization you can select the Resource Utilization Metrics tab. The lower pane then displays additional data.

The specific data varies depending on the type of entity selected. See "Application Explorer Available Metrics by Entity Type" on page 324.

The following is an example of the Application Explorer Status tab.



The following user interface elements are included:

UI Elements	Description
<trend graphs>	Trend graphs show data such as Average CPU Time or Heap Total/Used over time for a selected entity. Double-clicking a graph allows you to drill down to more detail.

12

Application Metrics View

This chapter describes the Application Metrics view.

This chapter includes:

Concepts

- ▶ Application Metrics Overview on page 338

Tasks

- ▶ How to Use the Application Metrics View on page 339

Reference

- ▶ Application Metrics User Interface on page 341

Concepts

Application Metrics Overview

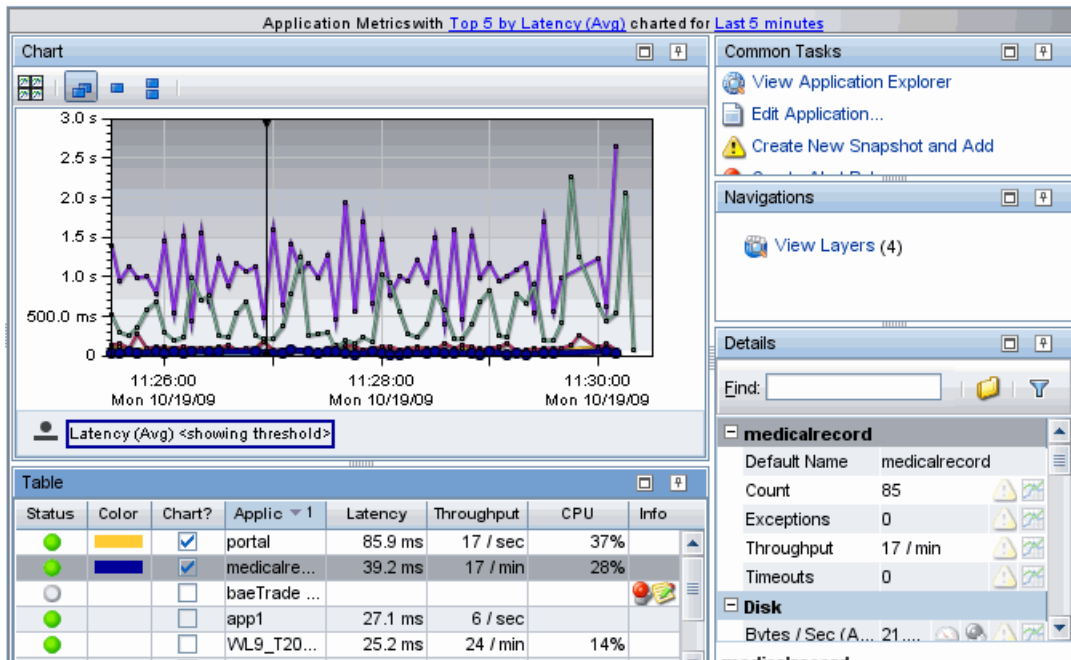
The **Application Metrics** view contains performance metrics for the applications you are monitoring. By default, the five applications that have the highest latency are charted in the graph.

The Application Metrics view gives you the ability to look at the resources used by the entire application in a rolled up fashion.

For details on the user interface, see "Application Metrics View Description" on page 341.

By default, the Application Metrics graph charts the average latency for the five applications that have the highest latency during the previous hour.

An example of the Application Metrics view is shown below:



Tasks

How to Use the Application Metrics View

If the metrics displayed in the Application Metrics view raise any concerns, you can drill down to find out more information from a view that depicts performance metrics at a lower level. For example, from an application, you can drill down to the Layers view to determine the application layer with the performance problem.

You can also take action to further analyze a performance problem. For example create a snapshot for later analysis.

You select drill down navigations and actions from the common tasks and navigations panes or by right-clicking an item and selecting from the pop-up menu, or in some cases you can double-click an item to navigate based on a pre-defined workflow. Diagnostics updates the available drill downs to allow only navigations that are appropriate for the selected entity.

This task includes the following steps:

- "Drilling Down to Application Layers from the Application Metrics View" on page 339
- "Editing an Application" on page 340

Drilling Down to Application Layers from the Application Metrics View

You can go to the application Layers view by:

- double-clicking the row for the application in the graph entity table.
- double-clicking a trend line for the application.
- clicking **View Layers** in the Navigations Pane.

In the Layers view you can see the latency contribution from the different layers in the application.

Editing an Application

You can edit an application in the Application Metrics View by:

- ▶ right-clicking the application's row in the graph entity table, and selecting **Edit Application** from the menu.
- ▶ clicking **Edit Application** in the Common Tasks Pane.

Diagnostics displays the Edit application dialog box. See "Creating Application Entities in Diagnostics" on page 47 for details.

Caution: You will only be able to edit an application if the user you are logged in as, has permissions for editing.

Reference

Application Metrics User Interface

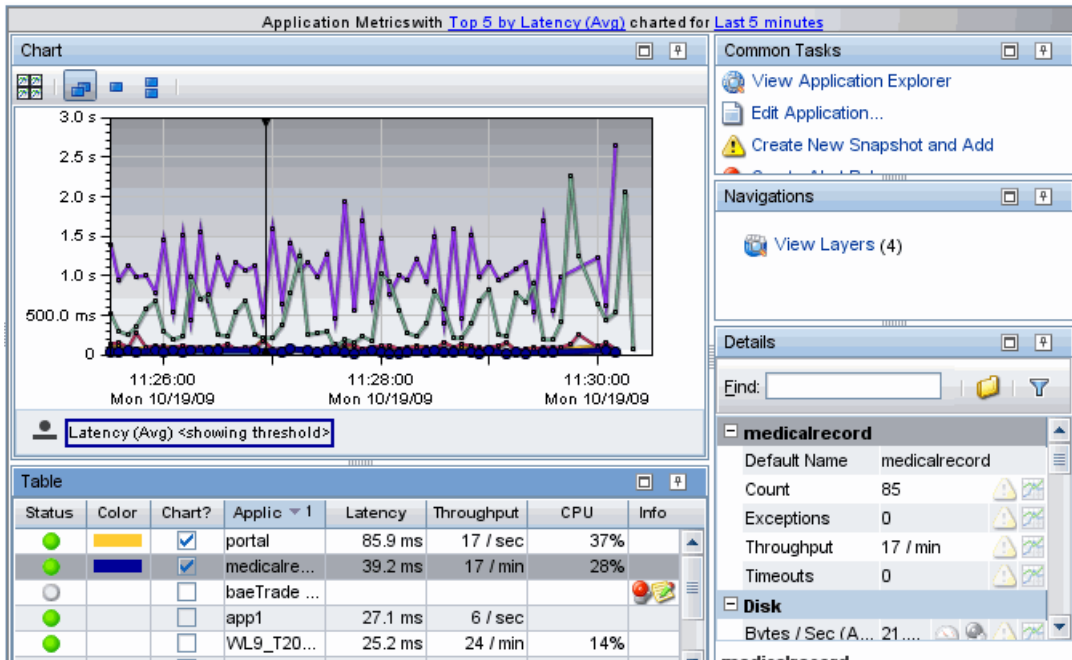
This section describes:


- ▶ [Application Metrics View Description on page 341](#)

Application Metrics View Description

The **Application Metrics** view provides a high level view of the applications you are monitoring. By default, the Application Metrics graph presents trend lines that represent the CPU utilization for each application.

The following is an example of the Application Metrics view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open, click Application Metrics.</p> <p>You can also access this view directly from the Diagnostics Applications window. To do this, select an application in the navigation pane. In the details pane, click the application metrics link.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Application Metrics View" on page 339 for information on how to analyze performance using data in this view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View Title and View Filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
<p>Chart</p>	<p>Displays a graph of the average latency for each application, as a trend line. By default, the five applications with the highest average latency during the previous five minutes are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Application Details tooltip for the trend line in the graph (or the Application name column in the table) displays the following detail data:</p> <ul style="list-style-type: none"> ➤ Application. Name of the application. ➤ Path. Path of the application. <p>Application Details tooltip for the nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the applications that pertain to the context selected, either Entire Enterprise or a specific application. The metrics reported in the table are aggregated based on the time period selected in the view filters in the title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed</p> <p>Status. By default the first column shows the status for the application. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Application. Application name.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU. Average CPU utilization (system metric)</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the applications displayed in this view. You can select View Application Explorer to see a detailed view of the selected application and you can select Edit Application to make changes to the application definition.</p> <p>The drill down selection based on a pre-defined workflow is View Layers.</p>
Details	<p>The Details pane shows application and system metrics. The metrics vary depending on the application.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

13

Topology View

This chapter describes the Topology view of the probes.

This chapter includes:

Concepts

- ▶ Topology View of the Probes on page 348
- ▶ Selecting an Entity from Topology Diagram or Table on page 349
- ▶ Topology Diagram Overview on page 351

Tasks

- ▶ How to Use the Topology Diagram Probe Grouping on page 356

Reference

- ▶ Probe Topology User Interface on page 358

Concepts

Topology View of the Probes

The Topology view (accessed from the Standard View bar) provides a graphical representation of the application being monitored by the selected probe.

When you first access the Topology view, you see a message in the place of the topology diagram instructing you to select an entity in the table to generate a topology. See "Selecting an Entity from Topology Diagram or Table" on page 349.

Diagnostics displays the topology diagram where a graph is normally displayed. The topology diagram is a mapping of entities in an application and the connections between entities. It provides you with a visualization of your application or transaction. See "Topology Diagram Overview" on page 351.

The topology display includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by grouping entities, changing diagram type, or changing connection (link) information. To find out how to use the diagram tool bar, see Chapter 8, "Working with Topologies."

The table in the Topology view contains the following tabs with detail data:

- ▶ Probes
- ▶ Probe Connections
- ▶ Consumer Connections
- ▶ Probe Groups
- ▶ Hosts

For details on the user interface, see "Topology View Description" on page 359.

An example of the Topology view is shown below. In the example, the probe ID is shown under the probe icon. Calls being made to various targets are shown as connection arrows. The remote target ID indicates the type of call being made. For example jdbc: indicates a database call, queue: or topic: indicates a JMS call. The legend describes the types of calls.

The screenshot displays the 'Diagnostics Topology' window. The main area shows a topology diagram with three nodes: 'Consumers', 'SOA_Manager_hpswros010', and 'Web Service'. An orange arrow points from 'Consumers' to 'SOA_Manager_hpswros010', and a blue arrow points from 'SOA_Manager_hpswros010' to 'Web Service'. A legend in the bottom right of the diagram area identifies the orange arrow as 'Consumer' and the blue arrow as 'Web Service ...'. Below the diagram is a 'Probes' table with the following data:

St.	Chart?	Probe	VM H. Used	Laten..	CPU (Avg)	Throug	Timeo..	Except	Info
1		WLS92_aqualogic_ovr...	372.0...	1.7 s	1.1 ms	36 / min	0	2,756	
	✓	SOA_Manager_hpswr...	30.3 ...	9.7 ms	6.5 ms	6 / min	0	186	
		TOMCAT_hpswros022...	17.4 ...						
		SB_QA_JMSClient_ros...	55.5 ...	184.9...	142.7...	96 / hr	0	0	
		L91_TestService2.Web...	215.3...	1.1 ms	81.4 μs	124 / hr	0	0	
		L91_MSPetShop4.NET...	56.8 ...	7.1 ms	5.3 ms	12 / min	0	0	
		WL81_SOA_Manager_...	51.2 ...						
		TIBCO_BW_hpswros0...	44.6 ...						

The right-hand side of the window contains several panels: 'Common Tasks' with options like 'Create New Snapshot and Add', 'View Profiler for SOA_Manager...', and 'Add to application...'; 'Navigations' with links for 'View Hosts (1)', 'View Layers (3)', 'View Outbound Calls (1)', 'View Server Requests (1)', and 'View Services (1)'; 'Details' with a search field; and a 'SOA_Manager_hpswros010' details panel showing metrics such as Default Name, Host IP Address (15.8.155.229), Host Name, Probe Type (Java), Count (557), Exceptions (186), Throughput (6 / min), and Timeouts (0). A 'Custom Attributes' section is also visible at the bottom of the details panel.

Selecting an Entity from Topology Diagram or Table

When you first access the Topology view, you see a message in the place of the topology diagram instructing you to select an entity in the table to generate a topology.

The topology diagram and the tables are integrated so that if you select an entity in the diagram then the relevant tab and entity in the table are highlighted. And if you select an entity in the table a topology is generated for that selection.

All the possible entities and connections in your enterprise are shown in the various tabs of the table: Probes, Probe Connections, Consumer Connections, Probe Groups, Hosts.

Once you generate a topology, each tab in the table will have check marks in the Chart column for those items that are represented in the currently displayed topology.

If you want to generate a different topology, you can click through the tabs in the table and select the entity you want to view from the appropriate tab. A new topology is then drawn representing the entity you just selected. Notice that the table will have different check marks in the Chart column of each tab for those items represented in the newly drawn topology.

You can also navigate to rows and tabs in the table by selecting entities in the currently displayed topology diagram (selectable entities have a blue 'halo'). For example, if the topology displayed has five probes, you will see check marks next to the five probes in the Chart column of the Probes tab. If you select a probe in the topology, the row for that probe in the table is highlighted.

If you select a connection in the topology diagram, the Probe Connections tab or Consumer Connections tab in the table will open with a check mark in the Chart column next to the consumers or probes represented by the connection arrow selected.

You can select the Group Probes by Host and Group Probes by Probe Group toolbar buttons in the topology diagram to see which hosts or probe groups are represented in the currently displayed diagram.

If you generate a topology diagram by selecting a host in the Hosts tab of the table, then the Group Probes by Host toolbar button in the topology is automatically selected. If you generate a topology diagram by selecting a probe group in the Probe Groups tab of the table, the Group Probes by Probe Group toolbar button in the topology is automatically selected.

Topology Diagram Overview

The topology display includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by grouping entities, changing diagram type, or changing connection (link) information. To find out how to use the diagram tool bar, see Chapter 8, "Working with Topologies."

Note: You can right-click items on the diagram to see a tooltip containing Name, Type, Latency, and Throughput.

You can filter the data in the topology diagram by **Probe Connections**, **Probe Group** and **Time Range**. And you can select to **Filter by** a particular string contained in a selected field like Target Name. The filter selections are located above the Topology toolbar.

All the filters apply to the topology diagram. The Filter by and Probe Connections filters are applied to the two Connections tab tables. The Probe Groups filter is applied to tables in all tabs.

So, for example, you have a topology diagram that includes both Web Service calls and JMS calls. Then you select to filter Probe Connections by Web Service. A new query is made for just web services and the topology is redrawn. The new topology diagram will only show the web service calls and not the JMS calls.

Note: With JMS calls the temporary queues are grouped into a single node. Diagnostics matches the queue/topic name to a list of regular expressions in the **capture.properties** file to find the temporary queue/topic names. The ones that match are replaced with either `queue:<probe-id>\TEMPORARY` or `topic:<probe-id>\TEMPORARY` depending on the type.

If you select a filter that results in no matching data the diagram will be replaced by the initial topology message.

The different entities displayed in the probe topology diagram are described in the following sections.

Probe Icons

The probe icons generally represent the applications that are being monitored. Probe icons (and other entities) with a blue 'halo' are selectable in the topology. When you select a probe in the topology diagram, the probe is displayed with a colored halo to indicate that it is the selected entity. The metrics for the selected probe are displayed in the details table, and the row for the probe in the table is shown as selected.

The colored dot next to each probe in the diagram indicates the status of probe, just as it does in the Status view and in the table.

Consumer Icon

The consumer icon represents calls to a probe (or other entity) coming from a non-instrumented source (not monitored by a probe). These calls are grouped together in the topology and are represented by the consumer icon.

You can't select the consumer icon but if you want to see which consumers are making the calls represented in the topology you can select the consumer connections arrow coming from the consumer icon. The Consumer Connections tab then opens to display details about all the different consumers making calls to a particular entity.

The default consumer ID is the IP address (for SOAP over HTTP web services) and queue name (for SOAP over JMS web services) but you can configure consumers in a number of ways. See "About Consumer IDs" on page 663 for details.

Connection (Links)

The connection arrows in the diagram represent aggregated calls to and from application components within your environment. Connections can be shown between various entities such as probes, probe groups, Web services and consumers, or targets. The arrow on the connection indicates the direction of the communication.

Note: Connections can be shown between Java and non-Java systems such as SAP and Oracle or SQL Server database systems.

The color of the probe connection can indicate the type of call or performance status and load, depending on the selection from the View Link drop-down menu.

The Legend shows the meaning of each connection line in the current display, according to whether you have selected Type or Load from the View Link drop-down menu.

When you click a probe connection in the diagram, the line is emphasized with a black stripe (indicating that you have selected it). You can then view information about that specific connection in the Consumer Connections or Probe Connections tabs in the table. The data displayed in the details pane is specific to the connection.

Consumer connections are indicated by an arrow pointing from the consumer icon to a probe (or other entity). The consumer icon shows the origin of an aggregation of calls coming in to an entity from a non-instrumented source (not monitored by a probe).

Probe connections are indicated by an arrow pointing to or from the probe to a target or other entity. These targets show the destination of an aggregation of calls from the caller's point of view.

When you select a probe connection in the topology diagram you may actually see several rows highlighted in the Probe Connections tab of the table. This is when the application monitored by one probe is making an inbound call to a target (not monitored by a probe) and then an outbound call is made from the target to the application monitored by another probe. In this case, since multiple connections are selected, the details pane will be empty. To see details select the connection of interest.

Probe Groups

The probe groups nodes represent groupings of probes. When you select a probe group from the probe groups tab in the detail table, the probe group is displayed with a colored halo to indicate that it is the selected entity. The metrics for the selected probe group are displayed in the details table, and the row for the probe group in the table is shown as selected.

The colored dot next to each probe group in the diagram indicates the status of probe group, just as it does in the Status view and in the table.

Hosts

The host nodes represent hosts being monitored. When you select a host from the hosts tab in the detail table, the host is displayed with a colored halo to indicate that it is the selected entity. The metrics for the selected host are displayed in the details table, and the row for the host in the table is shown as selected.

In a virtualized environment the VM guest value for a host is shown in the topology diagram and in the Details pane for the selected host.

The colored dot next to each host in the diagram indicates the status of host, just as it does in the Status view and in the table.

Status Bulb

The status bulb indicates probe status, which is based on a variety of factors, including metric thresholds. Status is only available when you view time ranges that are an hour or less. Status is shown as gray when no status is available, or when the time range exceeds an hour. See "About the Graph Entity Table" on page 108 for more information on how status is calculated.

Targets

The target icon represents calls to or from a non-instrumented source (not monitored by a probe). Targets, as determined through the instrumentation of your probes, represent the origin of inbound calls or the destination of outbound calls. A variety of icons are used to represent the different types of targets that Diagnostics recognizes. You cannot select targets in the diagram. You cannot see metrics for targets in the other parts of the Topology view.

Every type of target corresponds roughly to a type of communication technology.

JDBC Targets

For certain JDBC drivers, Diagnostics parses the JDBC URL and displays the result as the probe target (for example the resulting probe target may be in the form `host:port:sid` or `host:port:dbname`).

For the rest of the cases (where the JDBC URL is not parsed), the target is the unparsed JDBC URL. If passwords exist in the URL, they are obfuscated (they will be shown as `**`).

If you want to avoid showing the raw URL, the `jdbc.url.length` property in `server.properties` can be set to 0 which results in the unparsed JDBC connection URLs not being shown in the topology. Instead they are grouped into a target node called `"Database_<url's hash code>"`. The purpose of the hash code is so the entry is unique. This property does NOT apply to the cases where Diagnostics parses JDBC URL.

By default, Diagnostics will show the first 150 characters of the JDBC URL on the Diagnostics UI. This can be changed via the `jdbc.url.length` property in `server.properties`.

You may also see a target called `"Unknown Database"` which means that Diagnostics could not collect the JDBC connection URL.

Tasks

How to Use the Topology Diagram Probe Grouping

When Diagnostics first displays the topology diagram for a probe or probe connection that you selected from the table, the diagram is zoomed to fit the view. The default format for the diagram is hierarchical. The probes are not grouped by probe group or host.

Using the controls in the diagram and the diagram toolbar, you can control the format, layout, and resolution of the diagram, so you can see the topology the way you want to see it. See Chapter 8, "Working with Topologies" for details on using the topology toolbar and other controls in the topology views.

When the topology diagram is first displayed, the probes are not grouped. Each probe is shown individually in the diagram. You can change the way the diagram is drawn by selecting to group probes by hosts or by probe groups. Grouping probes can simplify the appearance of the diagram.



To group probes by the hosts, click **Group Probes by Host** in the toolbar. The probes in the diagram are replaced with probe hosts that contain the probes installed on the host.



To group probes by probe groups, click **Group Probes by Probe Group** in the toolbar. The probes in the diagram are replaced with probe groups that contain the probes added to the probe group.



To cause the probes to again be displayed as individual probes in the diagram, click **Don't Group Connected Probes** in the toolbar.



When probes are grouped in the diagram, a small toggle icon is shown above the icon for the group. If you click this icon, or right-click the group icon, the group is opened so you can see each of the probes in the group, along with the probe connections to the targets.



You can collapse the expanded probe group by clicking the icon in the group header, or by right clicking anywhere in the group box.



You can expand all of the groups in the diagram by clicking **Expand all Nodes** in the toolbar. Diagnostics opens all of the groups in the diagram.



You can collapse all of the groups in the diagram by clicking **Collapse all Nodes** in the toolbar. Diagnostics collapses all of the groups in the diagram.

Reference

Probe Topology User Interface

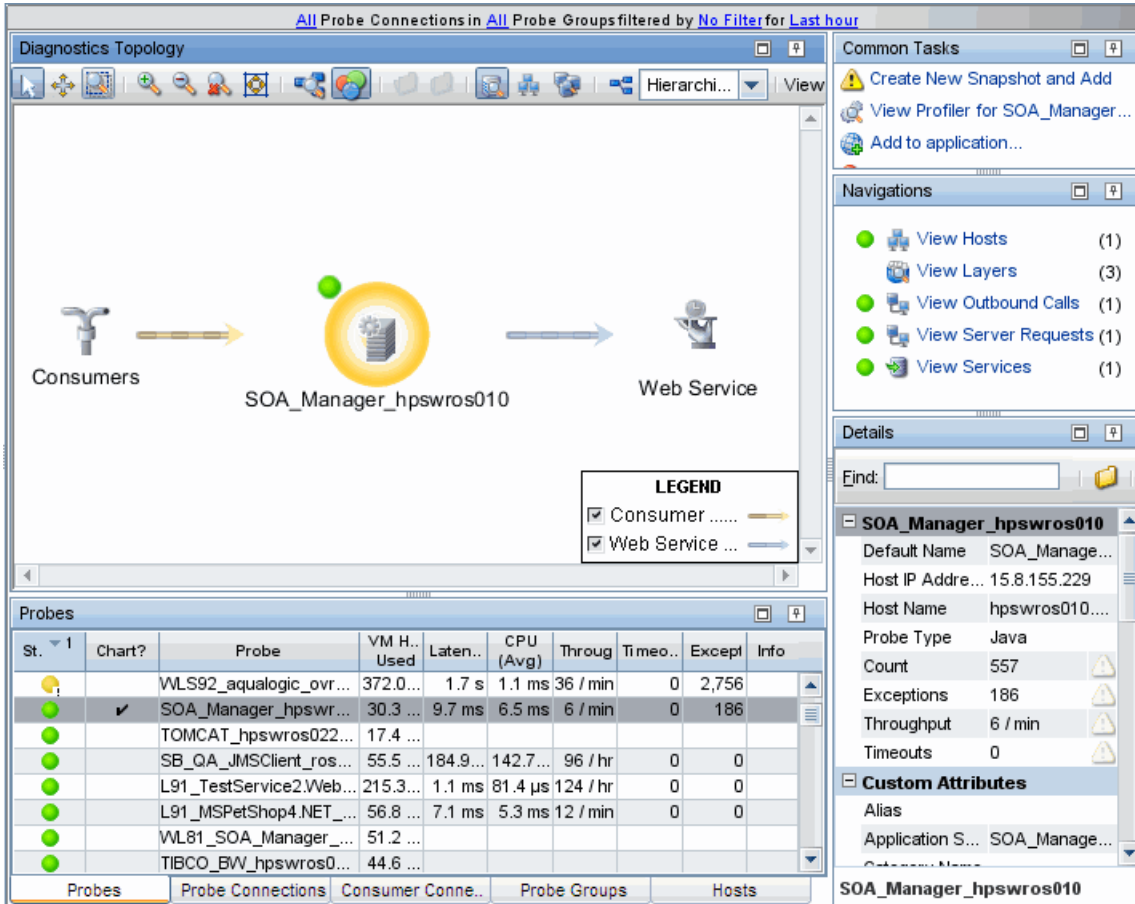
This section includes:

- ▶ [Topology View Description on page 359](#)

Topology View Description

The **Topology** view provides a graphical representation of the way your applications or transactions are working.

The following is an example of the Topology view.



The screenshot displays the 'Diagnostics Topology' window. The main area shows a flow diagram with three components: 'Consumers' (represented by a faucet icon), 'SOA_Manager_hpswros010' (represented by a server rack icon), and 'Web Service' (represented by a person icon). Arrows indicate the flow from Consumers to SOA_Manager_hpswros010, and then to Web Service. A legend below the diagram identifies the arrow types: a yellow arrow for 'Consumer' and a blue arrow for 'Web Service ...'.


Below the diagram is a table of probes. The selected probe is 'SOA_Manager_hpswr...'. The table columns are: St., Chart?, Probe, VM H. Used, Laten., CPU (Avg), Throug, Timeo., Except, and Info.

St.	Chart?	Probe	VM H. Used	Laten..	CPU (Avg)	Throug	Timeo..	Except	Info
		WLS92_aqualogic_ovr...	372.0...	1.7 s	1.1 ms	36 / min	0	2,756	
	<input checked="" type="checkbox"/>	SOA_Manager_hpswr...	30.3 ...	9.7 ms	6.5 ms	6 / min	0	186	
		TOMCAT_hpswros022...	17.4 ...						
		SB_QA_JMSClient_ros...	55.5 ...	184.9...	142.7...	96 / hr	0	0	
		L91_TestService2.Web...	215.3...	1.1 ms	81.4 μs	124 / hr	0	0	
		L91_MSPetShop4.NET...	56.8 ...	7.1 ms	5.3 ms	12 / min	0	0	
		WL81_SOA_Manager_...	51.2 ...						
		TIBCO_BW_hpswros0...	44.6 ...						

The details pane on the right shows information for the selected probe 'SOA_Manager_hpswros010':

- Default Name: SOA_Manage...
- Host IP Address: 15.8.155.229
- Host Name: hpswros010....
- Probe Type: Java
- Count: 557
- Exceptions: 186
- Throughput: 6 / min
- Timeouts: 0

Custom Attributes include: Alias, Application S... SOA_Manage...

<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Topology.</p> <p>You can access this view directly from the Applications window by selecting the Entire Enterprise and then selecting the "Viewing the application topology" link in the right pane.</p>
<p>Important information</p>	<p>When you first access the Topology view, you see a message in the place of the topology diagram instructing you to select an entity in the table to generate a topology.</p> <p>The probe connections that are available and subsequently shown in the topology are based on instrumentation.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Topology Diagram Probe Grouping" on page 356 for information on grouping probes to simplify the topology diagram.</p>
<p>See also</p>	<ul style="list-style-type: none"> ▶ See Chapter 8, "Working with Topologies" for details on using the topology toolbar and other controls in the topology views. ▶ See "Selecting an Entity from Topology Diagram or Table" on page 349. ▶ See "Topology Diagram Overview" on page 351 for details on the entities displayed in the topology.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	You can use the standard topology view controls. See Chapter 8, "Working with Topologies" for details on using the topology toolbar and other controls in the topology views.
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "Working With View Filters" on page 87.

Page Content

The following user interface elements are included:

UI Elements	Description
<Title bar filters>	<p>You can filter the data by Probe Connections, Probe Group or time range. And you can select to Filter by a particular string contained in a selected field like Target Name. The filter selections are located above the Topology toolbar.</p> <p>All the filters apply to the topology diagram. The Filter by and Probe Connections filters are applied to the two Connections tab tables. The Probe Groups filter is applied to tables in all tabs.</p> <p>So, for example, you have a topology diagram that includes both Web Service calls and JMS calls. Then you select to filter Probe Connections by Web Service. A new query is made for just web services and the topology is redrawn. The new topology diagram will only show the web service calls and not the JMS calls.</p> <p>If you select a filter that results in no matching data the diagram will be replaced by the initial topology message.</p>

UI Elements	Description
Topology Diagram	<p>The topology diagram is a mapping of entities in an application and the connections between entities. It provides you with a visualization of your application or transaction.</p> <p>If you see a message in place of the topology diagram, you must select an entity in the table to generate a topology.</p> <p>The topology display includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by grouping entities, changing diagram type, or changing connection (link) information. To find out how to use the diagram tool bar, see Chapter 8, "Working with Topologies." The topology diagram and the tables are integrated so that if you select an entity in the diagram then the relevant tab and entity in the table are highlighted. And if you select an entity in the table a topology is generated for that selection. See "Selecting an Entity from Topology Diagram or Table" on page 349 for details.</p> <p>The following entities can be seen in a topology:</p> <ul style="list-style-type: none"> ▶ Probe icons ▶ Consumer icon ▶ Connections (Links) - consumer connections and probe connections ▶ Probe Groups ▶ Hosts ▶ Targets ▶ Status Bulb <p>See "Topology Diagram Overview" on page 351 for more details on each of the above items.</p> <p>You can right-click items on the diagram to see a tooltip containing Name, Type, Latency, and Throughput.</p>

UI Elements	Description
Table	<p>Contains the following tabs:</p> <ul style="list-style-type: none"> ➤ Probes ➤ Probe Connections ➤ Consumer Connections ➤ Probe Groups ➤ Hosts <p>The topology diagram and the tables are integrated so that if you select an entity in the diagram then the relevant tab and entity in the table are highlighted. And if you select an entity in the table a topology is generated for that selection. Once you generate a topology, each tab in the table will have check marks in the Chart column for those items that are represented in the currently displayed topology. See "Selecting an Entity from Topology Diagram or Table" on page 349.</p> <p>All these tabbed displays contain the following columns:</p> <p>Status. By default the first column shows the status for the probe (or the entity selected). The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Chart?. From the table you can select the entities that are to be included in the topology diagram.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Probes Tab	<p>By default the Probes tab displays the following additional columns of data:</p> <p>Probe. The name of the probe.</p> <p>VM Heap Used. The average amount of heap used by the application.</p> <p>Latency. The average response time across all instances.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>From the Probes tab in the table, you can drill down to the Hosts view.</p> <p>In addition, you can drill down to detailed views specific to the type of application or data being collected by the probe. For example, a probe monitoring a WebLogic application may have drill down selections to the Layers view, Outbound Calls view, Server Requests view and the EJB Pooled Resource Contention, JDBC Resource Contention, JDBC Connection Status and Server Threads specialized views.</p>

UI Elements	Description
Probe Connections Tab and Consumer Connections Tab	<p>By default the Probe Connections tab and the Consumer Connections tab displays the following additional columns of data:</p> <p>Origin. Origin of the probe connection.</p> <p>Destination. Destination of the probe connection.</p> <p>Type. Type of probe connection.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>From the Probe Connections and Consumer Connections tab you can drill down to the Probes view. You may also be able to drill down to detailed views specific to the type of application or data being collected by the probes. For example, View Operations by Consumer ID.</p>

UI Elements	Description
Probe Groups Tab	<p>By default the Probe Groups tab displays the following additional columns of data:</p> <p>Probe Group. The name of the probe group.</p> <p>Contribution to the Probe Group. The percentage of the time contributed to the probe group's total time.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>From the Probe Groups tab in the table, you can drill down to detailed views specific to the type of application or data being collected by the probes in the probe group. For example, a probe group for monitoring a SQL Server database may have drill down selections for the SQL Server Probes and Wait Time specialized views, for a probe group with Java Probes you can drill down to the Probes view.</p>
Hosts Tab	<p>By default the Hosts tab displays the following additional columns of data:</p> <p>Host. The name of the host.</p> <p>CPU % Over Threshold. The percentage over threshold for Average CPU.</p> <p>CPU. Average CPU utilization.</p> <p>Context Switches/S. The average number of context switches per second.</p> <p>Disk Bytes/S. The average number of bytes read and written per second.</p> <p>Network Bytes/S. The average number of bytes sent and received per second.</p> <p>Page In/S. The average number of virtual memory pages swapped in per second.</p> <p>Memory. The average amount of memory in use.</p> <p>From the Hosts tab in the table, you can drill down to the Probes view for those hosts that have a probe installed.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the entity selected are shown in the Common Tasks pane. The drill down navigations available vary depending on the type of entity selected.
Details Pane	The Details pane lists the metrics for the selected row and selected tab of the table. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

14

Hosts View

This chapter describes the Hosts view.

This chapter includes:

Concepts

- ▶ Hosts Overview on page 370

Tasks

- ▶ How to Use the Hosts View on page 372

Reference

- ▶ Hosts User Interface on page 373

Concepts

Hosts Overview

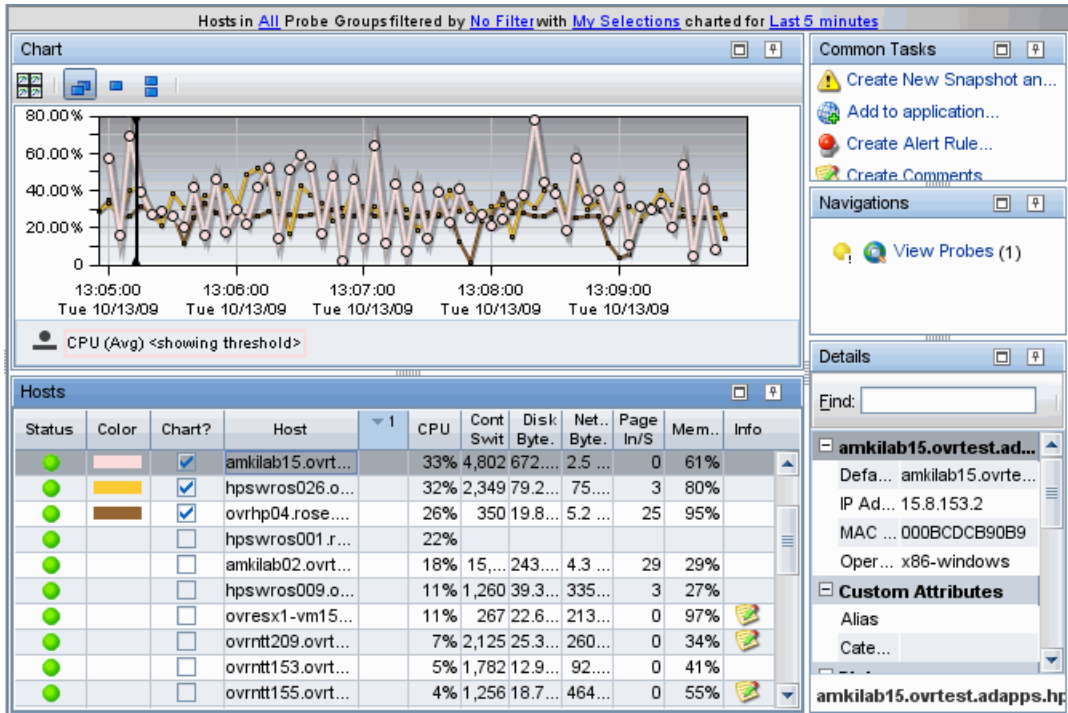
The **Hosts** view contains performance metrics for the machines that host the probes and the applications they are monitoring. By default, the Hosts View graph presents trend lines that represent the CPU utilization percentage for each host.

For details on the user interface, see "Hosts View Description" on page 373.

By default in Diagnostics standalone, the Host View graph charts the CPU utilization metrics for the five hosts that have the highest utilization percentage during the previous five minutes.

By default in Business Availability Center, the Host View graph charts the CPU utilization metrics for the five hosts that have the highest utilization percentage during the previous hour. By default in LoadRunner and Performance Center, the Host View graph charts the CPU utilization metrics for the entire load testing scenario.

An example of the Hosts view is shown below:



Tasks

How to Use the Hosts View

The Hosts view can be used in determining the health of a physical system or operating system. You can look at host system availability and operating system metrics for an indication of infrastructure problems and isolate problems to a particular host system. All the metrics that a host or operating system provides are trended over time.

For details on the user interface, see "Hosts View Description" on page 373.

If the metrics displayed in the Hosts view raises any concerns, you can navigate to other views to find out more information.

You can also take action to further analyze a performance problem. For example create a snapshot for later analysis.

Drill Down to the Probes View from Hosts

If you find a problem when looking at the Hosts view, you can navigate to view the probes running on a given host, and then into a view of the server requests captured by one of the probes.

You select drill down navigations and actions by right-clicking an item or in some cases you can double-click an item to navigate based on a pre-defined workflow. Diagnostics updates the available drill downs to allow only

Reference

Hosts User Interface

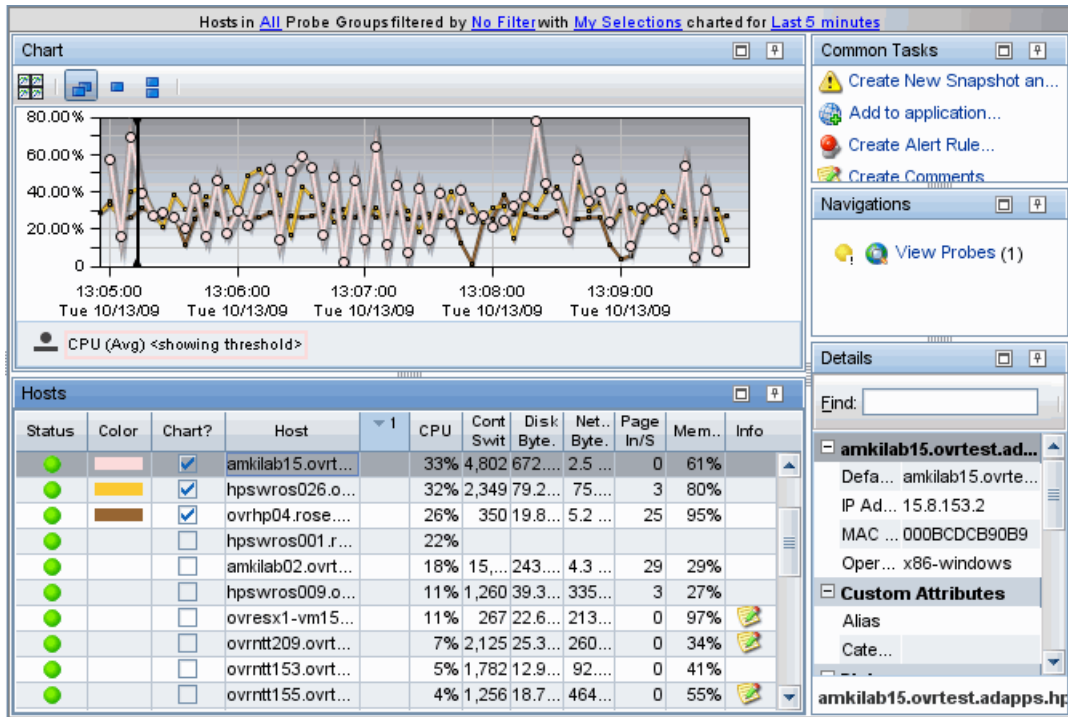
This section includes:


- ▶ [Hosts View Description on page 373](#)

Hosts View Description

The Hosts view displays latency and other performance metrics for the system that host the probes and the applications they are monitoring.

The following is an example of the Hosts view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Hosts.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Hosts View" on page 372 for information on how to analyze performance using the data in this view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p data-bbox="582 222 1218 378">Displays a graph of the average CPU for hosts, as a trend line. By default, the five hosts with the highest average CPU during the previous five minutes are charted. You can use the view filters in the Title bar to filter the data for an individual probe group.</p> <p data-bbox="582 395 1218 552">The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p data-bbox="582 569 1218 656">Host Details tooltip for the trend line in the graph (or the Host column in the table) displays the fully qualified name of the host.</p> <p data-bbox="582 673 1218 760">Host Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
<p>Table</p>	<p>Lists the hosts and the metrics for each host. Metrics are aggregated and reported, based on the time period specified in the Viewing filter.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the host. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Host. Host name.</p> <p>CPU Percent Over Threshold. The percentage over the threshold for average CPU.</p> <p>CPU. Average CPU utilization.</p> <p>Context Switches/S. The average number of context switches per second.</p> <p>Disk Bytes/S. The average number of bytes read and written per second.</p> <p>Network Bytes/S. The average number of bytes sent and received per second.</p> <p>Page In/S. The average number of virtual memory pages swapped in per second.</p> <p>Memory. The average percentage of memory in use.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the hosts displayed in this view.</p> <p>The drill down selection based on a pre-defined workflow is View Probes. The Probes view shows detail data for the probes executing on the selected host.</p>
<p>Details Pane</p>	<p>The Details pane shows metrics for the hosts. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p> <p>In a virtualized environment the VM guest value for a host is shown in the Details pane for the selected host.</p>

15

Server Requests View

This chapter describes the Server Requests view.

This chapter includes:

Concepts

- ▶ Server Requests Overview on page 380
- ▶ Display CPU Time for Server Requests on page 381
- ▶ Instance Trees and Call Profiles for Server Requests on page 382
- ▶ About Instance Tree Markers on page 383

Tasks

- ▶ How to Use the Server Requests View on page 385
- ▶ How to Delete a Server Request on page 388

Reference

- ▶ Server Requests User Interface on page 389

Concepts

Server Requests Overview

The **Server Requests** view provides the top entry points to the application server and is useful in identifying individual server requests with poor performance.

For details on the user interface, see "How to Use the Server Requests View" on page 385.

By default the Server Requests view charts the average latency for the five server requests that have the highest latency percent over threshold values.

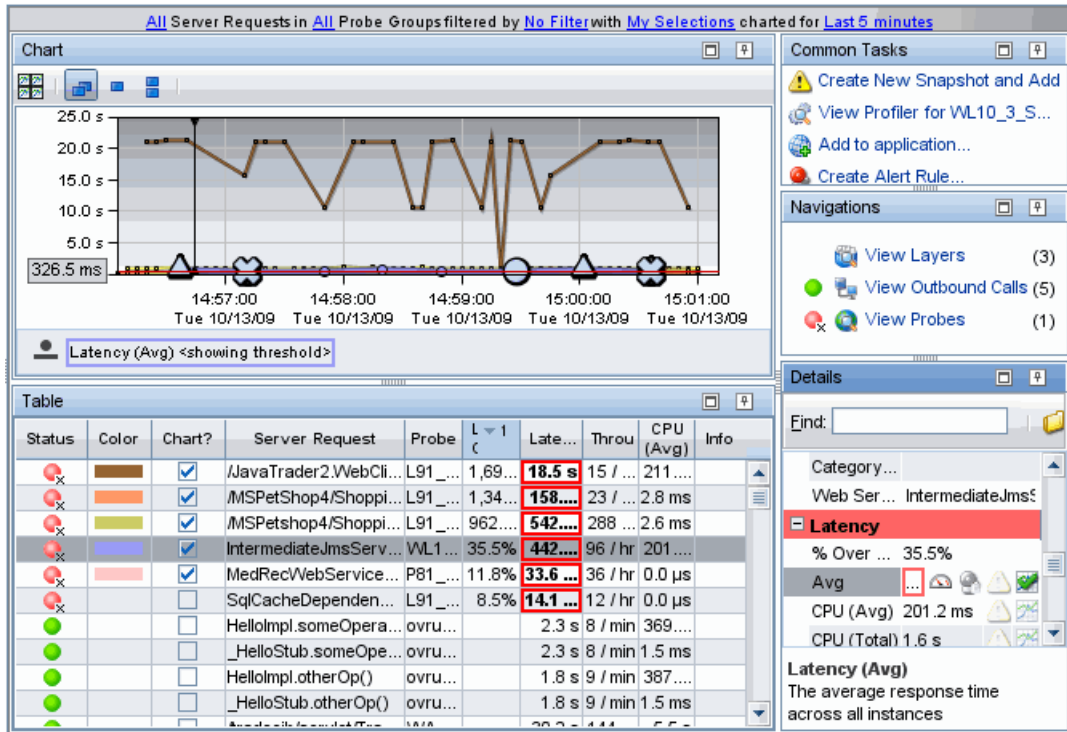
If you navigated to the Server Requests view from the View bar, the Server Requests view shows all of the Server Requests for all the probes monitoring the selected application. If you navigated to the Server Requests view from a drill down (for example from the Probes view), then only the requests for the selected probe are displayed.

Server request types can include the following: ADO, CICS, Corba, HTTP, JDBC, JMS, MSMQ, RMI, Remoting (.NET), SAP ABAP types, Web Services.

In addition, you may see server request types named Pseudo and RootRename. **Pseudo** means the server request was not really a server request defined as something that 'hits' your server, like http. If you instrumented just a regular method, then the server request type will be pseudo.

RootRename is also used for things that are not really server requests. Instead of pseudo the RootRename is based on a points file specification. These server requests will have a name that start with Background -. The suffix is the value of the rootRenameTo property specified in the point or the name of the first sub-layer if this property is not specified. To mark a point as a root rename, the detail 'when-root-rename' must be specified in the point. Refer to the *HP Diagnostics Installation and Configuration Guide* Custom Instrumentation sections for details.

An example of the Server Requests view is shown below:



Display CPU Time for Server Requests

By default, collection of CPU time metrics is enabled for server requests. The CPU time metrics are CPU (Avg) and CPU (Total), displayed in the Latency section of the Details pane.

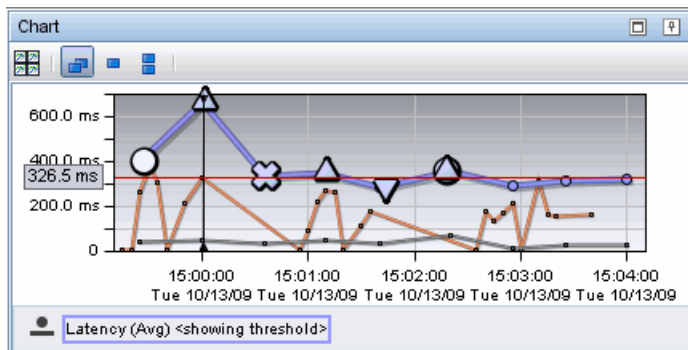
Collection of CPU time metrics can be disabled for server requests, in which case you will see N/A in the Details pane for the CPU (Avg) and CPU (Total) metrics.

You can enable, disable and configure the collection of CPU time metrics for both the Java and .NET probes. For details, see "Configure Metrics" on page 168 for details.

Instance Trees and Call Profiles for Server Requests

The Server Requests view displays a trend line showing the average of all the instances for a selected request.

In the following figure the circle and triangle icons that appear are the markers that indicate where minimum, maximum and average instance trees for the server request have been saved to help you understand the performance of your applications. The X indicates a cross VM instance tree. You may also see markers to indicate when there is an exception or a SOAP fault is thrown.



The icons mark the points at which instance trees have been created for significant method calls that were made during the execution of the server request. The instance examples let you view, for example, the minimum, maximum and average user and compare this to the running average in the trend line.

Visually analyzing the convergence of the min/max with the running average line lets you determine the real behavior of the application to determine if the application is seeing consistently bad performance or overall good performance with some outliers.

Once you've determined which instance to analyze, you can drill down from the icon to see a call profile. You can then compare a 'good' performing call profile with a 'bad' performing call profile. In addition, if you see a pattern in the convergence of the min/max instances with the running trend line you can create a snapshot and look for other events that follow a similar pattern.

The call profile is presented in the Call Profile view as an instance tree that depicts the method calls and their latency in a graph and in a table.

For more information about instance trees, see Chapter 9, "Working with Instance Trees."

For more information about the Call Profile view, see Chapter 16, "Call Profile View."

About Instance Tree Markers

Instance tree markers are displayed for the selected server request. The markers are placed on the graph to indicate the total latency and the end time of the server request for which the instance tree was created. For a detailed explanation of instance trees, and how Diagnostics captures them, see Chapter 9, "Working with Instance Trees."

Four instance tree marker icons are used to represent the four different types of instance trees that are created for a server request:



- ▶ **Maximum Instance Tree.** The maximum instance tree for a server request represents an instance tree with the largest latency for the time period. This tree depicts one of the worst-performing invocations of the server request and resulting root method. This is one of the primary tools used for diagnosing the root cause of poor application performance.



- ▶ **Minimum Instance Tree.** The minimum instance tree for a server request represents an instance tree with the smallest latency for the time period. This tree depicts one of the best-performing invocations of the selected server request and resulting root method. It can be useful for comparison to instances that perform poorly.



- ▶ **Average Instance Tree.** The average instance tree for a server request represents an instance tree that represents the average latency for the time period. This tree depicts a typical invocation of the selected server request and resulting root method call. It can be useful for comparison to instances that perform poorly.



- **Cross VM Instance Tree.** The cross VM instance tree for a server request represents an instance tree that includes a call to another server request by using a technology such as RMI or Web services.

The following icons are used to represent exceptions and faults. Additional information is collected by the probe when exceptions or SOAP faults occur. You can drill down to the call profile graph and view exception or fault details for a method, including the stack trace.



- **Exception.** The exclamation mark indicates where an exception occurred. When you click the icon, a call profile opens. This call profile has the method that threw the exception marked with a yellow dashed outline and an Exceptions tab providing details on the exception including the stack trace.



- **SOAP Faults.** The icon indicates where an soap fault occurred. See "Types of SOAP Fault Markers" on page 271 for details on the icons used to represent different types of faults. When you click the icon, a call profile opens. This call profile has a SOAP Faults tab providing details on the fault, including the stack trace.

You can view a Server Request Instance Details tooltip showing the type of instance, by holding the mouse pointer over the marker until the Server Request Instance Details tooltip is displayed.

Tasks

How to Use the Server Requests View

You can use the Server Requests view to find the top entry points to the application server and isolate performance problems to a particular server request. Then you can drill down to a breakdown of methods and a call profile.

This task includes the following steps:

- "Interpret the Data in the Server Requests View" on page 385
- "Drill Down to the Layers View for a Server Request" on page 386
- "Drill Down to the Server Request Topology View" on page 386
- "Drill Down to the Profiler for a Server Request" on page 386
- "Drill Down to an Instance Tree Call Profile" on page 387

Interpret the Data in the Server Requests View

You can use the Server Requests view to find the top entry points to the application server and isolate performance problems to a particular server request. Then you can drill down to a breakdown of methods and a call profile. The running trend line in the graph lets you see the average of all the instances of a single unique 'named' request. The instance tree icons let you view the minimum/maximum and average user request and compare them to the running average in the trend line.

When you drill down from a probe to server requests, all server requests for the probe are displayed, not just those filtered for the application you selected in the Applications window. When you are viewing data for a particular application there is a column in the graph entity table (**App?**) that indicates which server requests are part of the application context you selected.

Depending on the application and workload, you may see an average of zero latency if you look at average response time on the probe. This result is caused by the low-resolution java timer. (The java timer cannot measure things that take less than 10ms on many platforms, including Windows.) The Server Requests that typically fall into this category include static content (for example, GIFs and JPEGs) as well as Database and housekeeping threads in an application. If you want non-zero response times, you can turn on the native (JNI) timestamps for high-resolution timings.

If a server request involves EJB business method invocations you can see these methods in the call profile and see a business tier in the Load or Layers views.

Caution: Diagnostics does not track forwarded requests (HTTP redirection with HTTP 30x response code).

Drill Down to the Layers View for a Server Request

You can drill down from a server request to the Layers view. This is a pre-defined workflow so if you double-click a server request in the table, you navigate automatically to the Layers view.

Diagnostics displays the Layers view with more details for the selected server request. The Layers view displays a breakdown of the latency contribution for each layer in the server request.

Drill Down to the Server Request Topology View

You can drill down from a server request to the Server Request Topology view. Diagnostics displays a topology diagram giving a visual representation of the inbound and outbound calls for the selected server request.

Drill Down to the Profiler for a Server Request

You can drill down from a server request to the Profiler. The Profiler launches in a separate window and provides additional information for you to use in analyzing a performance problem.

In the Java Profiler you can get real time data such as:

- ▶ Dynamic instrumentation of a sampled method
- ▶ Threads data
- ▶ Allocation/Lifecycle analysis data
- ▶ Heap breakdown data and a heap walker

Drill Down to an Instance Tree Call Profile

There are two ways to drill down to the instance tree call profiles that are represented by the instance tree markers on the graph of the Server Requests view:

- ▶ Click the instance tree marker. Diagnostics displays the Call Profile view for the selected instance in the Diagnostics UI.
- ▶ Right-click the instance tree marker and select either **View Call Profile** or **View Call Profile in New Window**. Opening the call profile in a new window is useful when you want to compare the call profiles for several different server requests, or for the same server requests at different times.

Note: It is possible that the instance tree that is displayed for an instance tree marker is different than the one you anticipated. This can happen if the instance tree on the Diagnostics Server was replaced since the last update to the UI. As a result, even though a specific tree was selected, another tree that matches the type of the selected instance tree is displayed.

For more information about the Call Profile view, see Chapter 16, "Call Profile View."

How to Delete a Server Request

This task describes how to delete a server request.

To delete a server request:

- 1** Select a server request in the table.
- 2** Right-click and select **Delete <server request name>**.

Note: The server request data is deleted from the server, however if the probe is still collecting the data then it will again be sent from the probe to the server and will also be seen in the user interface.

Reference

Server Requests User Interface

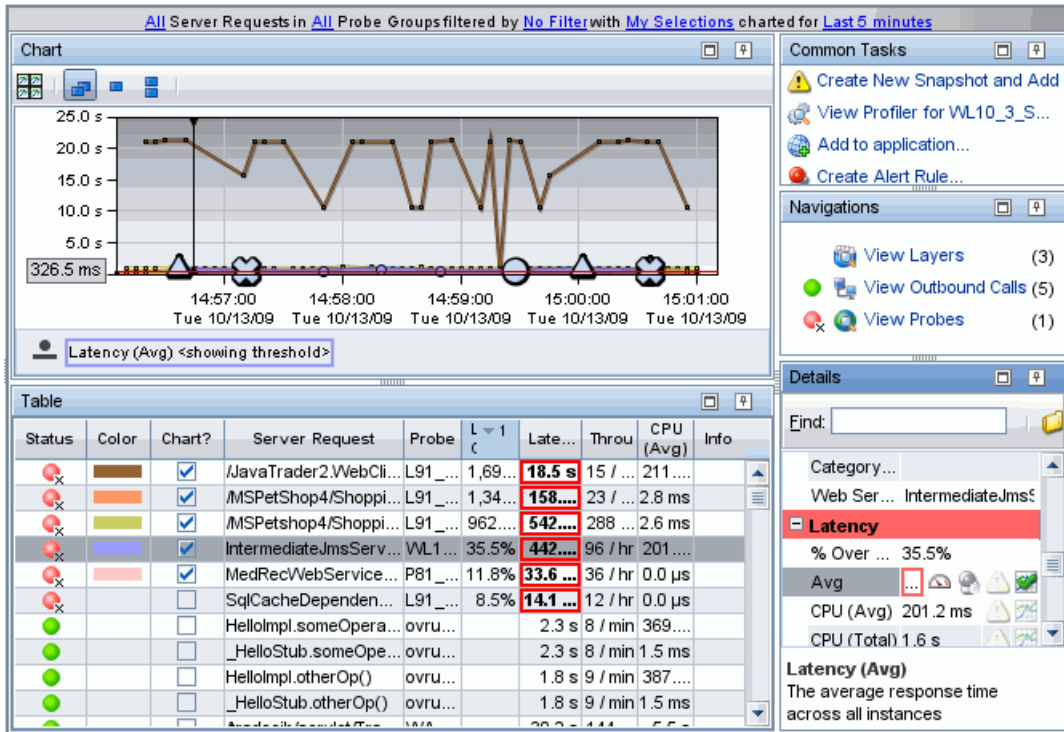
This section includes:


- ▶ [Server Requests View Description on page 389](#)
- ▶ [Server Request Topology on page 396](#)
- ▶ [Breakdown by Portal Components View Description on page 401](#)

Server Requests View Description

The **Server Requests** view provides the top entry points to the application server and is useful in identifying individual server requests with poor performance.

The following is an example of the Server Requests view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Server Requests.</p> <p>You can also access this view directly from the Applications window by selecting the link under Server Requests in the Application Contents pane.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>

Relevant tasks	See "How to Use the Server Requests View" on page 385 for information on how to analyze performance using the data in this view.
See Also	See "Instance Trees and Call Profiles for Server Requests" on page 382 for information on drilling down from an instance tree marker to the Call Profile view.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five server requests that have the highest average latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular server request or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Server Request Details tooltip for the trend line on the graph (or the Server Request column in the table) displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself.</p> <p>Type. The server request type. Examples of server request types include: ADO, CICS, Corba, HTTP, JDBC, JMS, MSMQ, RMI, Remoting (.NET), SAP ABAP types, Web Services. In addition, you may see server request types named Pseudo and RootRename.</p> <p>For details about Pseudo and RootRename, see "Server Requests Overview" on page 380.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Deployed Into. Tells where the web service was deployed into (usually an EAR file).</p> <p>Method Signature. The signature of the root method.</p>

UI Elements	Description
	<p>The information displayed in the tooltip depends on the type of server request (for example, SAP ABAP Server Request Details or Web Service Details).</p> <p>The Server Request Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created.</p> <p>You can drill down from the instance tree icon to see a call profile. See "Instance Trees and Call Profiles for Server Requests" on page 382.</p>

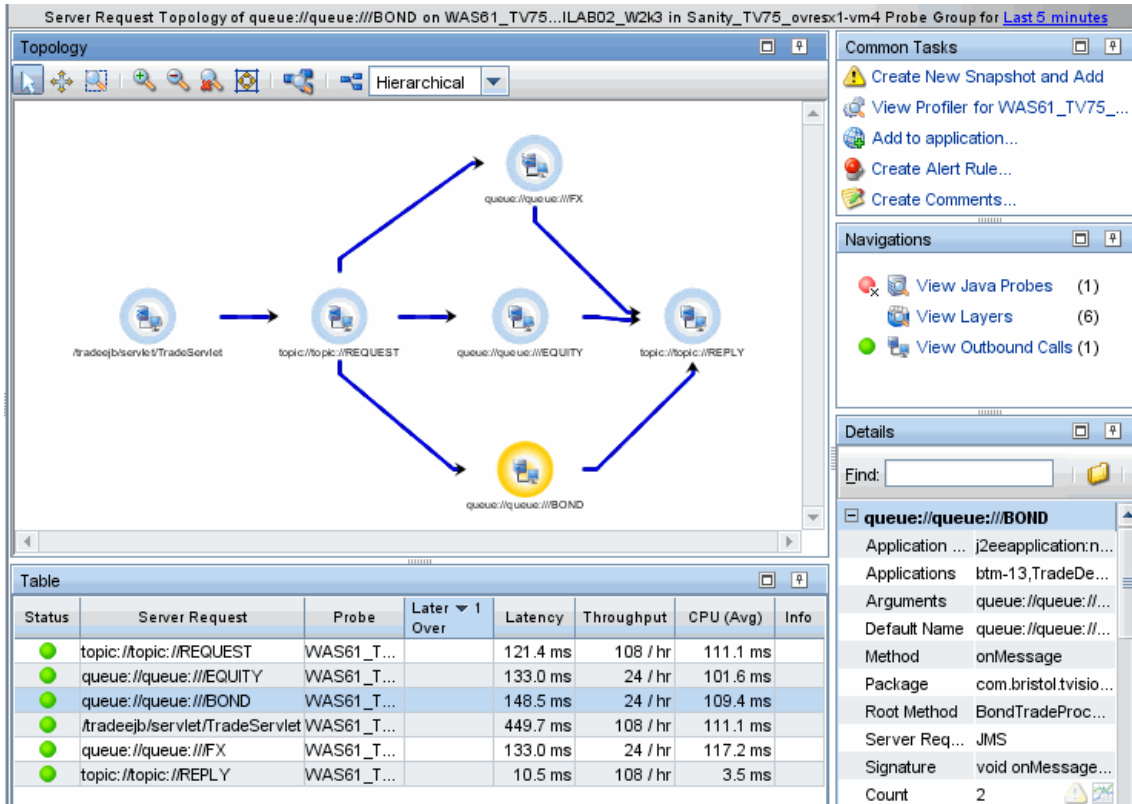
UI Elements	Description
Table	<p>Lists the server requests that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the server request. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Server Request. Name of the server request.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Layers. The Layers view displays a breakdown of the latency contribution for each layer in the server request. Another common drill down is to the Server Request Topology.</p>
Details Pane	<p>Shows metrics collected on server requests.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Server Request Topology

The Server Request Topology view gives you a visual representation of the inbound and outbound calls for a selected server request.

The following is an example of the Server Requests Topology view.



To access	Access the Server Requests Topology view by drilling down from a server request in the Server Requests view or other views displaying server requests (except when the Server Request view is a bread crumb of the Transaction view).
See also	For details about the controls in a topology view, see Chapter 8, "Working with Topologies".

Page Contents

User interface elements are described below:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.
Topology Diagram	<p>Displays a topology diagram that the selected server request participates in.</p> <p>The topology display includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by changing diagram type or turning off and on the display of the legend. To find out how to use the diagram tool bar, see Chapter 8, "Working with Topologies."</p> <p>You can right-click items in the topology diagram to see a tooltip containing information the entity.</p> <p>The topology diagram and the table are integrated so that if you select an entity in the diagram then the relevant row in the table is highlighted.</p>

UI Elements	Description
Table	<p>Lists the server requests that participate in the topology of the selected server request. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the server requests. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Server Request. Name of the server request.</p> <p>Probe. Name of the probe collecting metrics.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

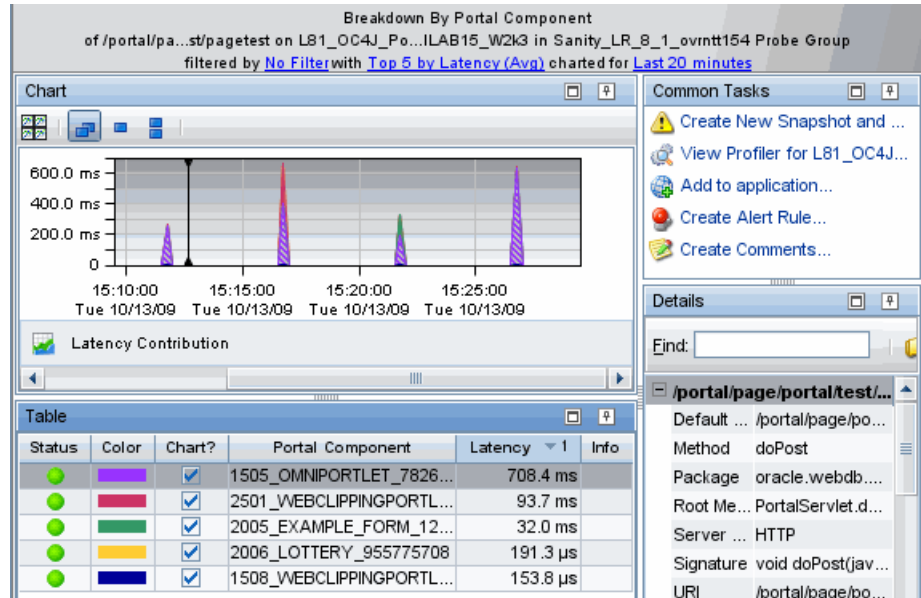
UI Elements	Description
	<p>Server Request Details tooltip for the trend line on the graph (or the Server Request column in the table) displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself.</p> <p>Type. The server request type. Examples of server request types include: ADO, CICS, Corba, HTTP, JDBC, JMS, MSMQ, RMI, Remoting (.NET), SAP ABAP types, Web Services. In addition, you may see server request types named Pseudo and RootRename.</p> <p>For details about Pseudo and RootRename, see "Server Requests Overview" on page 380.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Deployed Into. Tells where the web service was deployed into (usually an EAR file).</p> <p>Method Signature. The signature of the root method.</p> <p>The information displayed in the tooltip depends on the type of server request (for example, SAP ABAP Server Request Details or Web Service Details).</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Layers. The Layers view displays a breakdown of the latency contribution for each layer in the server request. Another common drill down is to the Profiler.</p>
Details Pane	<p>Shows metrics collected on server requests.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Breakdown by Portal Components View Description

Displays a breakdown of the server request by portal components.

The following is an example of the Breakdown by Portal Components view.



To access

Drill down to the Breakdown by Portal Component view from an originating portal request in the Server Requests view.

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the latency contribution for the portal component with the highest percent contribution for the selected server request. The latency for each portal component is shown in a stacked area chart. By default, the five portal components with the highest percent contribution are charted. You can use the view filters in the Title bar to filter the data.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the total latency contribution in seconds.</p> <p>Server Requests Details tooltip for the stacked area chart displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Method Signature. The signature of the root method.</p> <p>URI.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself.</p>

UI Elements	Description
Table	<p>Lists the portal components that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the portal component. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Portal Component. The name of the portal component.</p> <p>Latency. The average response time across all instances.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	Common tasks and actions relevant to the data displayed in this view.
Details Pane	<p>Shows metrics collected on portal component.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

16

Call Profile View

This chapter describes the Call Profile view.

This chapter includes:

Concepts

- ▶ Call Profile Overview on page 406
- ▶ Configure Collection and Display of Exception Data on page 409
- ▶ Configure Collection of SOAP Faults and Payloads on page 409
- ▶ Asynchronous Thread Stack Trace Sampling on page 411

Tasks

- ▶ How to Use the Call Profile on page 414
- ▶ How to do Call Profile Analysis on page 422

Reference

- ▶ Call Profile User Interface on page 426

Concepts

Call Profile Overview

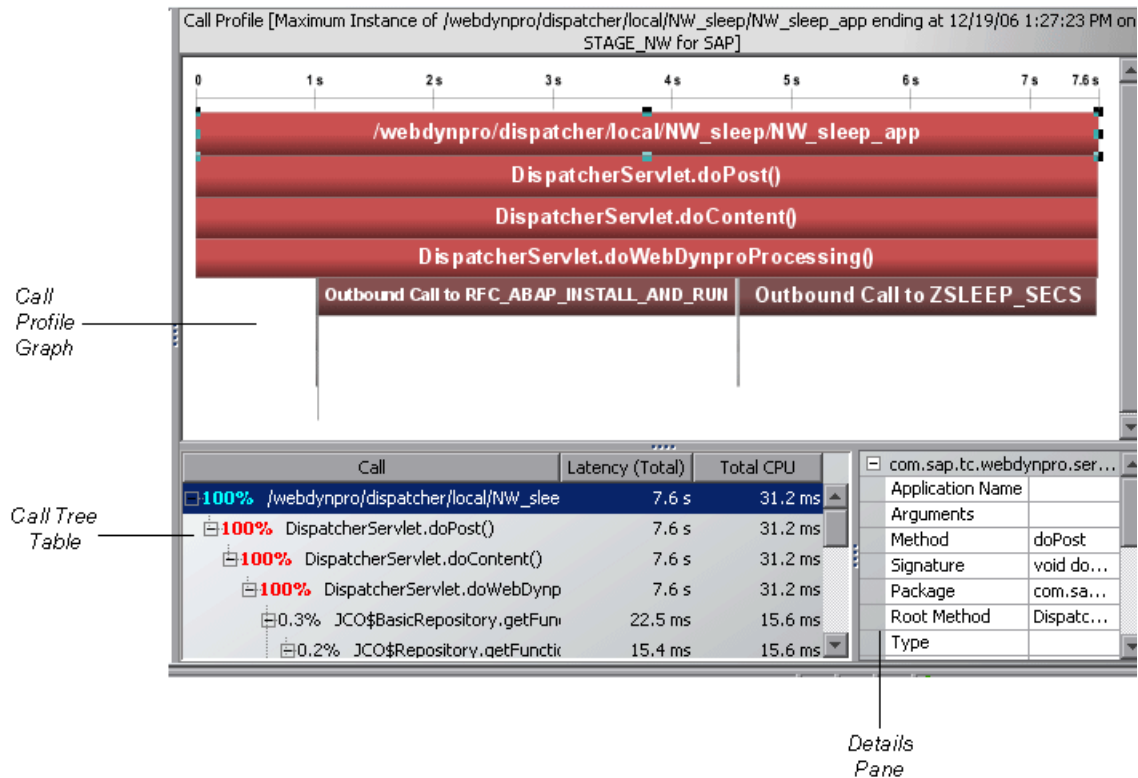
The **Call Profile** view displays the method calls and their latency for a particular instance of a monitored server request in your application.

For details on the user interface, see "Call Profile View Description" on page 427 .

The Call Profile view is made up of three areas in addition to the Title bar that is part of all Diagnostics views:

- Call Profile Graph
- Call Tree Table
- Details pane

An example of the Call Profile view is shown below:



Call Profile Graph

The Call Profile graph displays the method calls that make up a server request in a graphical format, highlighting the latency of each call, the time when each call took place, and the call stack for each call.

Each box in the call profile represents a call where the left edge of the box is the start of the method call and the right edge is the return from the call. The length of the box indicates the duration of the call execution. The position of the call box along the horizontal axis indicates the actual time that the call started and ended. The call boxes that appear directly beneath a parent call box are the child calls that are invoked by the parent.

The gaps between the call boxes on a layer of the call profile indicate one of the following processing conditions:

- ▶ The processing during the gap was taking place in code that is local to the parent at the previous higher level, and not in child calls in a lower layer.
- ▶ The processing during the gap was taking place in a child call that was not instrumented or included in a capture plan for the run.

The call boxes are colored to emphasize the critical path calls. The calls that are part of a path through the profile that has the highest latency are colored red. Call path components that are not part of a critical high-latency path are colored grey. Note that for a call profile showing a cross-VM call tree, each "hop" will be colored differently to help visually distinguish the calls that occurred on each tier.

If the duration of a call is very short or if the call appears further down in the call stack, the name of the method that is represented by the call box can become too small to read. You can view the details for a particular call by holding your mouse pointer over the call box until a tooltip is displayed. You can view additional information for a call by clicking the call box. This causes the selection in the tree table to move to that call, and prompts the Details pane to show the details for the call you clicked.

The call profile graph may have tabs across the top if data for exception instances and SOAP faults or payload was captured.

Call Tree Table

The Call Tree table appears directly below the Call Profile graph. This table presents the information from the Call Profile graph in a tabular format.

The first row in the table contains the root of the call stack, which is the server request that you drilled down to when you requested that the Call Profile view be displayed for a server request. The child rows in the tree are the method calls that were made as a result of the server call. The method calls that are parents in the call stack have the expand/collapse controls in front of them so that you can display the parents and children as required.

Note: When you click a row call in the Call Tree table, the corresponding box is selected in the Call Profile graph, and the metrics for the selected call are displayed in the Details pane.

Details Pane

The Details pane lists the metrics for the call that is selected in the Call Tree table and the Call Profile.

Note: The Details pane in the Call Profile view does not enable setting thresholds, adding to a snapshot, or charting other metrics.

Configure Collection and Display of Exception Data

By default, every exception that occurs in the monitored application is a candidate for the exception instance tree. Collecting all exception information is usually undesirable however, because exceptions that are not of interest overload the display as well as the data collection and transfer operations. You can therefore limit the exception types for which data is collected by the probe.

See the *HP Diagnostics Installation and Configuration Guide* for details on limiting exception tree data.

Configure Collection of SOAP Faults and Payloads

Diagnostics captures SOAP faults and saves instance trees for the faults. By default, up to two SOAP faults per time period are captured.

See the *HP Diagnostics Installation and Configuration Guide* for details on configuring the probes to capture SOAP fault data.

If a SOAP fault is detected, the SOAP payload is added to the SOAP fault data being collected, (for all SOAP faults except the Invalid Operations SOAP Fault, or as configured). You can view the payload information as part of the SOAP fault instance tree.

Payload capture on SOAP faults is enabled by default. But you can configure this for trimming.

Caution: With the Java probe, capturing SOAP payload requires the Diagnostics SOAP message handler. For some application servers, special instrumentation is provided in Diagnostics to automatically load the Diagnostics SOAP message handler.

However, some manual configuration is required for WebSphere 5.1 JAX-RPC and Oracle 10g JAX-RPC, see the *HP Diagnostics Installation and Configuration Guide* for details on configuring the SOAP message handler and configuring probes to capture SOAP fault data.

In addition, the Diagnostics SOAP message handler is not available for all application servers nor is custom instrumentation available to capture SOAP faults or consumer IDs from SOAP payloads, and therefore this feature is not available on all versions of all application servers. For the most recent information on Diagnostics SOAP message handler support refer to the Diagnostics Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp.

Note: For the Java probe, SOAP payload capture on SOAP faults is for both SOAP over HTTP and SOAP over JMS web services since the mechanism used to capture the payload is transport independent.

Asynchronous Thread Stack Trace Sampling

Asynchronous thread stack trace sampling for Java applications can be used to identify what the application is doing when exceptionally long executing times are observed. It is a collection technique in which some threads are periodically looked at and their state is recorded. The state includes the thread stack trace.

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

Caution: To use thread sampling you need JVM 1.5 or higher. And please make sure that all recommended operating system patches are installed.

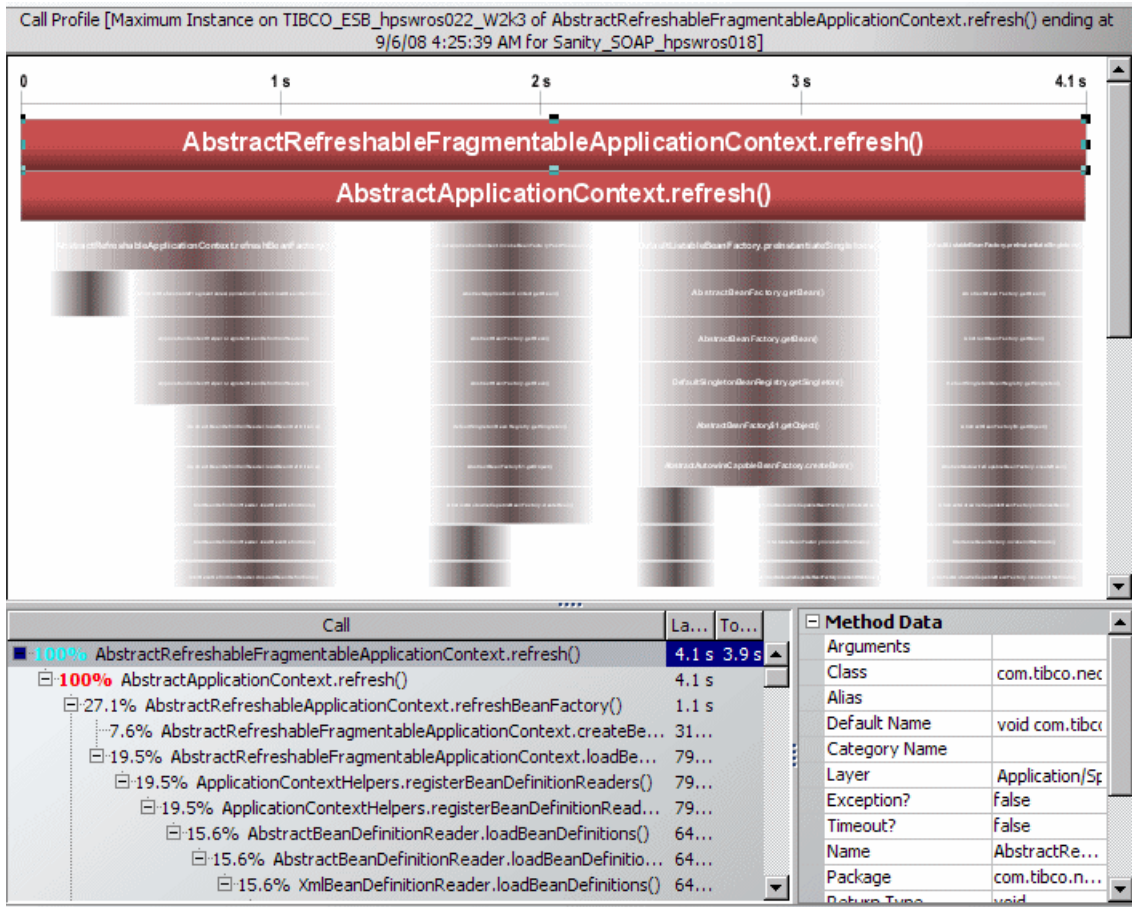
Collecting stack trace samples is a costly and heavyweight operation for all contemporary Java Virtual Machines. At the same time, there's a natural trade-off between the quality of data reported by sampling, which calls for frequent sampling, and performance requirements for production environments.

Several properties are available to enable and control the sampling process. In **dynamic.properties**: `enable.stack.trace.sampling`, `tardy.method.latency.threshold`, `stack.trace.sampling.rate`, `stack.trace.depth.max`. In **dispatcher.properties**: `enable.stack.trace.aggregation`, `aggregated.stack.trace.validity.threshold`.

See the *HP Diagnostics Installation and Configuration Guide* chapter on "Advanced Java Agent and Application Server Configuration" for more information on how to configure thread sampling.

When asynchronous thread sampling is enabled you can see additional nodes added into the call profile view by sampling. These nodes are distinguished by their different (fuzzy) shading to emphasize lack of data about the represented method start and end times.

An example of a Call Profile with asynchronous thread sampling enabled is shown below.



It is important to remember that the method invocations presented by the solid nodes - the part of the call profile corresponding to the instrumented methods - do not show direct call relationship, but merely the indirect call hierarchy and order.

The role of sampling is to show what the topmost (on the call stack) instrumented methods were doing, so the additional nodes are added only at the bottom (on the graph) of the Call Profile as obtained from execution of the instrumented methods.

An instrumented method will never be presented by a sampling (fuzzy) node, and vice versa, a non-instrumented method will never be presented by a solid node. Therefore, the only non-instrumented methods presented are those that were called directly or indirectly from the currently topmost instrumented method.

To avoid confusion, you should keep in mind that sub-trees consisting of sampling nodes are consistent only as long as the topmost instrumented method remains the same. For example, if a solid node A has two children: a sampling node B and a solid node C, we cannot conclude that C was called directly from A. It is possible that C was called from B, or any other non-instrumented method called directly or indirectly from A.

Each sampling node may represent a single, or multiple invocations of the method. No call arguments are available. Exceptions thrown by these methods are shown as thrown by the topmost (on the call stack) instrumented method. The only attributes shown by the sampling nodes are the call hierarchy and order, and an estimate of summary latency. Due to the nature of sampling the sub-graphs of sampling nodes are almost always incomplete.

See the *HP Diagnostics Installation and Configuration Guide* chapter on "Advanced Java Agent and Application Server Configuration" for troubleshooting information if you don't see any sampling nodes after enabling stack trace sampling.

Caution: Dynamic instrumentation (the **Instrument** menu item which allows you to instrument sampled methods) is **ONLY** available for sampled data and you need to access the Call Profile from the Diagnostics Profiler for Java. It is **NOT** available when accessing the Call Profile from an instance tree icon in the main Diagnostics UI. See "Instrumenting a Sampled Method Dynamically" on page 976 of the Java Profiler topics for details.

Tasks

How to Use the Call Profile

This task describes how to drill down to the call profile and how to find exceptions and SOAP faults.

This task includes the following steps:

- "Drill Down to a Call Profile" on page 414
- "Find Exceptions" on page 416
- "See Exceptions in the Call Profile" on page 416
- "Get More Information on Exceptions" on page 417
- "Arguments Displayed in the Exception Stack Trace Detail" on page 418
- "Find SOAP Faults" on page 419
- "Get More Information on SOAP Faults (and SOAP Payload)" on page 420

1 Drill Down to a Call Profile

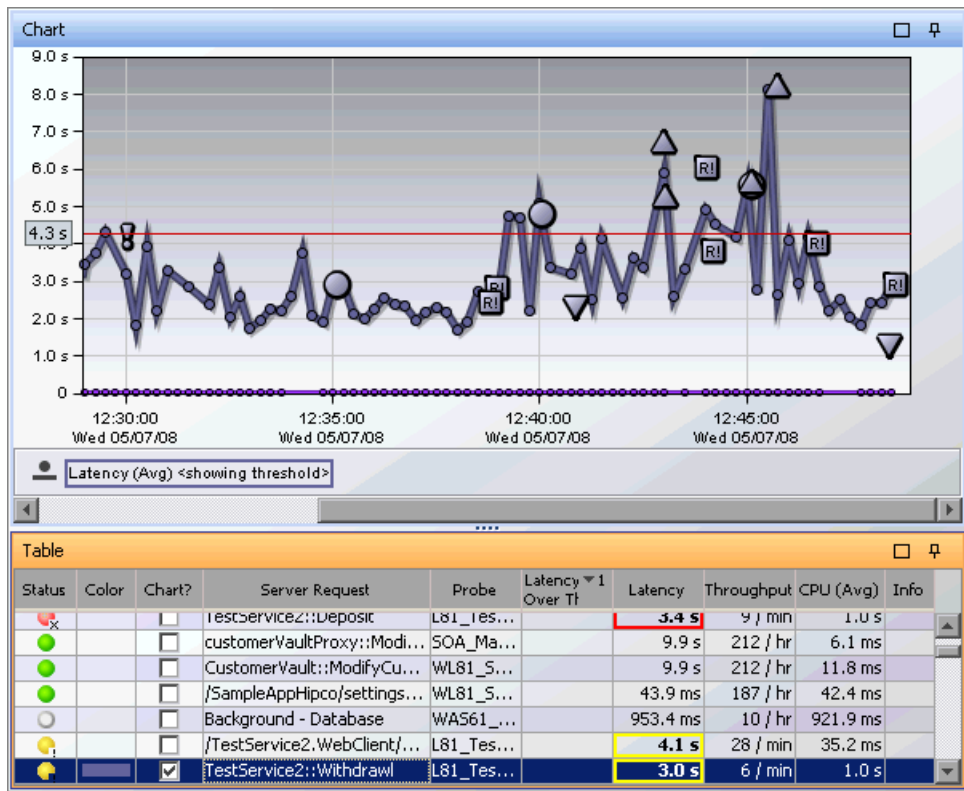
You can access the Call Profile view by clicking on an instance marker in the Server Requests view. Special instance marker icons are displayed in the Server Requests view to mark the points at which instance trees (for example, minimum, maximum and average) have been created for significant method calls that were made during the execution of the server request.

Once you've determined which instance to analyze, you can drill down from the icon to see a call profile.

The instance trees are available on the Min, Max, and Average Latency metrics and for cross-VM calls, exceptions and SOAP faults.

Example:

An example of the Server Requests view is shown below. The circle and triangle icons that appear above, below, and on top of the selected trend line are the markers that indicate where minimum, maximum and average instance trees for the server request have been saved. The exclamation point icon indicates exceptions occurred and the square icon indicates SOAP faults. Clicking on one of these icons opens the Call Profile view displaying a call tree for the instance.



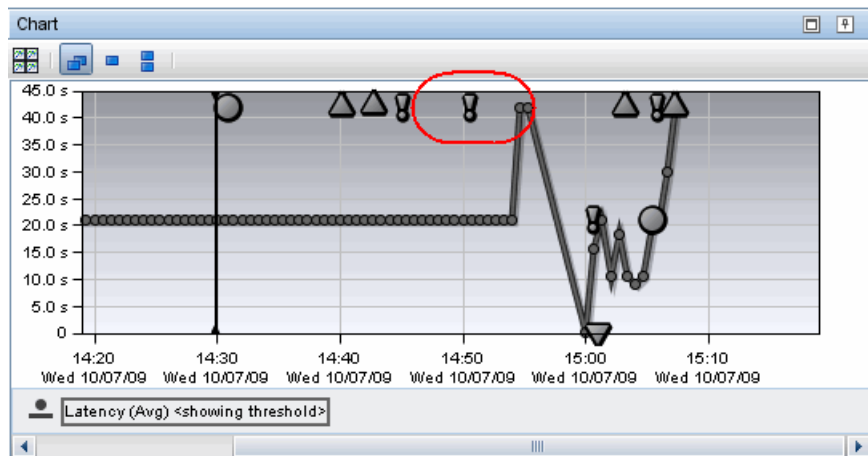
2 Find Exceptions

Information on exceptions is collected by the probe. You can find out which probes collections have encountered exceptions by looking at the Probes view. In the Probes view table you can see the number of exceptions thrown for each probe.

If you drill down from the Probes view to the server requests for this probe you will see the exceptions indicated by an instance marker on the graph. The exclamation mark icon indicates where an exception occurred. When you click the exception instance marker, a call profile opens as described in the previous step.

Example:

An example of a server request that threw an exception is shown below:



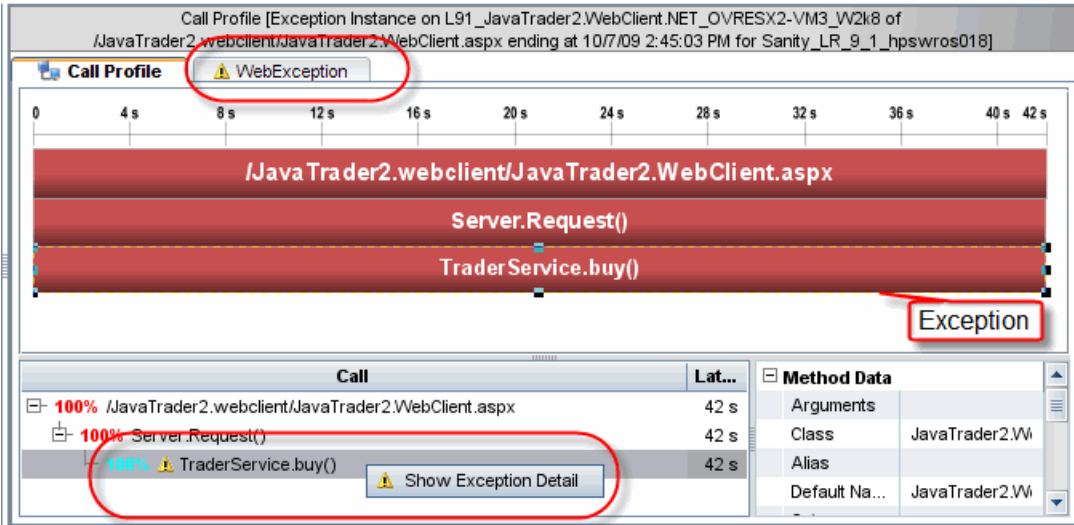
3 See Exceptions in the Call Profile

The call profile has the method that threw the exception marked with a yellow dashed outline. There are also Exceptions tabs providing details on the exceptions, including the stack trace.

In the Call Profile you can select the Exceptions tab to get details on the exception including the stack trace. Or you can right-click the exception in the call tree and select Show Exception Detail. This is useful when there are several tabs with detail data.

Example:

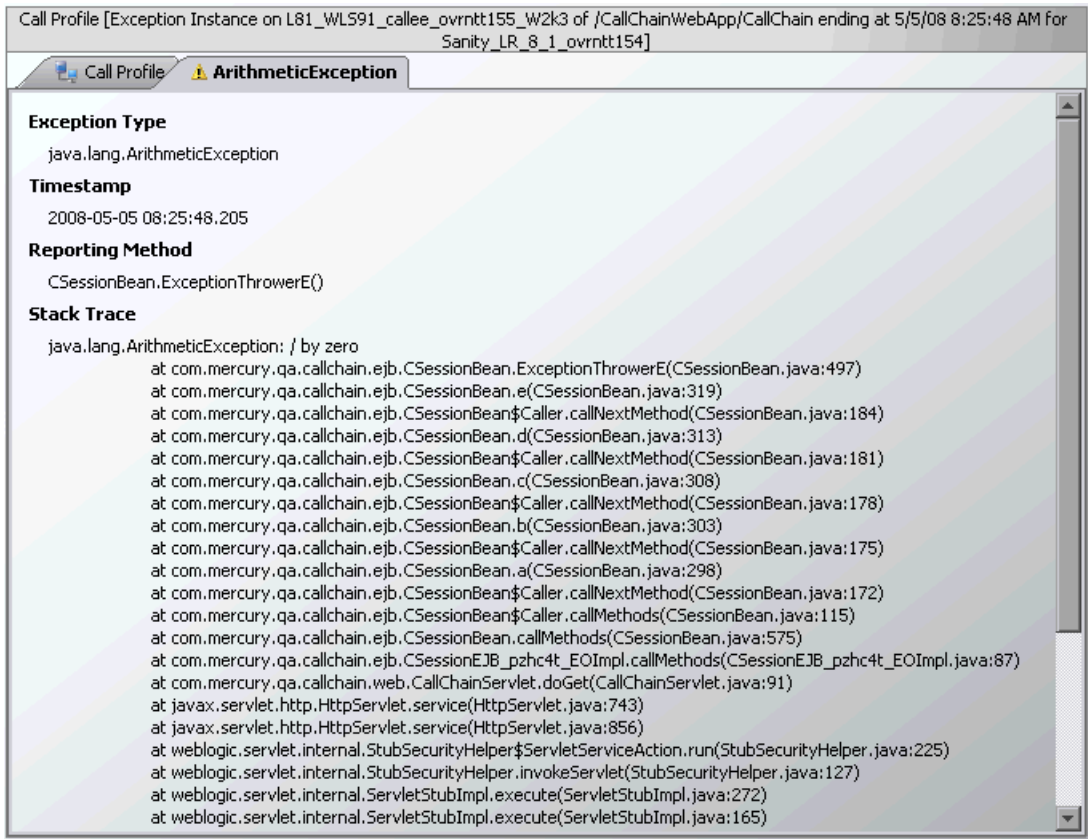
An example of a Call Profile view with the exceptions highlighted is shown below:

**4 Get More Information on Exceptions**

The Exception tab is labeled with the exception name (not including the package and class information). You can copy and paste the stack trace listing in this view to send to others for use in diagnosing a problem.

Example:

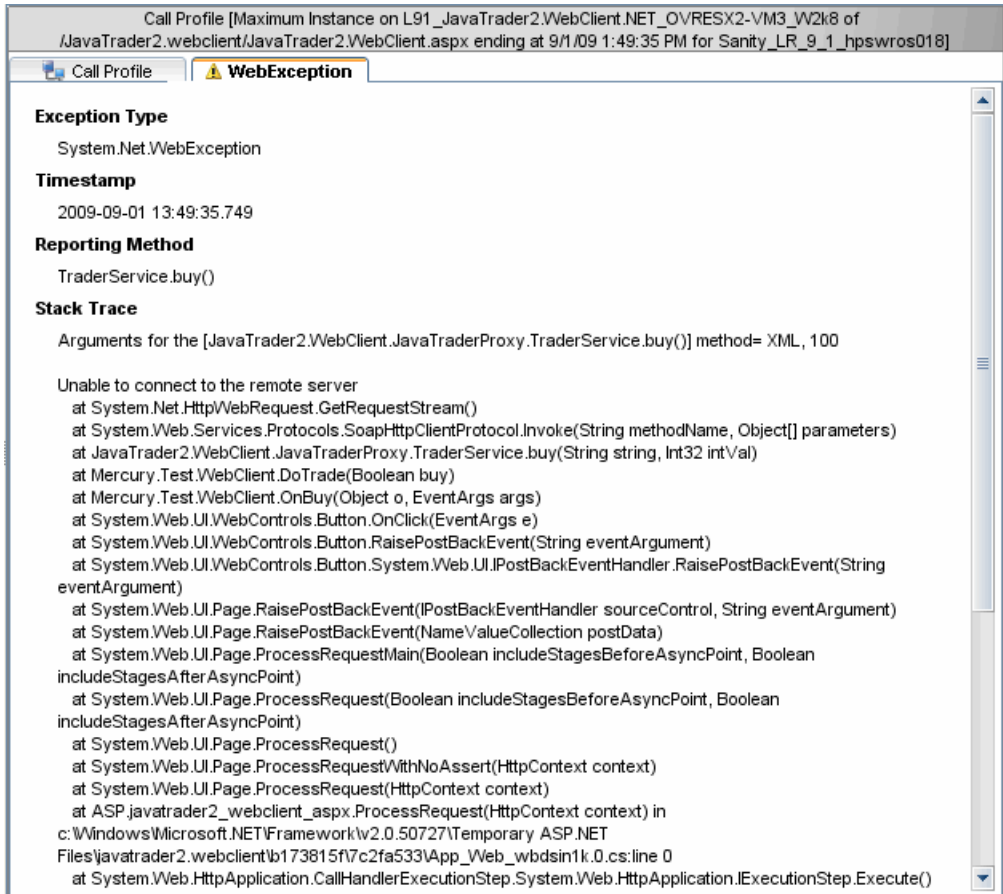
An example of the details in the Exceptions tab of the Call Profile is shown below:

**5 Arguments Displayed in the Exception Stack Trace Detail**

Note that with the .NET probe, if an exception is thrown somewhere in the call stack, arguments are captured and displayed in the Stack Trace section. Arguments are captured on the first instrumented method after the method that threw the exception. You can use the arguments to help determine what caused the exception, to identify what the inputs were. Arguments captured when an exception is thrown are separate from any argument capture you have configured for specific methods.

Example:

An example of Arguments in the Stack Trace details in the Exceptions tab of the Call Profile is shown below:



Call Profile [Maximum Instance on L91_JavaTrader2.WebClient.NET_OVRESX2-VM3_W2k8 of /JavaTrader2.webclient/JavaTrader2.WebClient.aspx ending at 9/1/09 1:49:35 PM for Sanity_LR_9_1_hpswros018]

Call Profile | **WebException**

Exception Type
System.Net.WebException

Timestamp
2009-09-01 13:49:35.749

Reporting Method
TraderService.buy()

Stack Trace
Arguments for the [JavaTrader2.WebClient.JavaTraderProxy.TraderService.buy()] method= XML, 100

```

Unable to connect to the remote server
at System.Net.HttpWebRequest.GetRequestStream()
at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters)
at JavaTrader2.WebClient.JavaTraderProxy.TraderService.buy(String string, Int32 intVal)
at Mercury.Test.WebClient.DoTrade(Boolean buy)
at Mercury.Test.WebClient.OnBuy(Object o, EventArgs args)
at System.Web.UI.WebControls.Button.OnClick(EventArgs e)
at System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument)
at System.Web.UI.WebControls.Button.System.Web.UI.PostBackEventHandler.RaisePostBackEvent(String eventArgument)
at System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument)
at System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData)
at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
at System.Web.UI.Page.ProcessRequest(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
at System.Web.UI.Page.ProcessRequest()
at System.Web.UI.Page.ProcessRequestWithNoAssert(HttpContext context)
at System.Web.UI.Page.ProcessRequest(HttpContext context)
at ASP.javatrad2_webclient_aspx.ProcessRequest(HttpContext context) in
c:\Windows\Microsoft.NET\Framework\v2.0.50727\Temporary ASP.NET
Files\javatrad2_webclient\b173815f17c2fa533\App_Web_wbdsin1k.0.cs:line 0
at System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute()

```

6 Find SOAP Faults

A SOAP fault is used to carry error information within a SOAP message. Diagnostics captures SOAP faults and classifies them into different types, based on their SOAP fault codes.

If methods throw SOAP faults, these are marked in the Server Requests view, SOA Services - Services view or the SOA Services - Operations view with an instance marker (square icon).

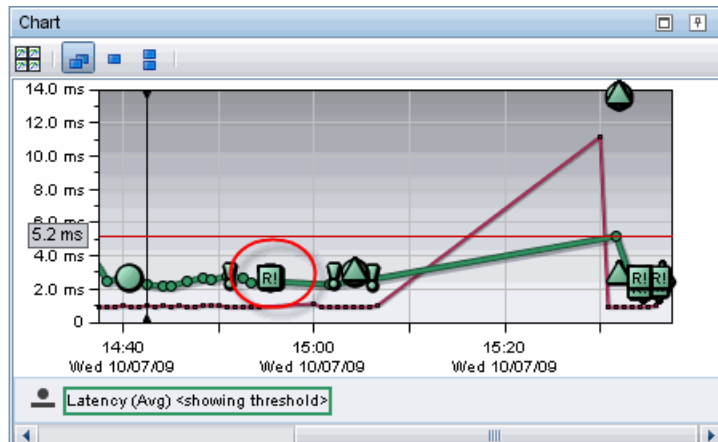
Different SOAP fault icons represent different types of SOAP fault. See "Types of SOAP Fault Markers" on page 271 for details on the icons used to represent different types of faults.

You can also see the number of SOAP faults in the details pane of any of these views. You may need to turn off the filter in the details pane to see the SOAP fault count.

When you click the SOAP fault instance marker in the Server Requests view, SOA Services - Services view or SOA Services - Operations view, a call profile opens.

Example:

An example of the SOAP fault instance markers on the Server Requests view is shown below.



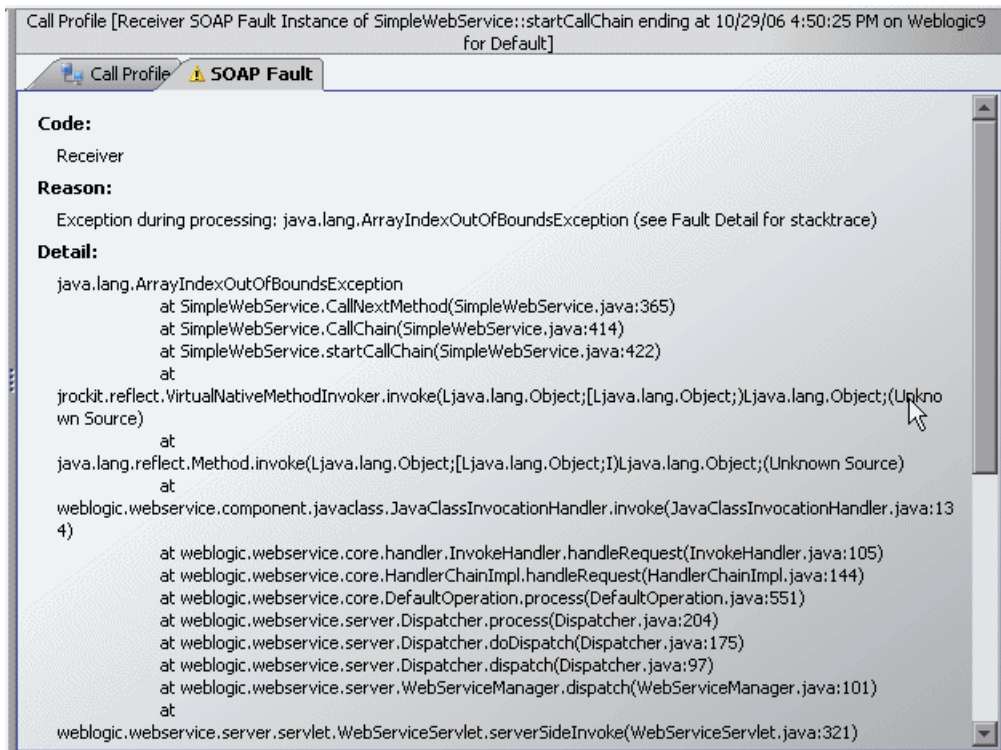
7 Get More Information on SOAP Faults (and SOAP Payload)

In the Call Profile you can select the SOAP Fault tab to get detailed information about the SOAP fault. You may also see a SOAP Payload tab if the payload was captured on the SOAP fault.

When you click the SOAP Fault tab in the Call Profile window, you can view detailed information about the SOAP fault. The SOAP Payload tab in the Call profile window includes the SOAP request information (data that the client sent).

Example:

SOAP Fault tab example is shown below:



SOAP Payload tab example is shown below:

```

Payload:
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<ns:CreateAccountCheckingRequest
xmlns:ns="http://www.openuri.org/"><CustomerInfo><FirstName>Marcus</FirstName><LastName>Johnson<
/LastName><MiddleName>Robert</MiddleName><SSN>243739841</SSN><CustomerNumber>0</CustomerNu
mber><CreationDate>2008-05-07T14:22:52.240-07:00</CreationDate><LastModifiedDate>2008-05-07T14:22
:52.240-07:00</LastModifiedDate><Status
xsi:nil="true"/></CustomerInfo><FundingInfo><Amount>500.0</Amount><CurrentDate>2008-05-07T14:22:
52.240-07:00</CurrentDate><AccountNumber>0</AccountNumber></FundingInfo><CommonHeader/></ns:C
reateAccountCheckingRequest></SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

How to do Call Profile Analysis

The Call Profile graph enables you to do the following analysis:

- ▶ Determine whether an observed latency has one cause at a certain point in the code, or many causes distributed throughout the code.
- ▶ In a multi-tier correlated diagram, determine which tier contributes the highest percentage of the total latency.
- ▶ Explore and inspect the dynamic behavior of a complex system.
- ▶ See if any unexpected exceptions affected the performance of the operation.

This task includes the following steps:

- ▶ "Compare Call Profile Graphs" on page 423
- ▶ "Analyze Java to SAP ABAP Remote Function Calls" on page 423
- ▶ "Analyze Remote Method Invocation (RMI)" on page 424

- "Analyze Lifecycle Methods for Portlets" on page 425

Compare Call Profile Graphs

The Call Profile view enables you to compare two or more call profiles.

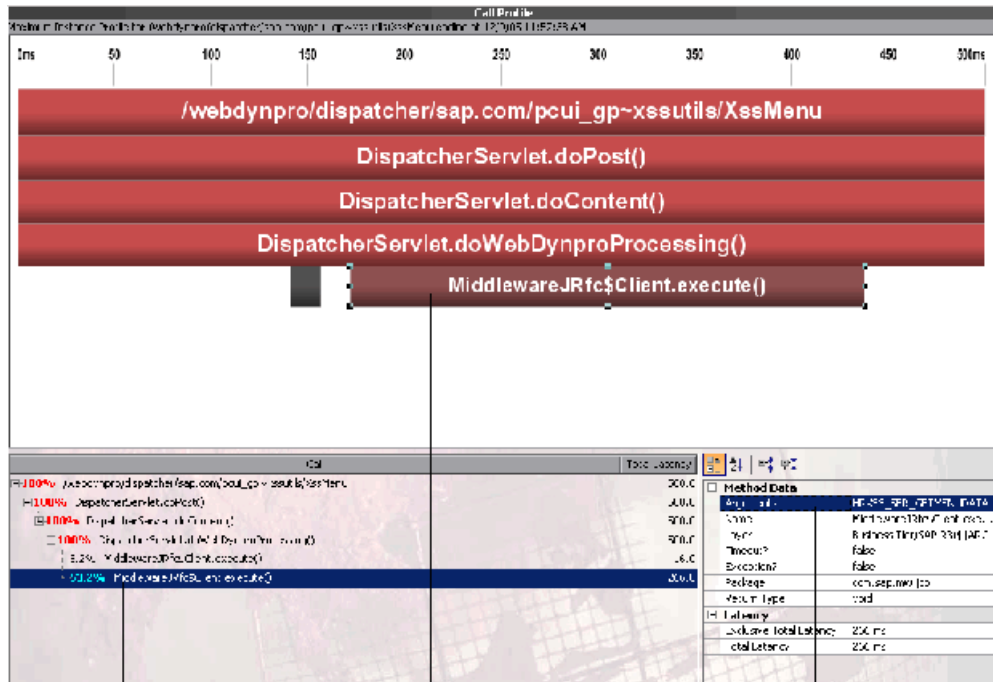
When you right-click an instance tree marker, the **Open Call Profile in New Window** menu option is displayed. Selecting this option causes the Call Profile view to be displayed in a new window without the **View Bar**. You may then return to the original window and navigate to select another instance tree to display in the Call Profile view. Once the new tree is displayed, you can compare it with the tree that is displayed in the alternate window.

Analyze Java to SAP ABAP Remote Function Calls

The Remote Function Call (RFC) protocol in SAP allows communication to take place between SAP Java and SAP ABAP environments. The remote calls that are made between SAP Java and SAP ABAP environments are displayed in the Call Profile view, as shown in the following example.

When a Java method executes an RFC, it specifies the SAP function as an argument in the RFC. Diagnostics displays the RFC as a child box under the service request in the Call Profile graph, and lists it as a child of the service request in the Call Tree table. When the RFC is selected from the Call Tree table, the details for the call are displayed in the Details pane.

Example:



Remote Function Call

Remote Function Call

R3 Function Requested by RFC

Analyze Remote Method Invocation (RMI)

When analyzing RMI in the Call Profile view, you must be aware that the latency displayed for the remote caller in the Call Profile graph and the Call Tree table includes the processing time for the remote callee. The remote callee is also displayed separately in the Call Profile graph and Call Tree table with just the latency for its own processing.

Analyze Lifecycle Methods for Portlets

Life cycle methods for portlets are identified by the name of the method (beginRender, endRender, and so on), and the portlet on which the method was invoked. The portlet name is a combination of its title and label.

Reference

Call Profile User Interface

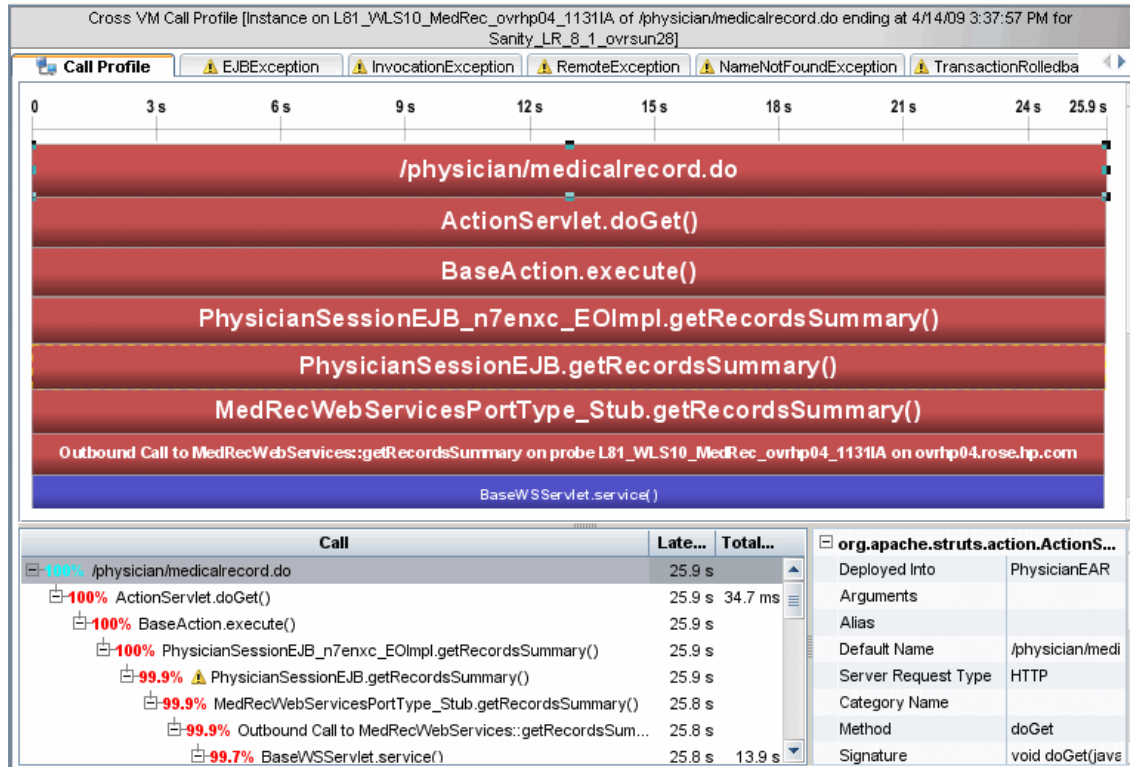
This section includes:

- ▶ [Call Profile View Description on page 427](#)

Call Profile View Description

The **Call Profile** view displays the method calls and their latency for a particular instance of a monitored server request in your application. The Call Profile view displays the metrics in a graphical call profile, and in a call tree table.

The following is an example of the Call Profile view.



<p>To access</p>	<p>The Call Profile view cannot be accessed directly from the Standard View bar. You can access the Call Profile view by drilling down from the Server Request view. Click an instance marker icon in the Server Requests graph to open the Call Profile View.</p>
<p>Important information</p>	<p>It is possible that the instance tree that is displayed for an instance tree marker on the Server Requests view is different than the one that was anticipated. This is because the instance tree on the Diagnostics Server may have been replaced since the last update to the UI. So, even though a specific tree was selected, another tree that matches the type of the selected instance tree is displayed.</p> <p>You may not save a Call Profile view as a custom view. When you use the Call Profile view, you must configure the view to customize the displayed information each time you access the view.</p> <p>The Details pane in the Call Profile view does not enable setting thresholds, adding to a snapshot, or charting other metrics.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Call Profile" on page 414.</p>

Call Profile Content

The following user interface elements are included:

UI Elements	Description
Call Profile Graph	<p>Displays the method calls that make up a server request in graphical format. Shows the latency of each call.</p> <p>The legend across the top of the graph denotes the amount of time, in seconds, since the server request was started.</p> <p>The x-axis of the call profile represents elapsed time where time progresses from left to right. The calls are distributed across the x-axis based on when they took place, and sequence of the calls relative to each other.</p> <p>The y-axis on the call profile represents the call stack, or nesting level. The calls made at the higher levels of the call stack are shown at the top of the profile. Calls made at deeper levels of the call stack are shown at the lower levels of the profile.</p> <p>For information on what the boxes in the call profile graph represent, see "Call Profile Graph" on page 407.</p> <p>When you click a call box in the Call Profile graph, the corresponding row is selected in the Call Tree table, and the metrics for the selected call are displayed in the Details pane.</p> <p>The tooltip displays the following call details:</p> <ul style="list-style-type: none"> ▶ The method call ▶ The Diagnostics layer where the call occurred ▶ The percentage contribution to the total latency represented by the call ▶ The total latency of the call

UI Elements	Description
Call Tree Table	<p>Presents the information from the Call Profile graph in a tabular format.</p> <p>For information on interpreting the information in the call tree, see "Call Tree Table" on page 408.</p> <p>The following columns are displayed:</p> <ul style="list-style-type: none"> ▶ Call. The server request call or method call whose performance metrics are reported in the row. Preceding the call name is the percentage contribution of the method call to the total latency of the service request. ▶ Latency (Total). The total latency for the call. The time is reported in appropriate units. The total latency for a parent call includes not only the sum of the latency of each of its children, but also the latency for the processing that the method did on its own. ▶ Total CPU. The time spent by the CPU executing this call.
Details Pane	The Details pane lists the metrics for the call that is selected in the Call Tree table and the Call Profile graph.

Exceptions Tab Content

The Exception tab is labeled with the exception name (not including the package and class information).

The following user interface elements are included:

UI Elements	Description
Exception Type	The type of exception.
Timestamp	Time the exception occurred.
Reporting Method	Method reporting the exception.
Stack Trace	<p>You will see arguments captured on the method that threw the exception.</p> <p>You can copy and paste the stack trace listing.</p>

SOAP Fault Tab Content

The following user interface elements are included:

UI Elements	Description
Code	The type of SOAP fault.
Reason	The reason that the SOAP fault was generated.
Detail	Either the exception stack trace or other detailed information contained in the generated SOAP fault.

SOAP Payload Content

The SOAP Payload tab in the Call profile window includes the SOAP request information (data that the client sent).

Payload capture on SOAP faults is enabled by default. But you can configure this for trimming.

Capturing SOAP payload requires the Diagnostics SOAP message handler. See "Configure Collection of SOAP Faults and Payloads" on page 409 for more information.

17

Probes View

This chapter describes the Probes view.

This chapter includes:

Concepts

- ▶ Probes Overview on page 434

Tasks

- ▶ How to Use the Probes View on page 436
- ▶ How to Configure Instrumentation Points for Method Trending on page 437

Reference

- ▶ Probes User Interface on page 439

Concepts

Probes Overview

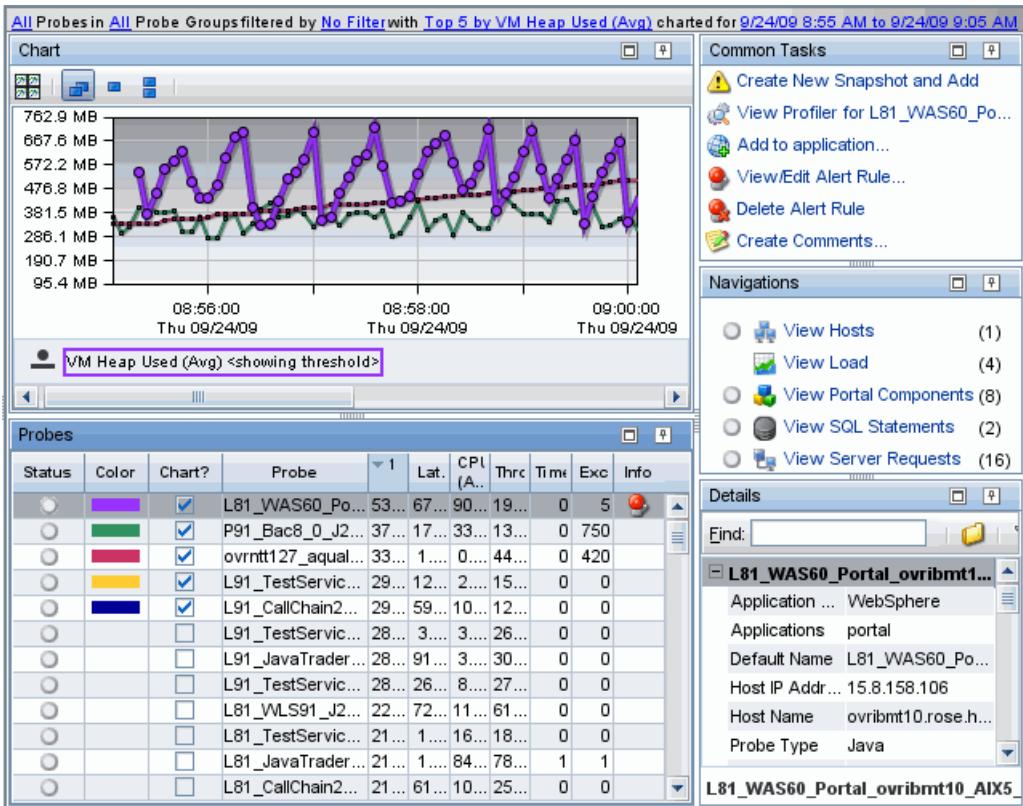
The **Probes** view displays performance metrics for the probes that are monitoring your applications.

For details on the user interface, see "Probes View Description" on page 440.

Note: Diagnostics continues to display information about a probe or probe group for as long as the data remains valid. If, for example, a probe disconnects, it disappears from the view after several minutes. A probe will continue to appear in the table for as long as a week, while Diagnostics continues to report on the probe's availability.

By default, the Probes view charts the heap usage for the five probes that have the highest average heap usage during the selected time range. The table lists the probes related to the context shown in the breadcrumb above the graph. Which probes are listed in the table depends on the context selected. So for example, if you navigated to the Probes view from the View bar, the Probes view shows all of the probes. If you navigated to the Probes view from the Hosts view, the Probes view lists only the probes that were installed on the selected host.

An example of the Probes view is shown below:



Tasks

How to Use the Probes View

Using the information displayed in the Probes view, you can isolate performance problems to applications monitored by a particular probe. If the information displayed in the Probes view raises any concerns, you can drill down to find out more information from a more detailed view of the underlying performance metrics.

This task includes the following steps:

- ▶ "Drill Down to Server Requests and other views from Probes View" on page 436
- ▶ "Drill Down to the Profiler from Probes View" on page 437

Drill Down to Server Requests and other views from Probes View

When you drill down from a probe, Diagnostics displays the Server Requests view with more details for the server requests collected by the selected probe.

Other navigation links and common tasks are available for a probe depending on the type of data being monitored. Some possible navigation links include: View Hosts, View Load, View Portal Components, View Outbound Calls and View Services.

If the application the probe is monitoring has an SQL statement that exceeds the predefined SQL trending threshold then you can drill down to View SQL Statements by probe.

If the application the probe is monitoring has collection leaks and collection leak pinpointing is enabled for the probe you can drill down to View Collection Leaks.

If you have configured instrumentation points for method trending in the application being monitored by a probe you can drill down to View Trended Methods by probe.

Some common tasks you may want to perform in analyzing performance problems include: View Profiler for the probe and Create New Snapshot.

Drill Down to the Profiler from Probes View

You can drill down from a probe to the Profiler. The Profiler launches in a separate window and provides additional information for you to use in analyzing a performance problem.

In the .NET Profiler you can see detailed performance data for the .NET probe.

In the Java Profiler you can get real time data for the Java probe such as:

- ▶ Threads data
- ▶ Allocation/Lifecycle analysis data
- ▶ Heap breakdown data and a help walker

How to Configure Instrumentation Points for Method Trending

To see data in the Trended Methods view, configure the instrumentation point for the method so that the latency for this method gets captured. This data can then be displayed as a trend line in the Trended Methods view.

Caution: By default, this view is empty. For a method to be displayed in this view, it needs to be configured.

To configure a method to be displayed in the Trended Methods view:

- 1** Open the relevant capture points file.

- ▶ By default, the Java Probe capture points file is located at `<probe_install_dir>\etc\auto_detect.points`.
 - ▶ For the .NET Probe, the capture points files are located in the `<.NET probe_install_dir>\etc\` directory.
- 2 In the capture points file, locate or add the point that includes the method you want to display in the Trended Methods view.
 - 3 Add the following argument to the point definition:

```
layertype = trended_method
```

- 4 Restart the application server on which your probe is running.

Caution:

Configuring methods to be displayed in the Trended Methods view might add significant overhead to the Diagnostics system.

If you configured the probe to capture arguments for a point that is also configured for method trending, each unique argument will result in a unique trended method. This can have a significant impact on memory overhead.

Trended methods are not latency trimmed. This means that any frequently-executing method with low latency could cause a noticeable performance impact on the application.

Reference

Probes User Interface

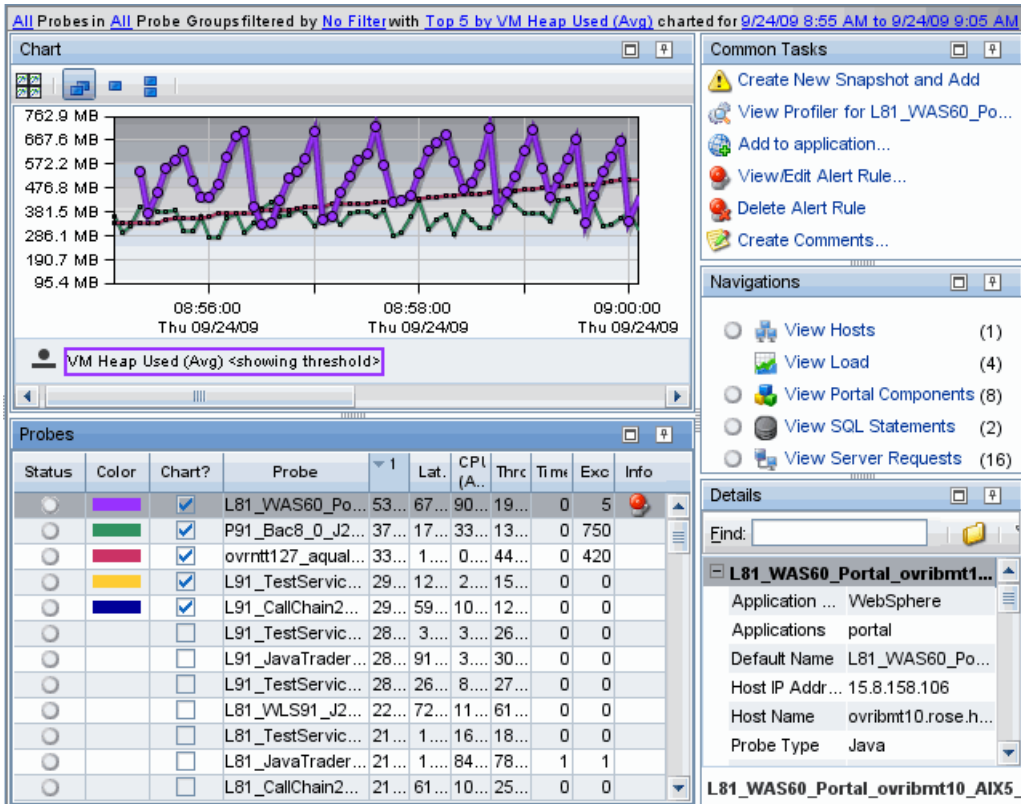
This section includes:


- ▶ Probes View Description on page 440
- ▶ Java Probes View Description on page 445
- ▶ Trended Methods (by probe) View Description on page 450
- ▶ Collection Leaks View Description on page 453

Probes View Description

The Probes view shows performance metrics for the probes monitoring the selected application. The .NET Probes view shows the same information for .NET probes only.

The following is an example of the Probes view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Probes.</p> <p>You can also access this view directly from the Applications window by selecting the link to Java Probes or .NET Probes under Application Environments in the Application Contents pane.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p> <p>Diagnostics continues to display information about a probe or probe group for as long as the data remains valid. If, for example, a probe disconnects, it disappears from the view after several minutes. A probe will continue to appear in the table for as long as a week, while Diagnostics continues to report on the probe's availability.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Probes View" on page 436 for information on how to analyze performance using the data in this view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average VM heap used, as a trend line. By default, the five probes that have the highest average heap usage during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe or probe group.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the heap usage amount in megabytes (mb) or other related units, if more appropriate.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <ul style="list-style-type: none"> ➤ Probe. The name of the probe collecting the metrics represented. ➤ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ➤ Host. The host on which the probe is running. ➤ Probe Type. Java, .NET, Oracle, SAP, MQ, SQL Server. <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

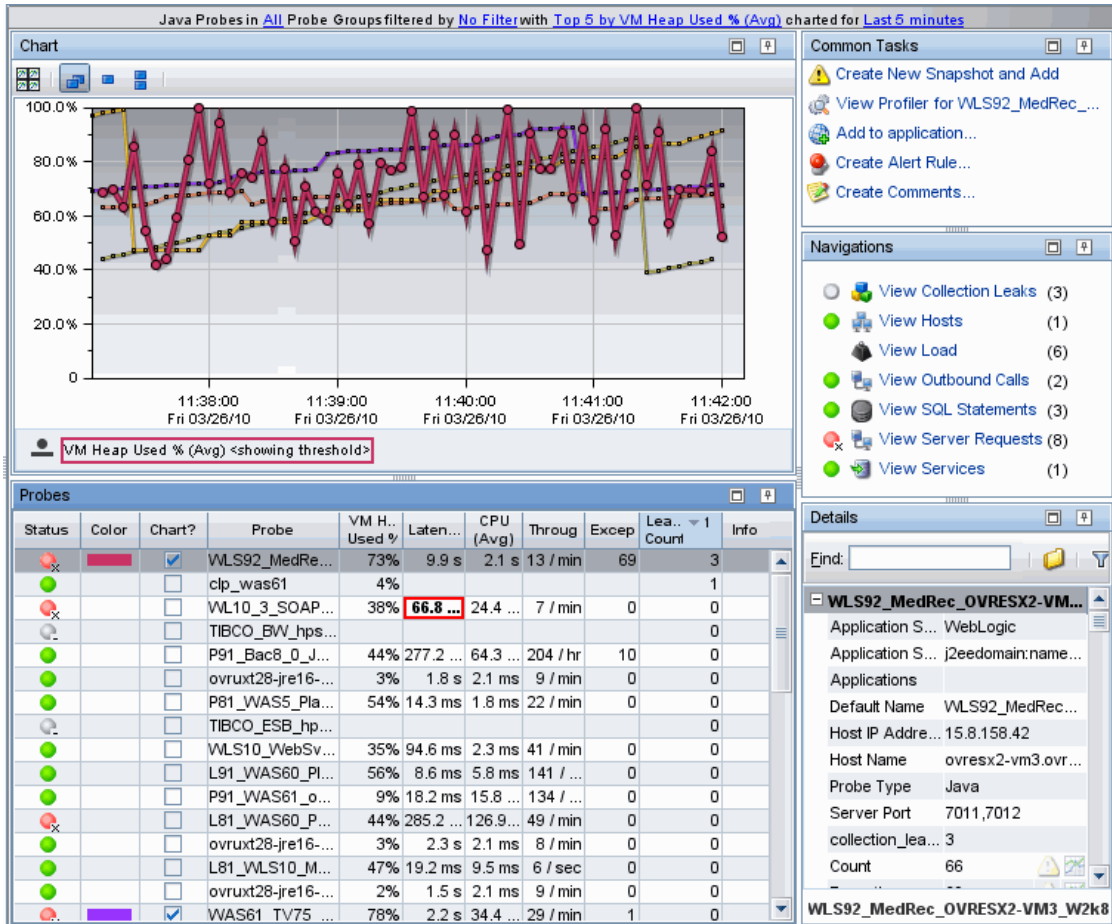
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>VM Heap. The average amount of heap used by the selected application being monitored by this probe.</p> <p>Latency. The average response time across all instances.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>


UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigations available vary depending on the type of application being monitored by the probe and the data collected as defined by the instrumentation. Common drill down navigations for a probe include: View Server Requests and View Profiler for the probe.</p> <p>A drill down to the SQL Statements view may also be available if this data was collected by the probe.</p>
<p>Details Pane</p>	<p>The Details pane shows metrics for the probes monitoring the selected application.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Java Probes View Description

The Java Probes view shows performance metrics for the Java probes monitoring the selected application.

The following is an example of the Java Probes view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Java Probes.</p> <p>You can also access this view directly from the Applications window by selecting the link to Java Probes under Application Environments in the Application Contents pane.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p> <p>Diagnostics continues to display information about a probe or probe group for as long as the data remains valid. If, for example, a probe disconnects, it disappears from the view after several minutes. A probe will continue to appear in the table for as long as a week, while Diagnostics continues to report on the probe's availability.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Probes View" on page 436 for information on how to analyze performance using the data in this view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average VM heap used percentage, as a trend line. By default, the five probes that have the highest average heap used percentage during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe or probe group.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the heap used percentage.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <ul style="list-style-type: none"> ▶ Probe. The name of the probe collecting the metrics represented. ▶ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ▶ Host. The host on which the probe is running. ▶ Probe Type. Java probe. <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

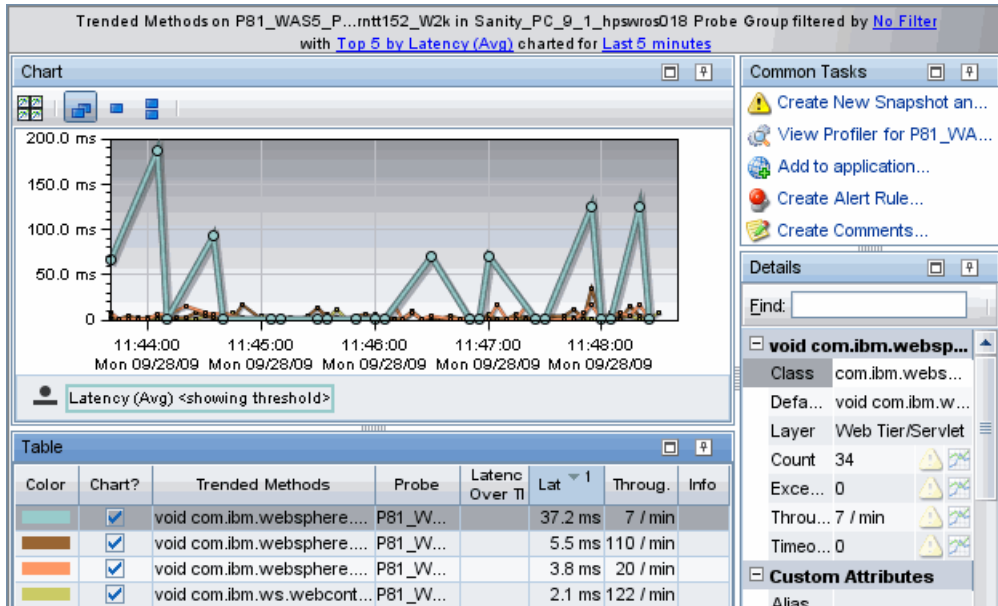
UI Elements	Description
Table	<p>Lists the Java probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the Java probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>VM Heap Used %. The average percentage of heap used by the selected application being monitored by this probe.</p> <p>Latency. The average response time across all instances.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>Leak Counts. The number of detected leaking collections. If the collection leak pinpointing feature is not enabled in a probe, this field will be blank. If no collection leaks are detected, this field will show 0.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigations available vary depending on the type of application being monitored by the probe and the data collected as defined by the instrumentation. Common drill down navigations for a probe include: View Server Requests and View Profiler for the probe.</p> <p>A drill down to the SQL Statements view or Trended Methods view may also be available if this data was collected by the probe.</p> <p>If collection leaks were detected, you may also drill down to the Collection Leaks view.</p>
<p>Details Pane</p>	<p>The Details pane shows metrics for the probes monitoring the selected application.</p> <p>Refer to the <code><probe_install_dir>/etc/metrics.config</code> file on the Java Agent system for details on metrics collected.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Trended Methods (by probe) View Description

This page shows performance metrics for the methods you've configured for trending for a probe. You must have configured the instrumentation point for the method so that latency for the method gets captured by the probe for you to see any data in this view.

The following is an example of the Trended Methods view for a probe.



To access

Access the Trended Methods view for a probe by drilling down from a probe (right-click a probe and select View Trended Methods).

Important information	By default, this view is empty. For a method to be displayed in this view, an instrumentation point needs to be configured so that the latency for the method gets captured by the probe.
Relevant tasks	For details on how to configure an instrumentation point for a method trending, see "How to Configure Instrumentation Points for Method Trending" on page 437.

Common Elements

User interface elements are described below:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
View title bar and view filters	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

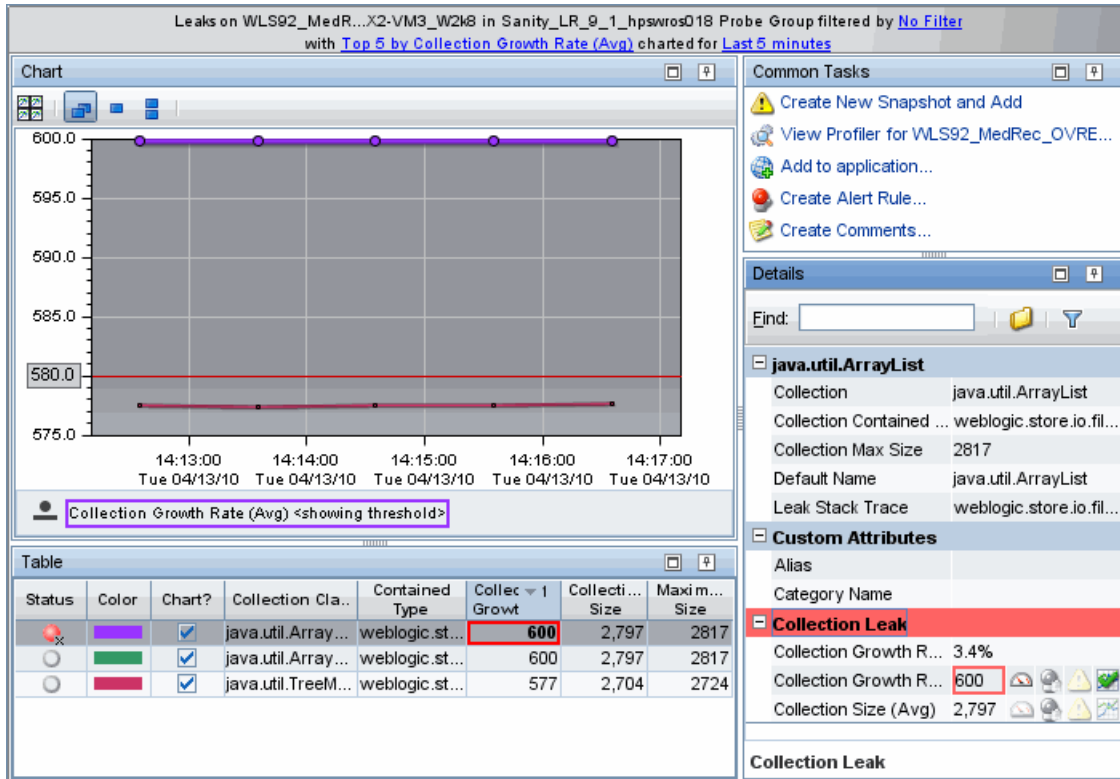
UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five trended methods that have the highest average latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Method Details tooltip for the trend line on the graph (or the Trended Methods column in the table) displays the following detail data:</p> <p>Method Name. The name of the trended method.</p> <p>Probe. The name of the probe in which the method was executed.</p> <p>Probe Group. The name of the probe group containing the probe.</p> <p>Class. The class associated with the executing method.</p> <p>Layer. The layer in which the method executed.</p> <p>The Method Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the trended methods that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Trended Methods. The name of the method.</p> <p>Probe. The name of the probe.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	Common tasks and actions relevant to the trended methods displayed in this view.
Details Pane	Shows metrics collected on trended methods. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

Collection Leaks View Description

This page shows information about detected memory leak locations in Java applications

The following is an example of the Collection Leaks view for a Java probe.



To access	Access the Collection Leaks view for a probe by drilling down from a probe (right-click a probe and select View Collection leaks).
Important information	Collection leak pinpointing must be enabled for this view and is enabled by default. The clp.diagnostics.reporting point in the probe's etc/auto_detect.points file is set to true by default. Also note that for this feature you need to enable CLP instrumentation by running the Diagnostics JRE instrumenter to pre-instrument the Java Collection classes in the JRE your application/application server will run with; and copy the java parameter to include it in your java options.
Relevant tasks	In the Collection Leaks view you can see a stack trace at the detected leak location by clicking on the Leak Stack Trace value in the Details pane.

Common Elements

User interface elements are described below:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
View title bar and view filters	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average collection growth rate, as a trend line. By default, the five collection classes that have the highest average collection growth rate during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. With the default metrics selection the y-axis of the graph shows the average collection growth rate in number of elements per hour.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Collection Leak Details tooltip for the trend line on the graph (or the Collection Classes column in the table) displays the following detail data:</p> <p>Collection Leak. Class name of the leaking collection.</p> <p>Probe. The name of the probe.</p> <p>Probe Group. The name of the probe group containing the probe.</p> <p>The Collection Leak Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the collection classes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the collection class. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Collection Class. Class name of the leaking collection.</p> <p>Contained Type. The collection contained type if it could be determined.</p> <p>Collection Growth Rate. The average hourly size change of the leaking collection.</p> <p>Collection Size. The average size of the leaking collection.</p> <p>Maximum Size. The observed maximum size of the collection.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the leaking collection classes displayed in this view.
Details Pane	Shows metrics collected on leaking collections: Collection Growth Rate % Over Threshold Collection Growth Rate Collection Size (Avg) You can click on the Leak Stack Trace value in the Details pane to open a window containing the stack trace at the detected leak location. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

18

Collections and Resources Views

This chapter describes the Collections view and the Resources view.

This chapter includes:

Concepts

- ▶ Collections and Resources Overview on page 460

Tasks

- ▶ How to Enable LWMD for a .NET Probe on page 461
- ▶ How to Enable LWMD for a Java Probe on page 461
- ▶ How to Use the Collections and Resources Views on page 463

Reference

- ▶ Collections and Resources User Interface on page 464

Concepts

Collections and Resources Overview

The **Collections** view and the **Resources** view show the collections in your system and the resource allocating the collections.

Note: The Collections view is available for both the Java and .NET probe. The Lightweight Memory Diagnostics (LWMD) feature must be enabled for the probe for this view to be available.

The Resources view is only available for the Java probe, provided Lightweight Memory Diagnostics (LWMD) is enabled for the probe.

The Collections view displays the collections in your system as well as the allocation point for each collection. The metrics available are minimum, maximum and average size of the collection. By default, the average size is charted.

The Resources view displays the resources that are allocating the collections (for example, a resource might be a .jar file). For each resource, the following metrics are available:

- ▶ Largest collection size
- ▶ Number of collection instantiations
- ▶ Total elements

Tasks

How to Enable LWMD for a .NET Probe

In order for the Collections view to be available, LWMD must be enabled for a .NET probe. When the .NET probe is installed, the default configuration is to have LWMD turned off. To enable the LWMD feature you must perform two modifications to the `probe_config.xml` file:

- ▶ You must enable the `<lwmd>` element (see `<lwmd>` element in the *HP Diagnostics Installation and Configuration Guide* chapter on understanding the .NET probe configuration file).
- ▶ You must add one or more references to the **Lwmd.points** file as described in the *HP Diagnostics Installation and Configuration Guide* chapter on Advanced .NET Agent Configuration.

Note: Enabling the probe to capture collections metrics could incur additional overhead on the host for an application (see the *HP Diagnostics Installation and Configuration Guide* chapter on Advanced .NET Agent Configuration for information on Controlling LWMD Instrumentation).

You can also use the .NET profiler for more detailed collections analysis. See Chapter 38, "Diagnostics Profiler for .NET" for more information.

How to Enable LWMD for a Java Probe

In order for the Collections view or the Resources view to be available, Lightweight Memory Diagnostics (LWMD) must be enabled for the Java probe.

To enable LWMD for the Java probe:

- 1 Set the property `lwm.diagnostics.capture=true` in the `<probe-install-dir>/etc/dynamic.properties` file.

- 2 Set **active=true** (for lwmd) in the **<probe-install-dir>/etc/auto_detect.points** file.

Additional configuration is as follows:

- It may be desirable not to show all the collections in the Collections view. This can be configured using property **lwm.diagnostics.top.n** (in **<probe-install-dir>/etc/dynamic.properties**). For example, if the value of this property is set to 10, only the top 10 collections will be reported. However you may see more than 10 collections since both the top N by size as well as the top N by growth are being reported.
- The resources reported can be controlled via regular expressions using the property **lwmd.resource.replace** (in **<probe-install-dir>/etc/dynamic.properties**). This property allows for pattern replacement of resources. The default regular expressions ensure two transformations.
 - Transformation 1 example:
From: jar:file:/C:/bea/weblogic81/common/eval/pointbase/lib/pbclient44.jar!/com/pointbase/net/netJDBCCConnection.class
To: jar:file:/C:/bea/weblogic81/common/eval/pointbase/lib/pbclient44.jar
 - Transformation 2 example:
From: file:/C:/bea/weblogic81/classes/com/pointbase/net/netJDBCCConnection.class
To: Exploded Classes
 - More transformations can be specified as desired.

For instructions on setting properties for the probes, see the *HP Diagnostics Installation and Configuration Guide*.

You can also use the Java profiler for more detailed memory use analysis. See Chapter 37, "Diagnostics Profiler for Java" for more information.

How to Use the Collections and Resources Views

A first look at the Resources view gives an understanding of which resource is the most expensive in terms of collection allocations. You can then move to the collections view (or the Diagnostics Profiler) to find the ‘responsible’ collections within that resource.

Analysis in the Collections View:

If a collection object (any subclass of `java.util.Collection` for Java or `System.Collections.ICollection` or `System.Collections.Generic.ICollection` for .NET) starts utilizing larger amounts of memory, the Collections view helps isolate the problem by showing which collection object is taking up the memory and where it is allocated.

Using the information displayed in the Collections view, you can get an immediate understanding of the largest collections in your system as well as where those collections are being allocated (class, method, line number).

Analysis in the Resources View:

The Resources view shows the resource that is allocating a collection object, whether it is part of a zip file, jar file, or other kinds of archives, or directly on the file system. You can see the total number of collection instantiations, largest collection size, and total memory occupied by collection objects.

Using the information displayed in the Resources view, you can get an immediate understanding of which resource is allocating the most collections. You also can see which resource holds the collection with the largest size, as well as the resource with the largest number of elements.

Reference

Collections and Resources User Interface

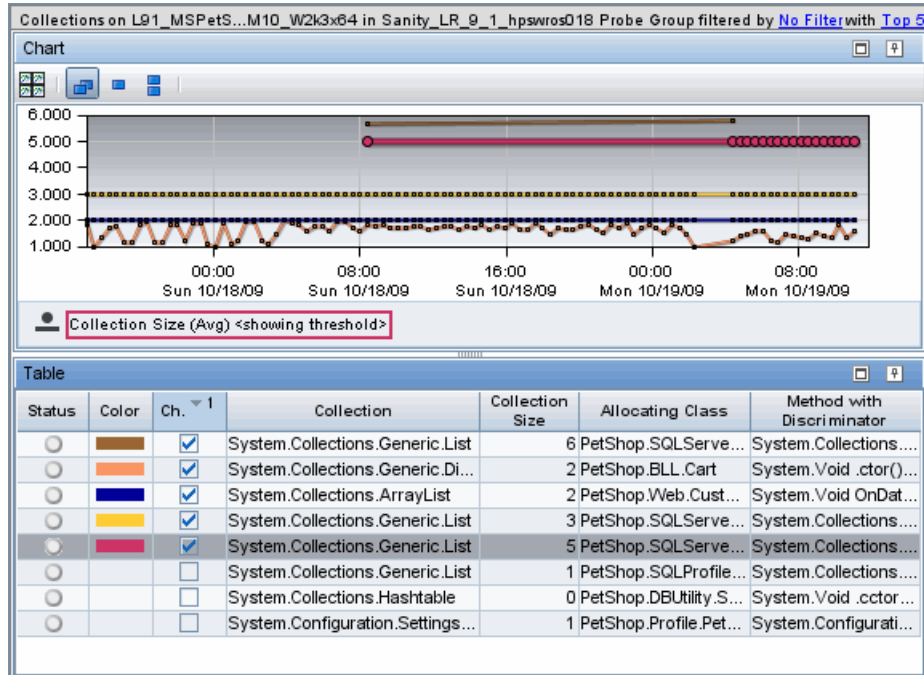
This section includes:

- ▶ Collections View Description on page 465
- ▶ Resources View Description on page 470

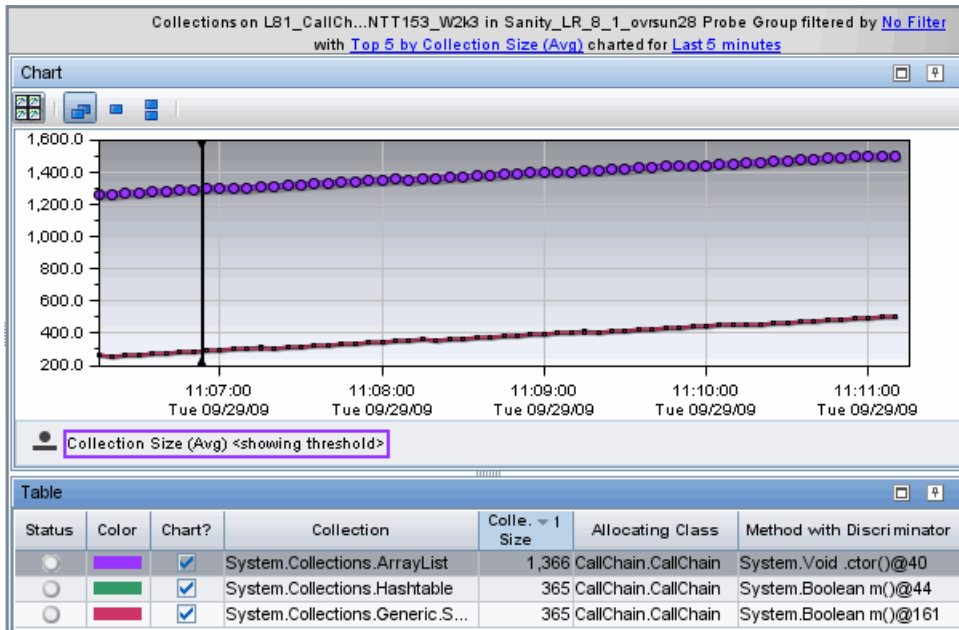
Collections View Description

The Collections view displays the top N collections by size as well as the top N collections by growth during the currently selected time frame.

The following is an example of the Collections view for Java.



The following is an example of the Collections view for .NET.



To access

The Collections view cannot be accessed directly from the Standard View bar because collections and resources are always displayed in the context of a probe.

You can access the Collections view by drilling down from a probe in the Status or summary views or in the Probes view. Right click the probe and select **View Collections**.

You can also access this view directly from the Applications window by selecting the link to Collections under Code in the Application Contents pane.

Important information	The Collections view is available for the .NET probe and the Java probe. Lightweight Memory Diagnostics must be enabled in order for the view to be available.
Relevant tasks	<ul style="list-style-type: none"> ➤ "How to Enable LWMD for a .NET Probe" on page 461 ➤ "How to Enable LWMD for a Java Probe" on page 461 ➤ "How to Use the Collections and Resources Views" on page 463

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

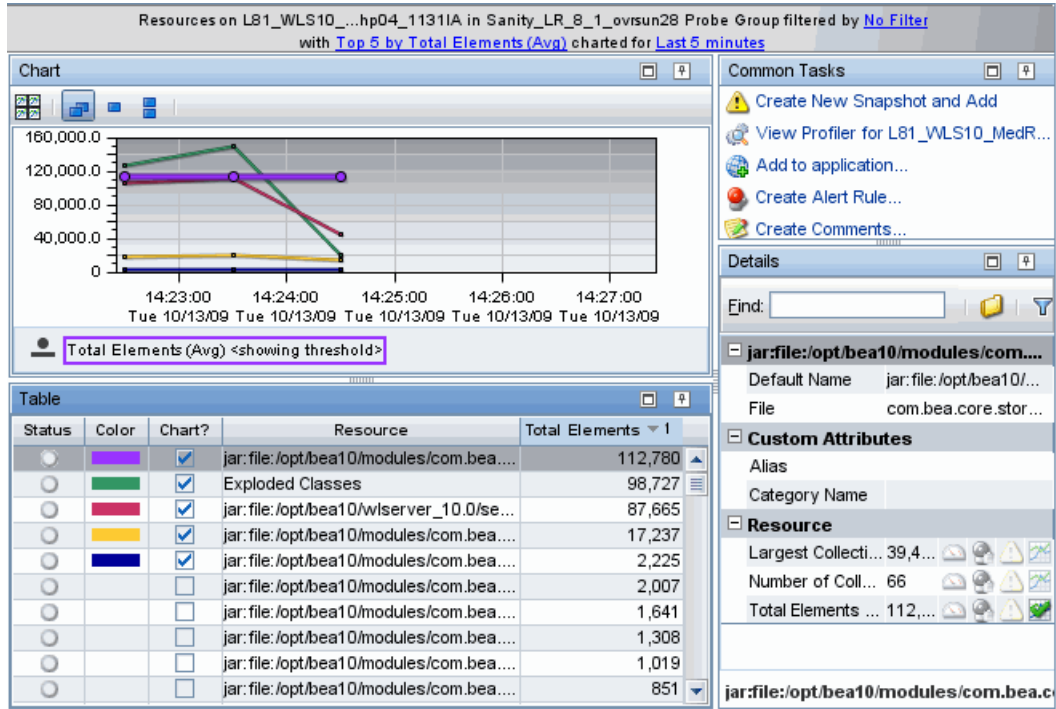
UI Elements	Description
Chart	<p>Displays a graph of the average collection size, as a trend line. By default, the five collections that have the largest collection size during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average collection size.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Collection Details tooltip for the trend line on the graph (or the Collections column in the table) displays the following data:</p> <ul style="list-style-type: none"> ➤ Collection and allocation point. This is made up of all four items below (concatenated). ➤ Collection. The name of the collection. ➤ Allocating Class. The class where this collection is allocated. ➤ Allocating Method. The method where this collection is allocated. ➤ Allocating Method with Discriminator. In Java this is the line number where the collection is allocated. In .NET this is the byte offset from the start of the method where the collection is allocated. ➤ Line Number. (Java only) The line number where this collection is allocated. <p>The Collection Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the collections that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the collection. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Collection. The name of the collection.</p> <p>Collection Size. The size of the collection.</p> <p>Allocating Class. The class where this collection was allocated.</p> <p>Method with Discriminator. In Java this is the line number where the collection is allocated. In .NET this is the byte offset from the start of the method where the collection is allocated.</p>
Common Tasks and Navigations	Common tasks relevant to the collections displayed in this view.
Details Pane	<p>The Details pane shows metrics for the collections found by the selected probe including:</p> <p>Collection Size - Average, Maximum and Minimum.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Resources View Description

The Code Resource view displays the average number of elements for each resource.

The following is an example of the Resources view.



To access	<p>The Resources view cannot be accessed directly from the Standard View bar because collections and resources are always displayed in the context of a probe.</p> <p>You can access the Resources view by drilling down from a probe in the Status or summary views or in the Probes view. Right click the probe and select View Resources.</p> <p>You can also access this view directly from the Applications window by selecting the link to Code Resources under Code in the Application Contents pane.</p>
Important information	<p>The Resources view is only available for the Java probe. Lightweight Memory Diagnostics must be enabled in order for the view to be available.</p>
Relevant tasks	<ul style="list-style-type: none"> ➤ "How to Enable LWMD for a Java Probe" on page 461 ➤ "How to Use the Collections and Resources Views" on page 463

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average number of elements as a trend line. By default the five resources that have the highest average number of elements during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average number of elements for each resource.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>The Resource Details tooltip for the trend line on the graph (or the resources column in the table) displays the following detail data:</p> <ul style="list-style-type: none"> ➤ Resource The resource name. ➤ Probe. The name of the probe. ➤ Probe Group. The name of the probe group. <p>The Resource Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node including sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the resources that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the resource. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart? From the table you can select the entities that are to be charted on the graph.</p> <p>Resource. Name of the resource.</p> <p>Total Elements. The total number of elements for the resource.</p>
Common Tasks and Navigations	Common tasks relevant to the resources displayed in this view.
Details Pane	<p>The Details pane shows metrics for the resource found by the selected probe including:</p> <p>Largest Collection Size</p> <p>Number of Collection Instantiations</p> <p>Total Elements</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

19

Outbound Calls View

This chapter describes the Outbound Calls view.

This chapter includes:

Concepts

- ▶ Outbound Calls Overview on page 476

Tasks

- ▶ How to Use the Outbound Calls View on page 478

Reference

- ▶ Outbound Calls User Interface on page 479

Concepts

Outbound Calls Overview

The **Outbound Calls** view shows calls that go to a different process.

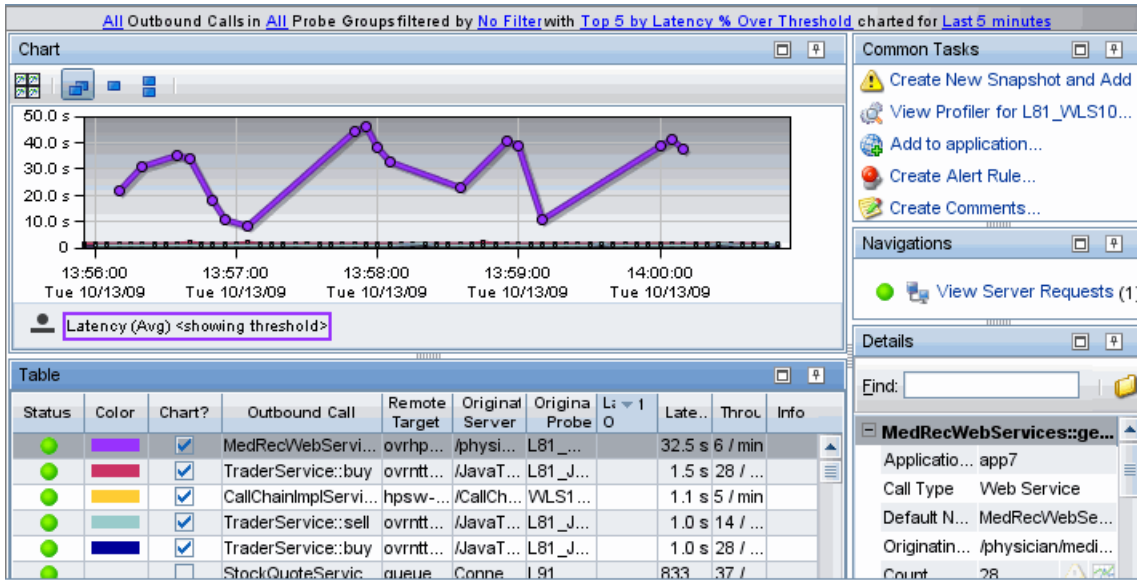
Diagnostics monitors the following types of outbound calls:

- ▶ **Web Service.** Calls to web services. Web services are also monitored in the SOA Service views. For more information, see "SOA Services Views" on page 651.
- ▶ **HTTP.** Outbound HTTP calls
- ▶ **Corba.** Calls to remote Corba objects
- ▶ **JDBC.** Calls to databases. By default these outbound calls are not shown. To enable showing these calls, set the **create.database.fragmentArcs** property to **true** in the `<probe_install_dir>/etc/dispatcher.properties` file.
- ▶ **JMS.** Calls between Java servers and message servers.
- ▶ **RMI.** Calls between Java servers.
- ▶ **RFC (SAP).** Calls between SAP servers.
- ▶ **CICS.** Calls within an IBM environment.
- ▶ **MSMQ.** Calls representing sends or puts to an MSMQ queue
- ▶ **.NET Remoting.** Calls to a .NET Remoting component

For details on the user interface, see "Outbound Calls View Description" on page 479.

By default, the Outbound Calls view charts the five calls that have the highest average latency during the selected time range.

An example of the Outbound Calls view is shown below:



Tasks

How to Use the Outbound Calls View

The Outbound Calls view can be used to see performance metrics for outbound requests and find out which server request initiated the call.

If the metrics displayed in the Hosts view raises any concerns, you can navigate to other views to find out more information.

You can also take action to further analyze a performance problem. For example create a snapshot for later analysis.

This task includes the following steps:

- ▶ "Drill Down to Server Requests from Outbound Calls" on page 478

Drill Down to Server Requests from Outbound Calls

When you drill down to the Server Requests view you see details for the originating server request. From the Server Requests view you can view the call profile for a request.

Reference

Outbound Calls User Interface

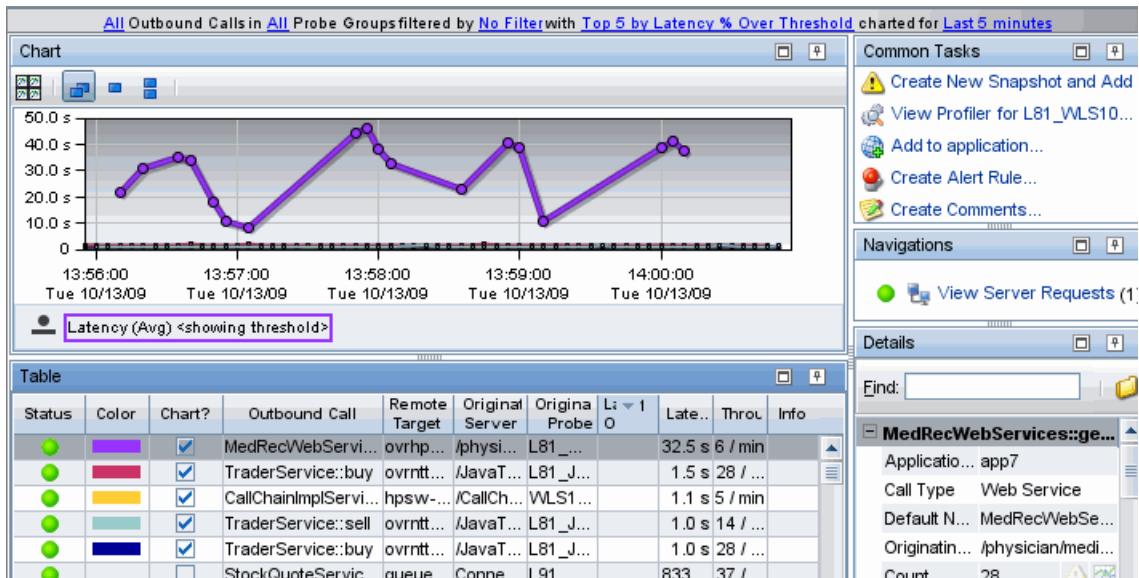
This section includes:


- Outbound Calls View Description on page 479

Outbound Calls View Description

The Outbound Calls view shows server activity that is distributed between multiple servers.

The following is an example of the Outbound Calls view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Outbound Calls.</p> <p>You can also access this view by drilling down from an originating server request.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Outbound Calls View" on page 478 for information on how to analyze performance using the data in this view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average latency for outbound calls, as a trend line. By default, the five calls that have the highest latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular call type or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Outbound Call Details tooltip for the trend line on the graph (or the Outbound Calls column in the table) displays the following detail data:</p> <ul style="list-style-type: none"> ➤ Name. The name of the outbound call. The name represents the entry point to the remote target. ➤ Remote Target. The destination of the outbound call. ➤ Probe Group. The name of the probe group that contains the probe originating the call. ➤ Originating Request. The server request originating the outbound call. ➤ Call Type. Type of outbound call. Can include: ADO, CICS, Corba, HTTP, JDBC, JMS, Remoting (.NET), RMI, SAP ABAP, Web Services. <p>The Outbound Call Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the outbound calls that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the outbound call. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Outbound Call. Name of the outbound call.</p> <p>Remote Target. Destination of the outbound call, can sometimes be a proxy or load balancer.</p> <p>Originating Server Request. The server request originating the outbound call.</p> <p>Originating Probe. The probe originating the call.</p> <p>Latency Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the outbound calls displayed in this view. The drill down navigation based on a pre-defined workflow is View Server Requests.
Details Pane	Shows metrics collected on outbound calls. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

20

SQL Statements View

This chapter describes the SQL Statements view.

This chapter includes:

Concepts

- ▶ SQL Statements by Probe Overview on page 486

Tasks

- ▶ How to Set SQL Trending Thresholds on page 487

Reference

- ▶ SQL Statements User Interface on page 489

Concepts

SQL Statements by Probe Overview

The **SQL Statements** by probe view displays performance metrics for probes collecting data for SQL statements executed by certain methods within your monitored environment. For an aggregate view of SQL statements shown at the probe group level, see the SQL Statements view in the Aggregate Views group.

You can right-click a statement in the table and select **View entire SQL statement**, which brings up the entire SQL statement in a separate dialog box.

Caution: The SQL Statements view should only be used when you are working with a Diagnostics probe from version 6.6 or later.

For details on the user interface, see "SQL Statements (by probe) View Description" on page 489.

The SQL statements in this view are aggregated according to probe group. This means that if the same SQL statement is called from two different probe groups it will be displayed twice. However, if the same statement is called from two different probes within the same probe group, it will be displayed only once.

Note: This behavior is significantly different from most other Diagnostics views. In other views, activity that takes place on different probes is displayed separately.

Tasks

How to Set SQL Trending Thresholds

This task describes how to set SQL trending thresholds.

Trending of the SQL statements begins only after an SQL statement exceeds the predefined latency threshold. After the SQL statement exceeds this threshold, trending continues, even if it falls below the latency threshold. An exception to this is where an application monitored by a Java probe is restarted, then trending stops until the SQL exceeds the threshold again.

The default SQL trending latency threshold is **one second**. You can change this threshold in the property files for the Diagnostics components.

For Java probes you can change the threshold using the **minimum.sql.latency** property in the `<probe_install_dir>\etc\dispatcher.properties` file.

For .NET probes you change the threshold using the **sql.latency.trim** property in either the `<diag_server_install_dir>\etc\server.properties` file or the `<probe_install_dir>\probeagggregator\etc\probeagggregator.properties` file. If you have installed and are using the Probe Aggregator on the .NET Agent system then you set the threshold in the `probeagggregator.properties` file. If the Probe Aggregator is not in use then you set the threshold in the `server.properties` file.

Note: Lowering the SQL trending latency below one second might add significant overhead to the Diagnostics system.

To define the SQL trending latency threshold for Java probes in the `dispatcher.properties` file:

- 1 Open the `<probe_install_dir>\etc\dispatcher.properties` file.

- 2 Adjust the latency threshold by setting the property shown in the following example:

```
minimum.sql.latency = 1s
```

To define the SQL trending latency threshold for .NET probes in the server.properties file:

- 1 Open the **server.properties** file of the Diagnostics Server in Mediator mode to which the probe is connected (<Diag_mediating_server_install_dir>\etc\server.properties file).
- 2 Adjust the latency threshold by setting the property shown in the following example:

```
sql.latency.trim=1000ms
```

Note: The threshold is changed for all .NET probes connected to this Diagnostics Server in Mediator mode (unless using the Probe Aggregator).

To define the SQL trending threshold for .NET probes in the probeagggregator.properties file

- 1 Open the <probe_install_dir>\probeagggregator\etc\probeagggregator.properties file on the .NET Agent system.
- 2 Adjust the latency threshold by setting the property shown in the following example:

```
sql.latency.trim=1000ms
```

Reference

SQL Statements User Interface

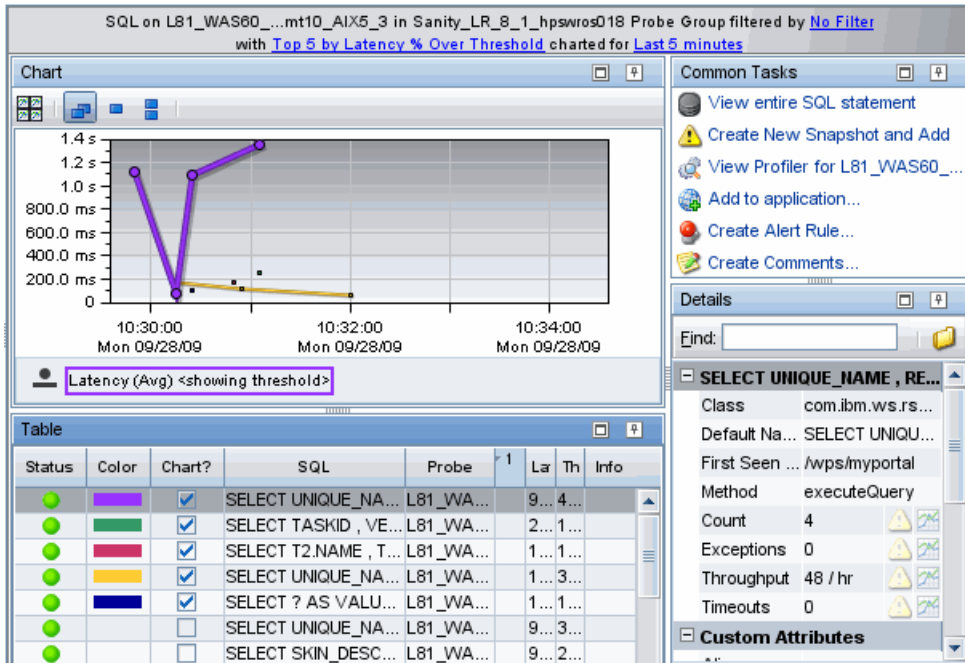
This section includes:


- ▶ [SQL Statements \(by probe\) View Description on page 489](#)

SQL Statements (by probe) View Description

This page shows performance metrics for the monitored SQL statement data for a probe. You must have enabled monitoring of SQL statements for the probe for you to see any data in this view. For details, see "How to Set SQL Trending Thresholds" on page 487.

The following is an example of the SQL Statements view for a probe.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click SQL Statements.</p> <p>You can also access this view directly from the Applications window by selecting the link to SQL Statements under Database Resources in the Application Contents pane.</p> <p>Or you can access the SQL Statements view for a probe by drilling down from a probe (right-click a probe and select View SQL Statements).</p>
<p>Important information</p>	<p>Trending of the SQL statements begins only after an SQL statement exceeds the predefined latency threshold. The default threshold is one second but the value is configurable.</p>
<p>Relevant tasks</p>	<p>For details on how to set the threshold to something other than the one second default, see "How to Set SQL Trending Thresholds" on page 487.</p>

Common Elements

User interface elements are described below:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title bar and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average latency over threshold. By default, the five statements for the selected probe that have the highest latency over the threshold during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>The threshold is shown as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>SQL Details tooltip for the trend line on the graph (or the SQL column in the table) displays the following detail data:</p> <p>SQL. The parsed SQL statement.</p> <p>Probe. The name of the probe.</p> <p>Probe Group. The name of the probe group.</p> <p>Class. The class that contains the SQL statement.</p> <p>First Seen In. The name of the method in which the SQL statement was first identified.</p> <p>The SQL Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the SQL statements that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the SQL statement. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>SQL. The parsed SQL statement.</p> <p>Probe. The name of the probe.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the SQL statements displayed in this view. The drill down navigation based on pre-defined workflow is View entire SQL statement which brings up the entire SQL statement in a separate dialog box.
Details Pane	The Details pane in the SQL Statements view lists the metrics for the selected row of the graph entity table.

21

Layers View

This chapter describes the Layers View.

This chapter includes:

Concepts

- ▶ Layers Overview on page 496
- ▶ Custom Layers on page 496

Tasks

- ▶ How to Use the Layers View on page 497

Reference

- ▶ Layers User Interface on page 498
- ▶ Layers Defined by Default on page 502

Concepts

Layers Overview

The **Layers** view presents a breakdown of the processing across the layers in your application. For information about how to define layers in Diagnostics, see the *HP Diagnostics Installation and Configuration Guide*.

For details on the user interface, see "Layers View Description" on page 498.

By default, Diagnostics charts the latency contribution for the five layers that have the highest percent contribution values during the currently selected time frame.

Custom Layers

HP Diagnostics groups the performance metrics for classes and methods into *layers* and *sub-layers* based upon the instructions provided in the capture points file. The default layers were defined so that the performance metrics for processing in the application that used similar system resources could be reported together. The layers make it easier for you to isolate and identify the areas of the system that may be contributing to performance issues. See "Layers Defined by Default" on page 502.

You may have a class in your application that does not directly implement a Java component, but contains Java functionality that you would like to monitor. Or you may want to monitor a particular class that is of special interest to you in a custom layer. In situations like these, where you want to monitor specific classes in specific ways, Diagnostics enables you to define custom layers.

You must configure the instrumentation of the probe when you want Diagnostics to monitor and report on custom classes or packages.

For instructions on configuring the instrumentation of the probes for custom layers, see the *HP Diagnostics Installation and Configuration Guide*.

Tasks

How to Use the Layers View

Using the information displayed in the Layers view, you can get an immediate understanding of the performance of your application in the layers that are being monitored. If the information displayed in the Layers view raises any concerns, you can drill down to find out more information from a more detailed view of the underlying performance metrics.

This task includes the following steps:

- ▶ "Isolate Problem to a Particular Layer" on page 497
- ▶ "Drill Down to Sub-Layers" on page 497

1 Isolate Problem to a Particular Layer

The Layers view can be used in isolating a performance problem to a particular layer in the application. For example: Business Tier, Database, Web Tier, Web Services layers. This type of triage can result in handing off the problem to the appropriate group for further analysis

2 Drill Down to Sub-Layers

You can drill down from a layer listed in the table by double-clicking or right-clicking the layer's row. Diagnostics displays the Layers view with the sub-layers for the selected layer. If there are no sub-layers defined for the selected layer, double-clicking the row will display a detail layout view, with a note in the table indicating that no data is available.

Reference

Layers User Interface

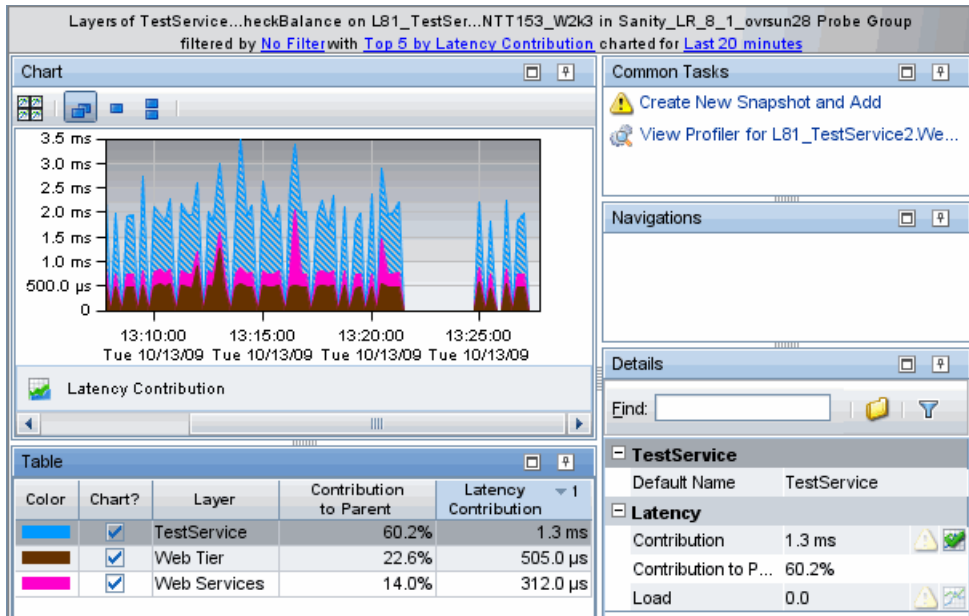
This section includes:

- ▶ Layers View Description on page 498

Layers View Description

The Layers view displays latency contribution for the layers with the highest percent contribution to latency during the time period selected.

The following is an example of the Layers view.



To access	<p>The Layers view cannot be accessed directly from the View bar because layers are typically displayed in the context of another entity such as the application, probe, server request, transaction or web service operation. You can access the Layers view by drilling down from various views such as the Application Metrics view, Transactions view, Server Requests view or SOA Services - Operations views.</p> <p>You can also access the layers view by drilling down from a probe selected in a Status table in one of the summary views.</p>
Important information	You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.
Relevant tasks	See "How to Use the Layers View" on page 497 for information on how to analyze performance using the data in this view.
See also	"Layers Defined by Default" on page 502

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the latency contribution for the layers with the highest percent contribution for the selected probe and probe group. The latency for each layer is shown in a stacked area chart. By default, the five layers with the highest percent contribution are charted. You can use the view filters in the Title bar to filter the data.</p> <p>When Diagnostics is integrated with Load Runner or Performance Center, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the total latency contribution in seconds.</p> <p>Layer Details tooltip for the area graph (or the Layer column in the table) displays the following:</p> <ul style="list-style-type: none"> ▶ Layer. The name of the layer. ▶ Probe. The name of the probe that captured the server request (or transaction). ▶ Probe Group. The name of the probe group to which the probe was assigned. <p>Layer Details tooltip for nodes displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Additional information is displayed in the tooltip depending on which views were accessed before drilling down to the Layers view. For example: Server Request name, Transaction name or Web Service and Operation name.</p>

UI Elements	Description
Table	<p>Lists the layers and the metrics for each layer. Metrics are reported based on the time filter selected.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Layer. Name of the layer.</p> <p>Contribution to Parent. The percentage of the time contributed to the parent's total time.</p> <p>Latency Contribution. The amount of time contributed to the parent's total time.</p>
Common Tasks and Navigations	<p>Common tasks and actions relevant to the hosts displayed in this view.</p> <p>The drill down selection based on a pre-defined workflow is View Layers. The Layers view shows detail data for the sub-layers.</p>
Details pane	<p>The Details pane shows detail metrics for the layers. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Layers Defined by Default

The following tables list the default layers and sub-layers that have been defined for typical Java EE and .NET classes and methods, and indicate the points in the capture points file where the classes and methods are assigned to the layers. This information should help you interpret and customize the points in the capture points file.

Platform specific layers have also been defined in the capture points file. These layers are, for the most part, sub-layers of the top-level parent layers defined in the tables below.

J2EE Layers

Layer	sublayers	Parent Layer
Web Tier	JSP Servlets Struts Session Spring Struts2	
Business Tier	EJB Corba	
Web Services		
EJB	Entity Bean Session Bean Local Home Stateless Session Bean Stateful Session Bean MessageDriven Bean	Business Tier
Directory Service	JNDI	
Database	JDBC	
JDBC	Execute Connection	Database
Messaging	JMS Spring	

Layer	sublayers	Parent Layer
JMS	Producer Listener Consumer	Messaging
Spring	Producer Consumer	Messaging
Hibernate		

.NET Layers

Layer	sublayers	Parent Layer
Web Tier	IIS	
IIS	ExecutionSteps	
Database	ADO	
ADO	Execute Connection Fill Update Cache	Database
Messaging	Sender Receiver	
Web Services	Soap Http WCF	
LWMD		
HTTP Client		
Outbound Calls		

Portal Layers

Diagnostics groups the performance metrics for the classes and method calls associated with processing for portals into Portal Component layers. Each Portal Component layer is broken down to layers for the portal lifecycle methods.

22

Load View

This chapter describes the Load view.

This chapter includes:

Concepts

- ▶ Load Overview on page 508
- ▶ How Load is Calculated in Diagnostics on page 508

Tasks

- ▶ How to Use the Load View on page 510

Reference

- ▶ Load User Interface on page 511

Concepts

Load Overview

The Load view presents a breakdown of the load across the layers, using a stacked area graph. For information about how to define layers in Diagnostics, see the *HP Diagnostics Installation and Configuration Guide*.

For details on the user interface, see "Load View Description" on page 511.

By default, Diagnostics charts the load for the five layers that have the highest load values during the previous five minutes.

How Load is Calculated in Diagnostics

The load in Diagnostics is determined based on a combination of your application's performance characteristics. Load is calculated to provide you with a powerful and concise view of how your application is performing. The characteristics that are used to determine the load for each Diagnostics layer are:

- ▶ The relative ratio of the amount of processing time spent in various layers over time.
- ▶ The relative amount of traffic on the monitored system over time.

The following formula provides a more specific explanation of how load is calculated in Diagnostics:

Load = Average Latency / Point Duration

Where:

- ▶ Average Latency = Total Time / Count
- ▶ Total Time is the sum of the latency of each of the threads that are executing in a layer.

- Count is the number of threads that executed in the layer during the time of interest.
- Point Duration is the length of time over which data has been aggregated to display a level of granularity in the Diagnostics views.

Tasks

How to Use the Load View

Using the information contained in the Load view, you can get an immediate understanding of the performance of your application in the layers that are being monitored. If the information displayed in the Load view raises any concerns, you can drill down to the sub-layers.

This task includes the following steps:

- ▶ "Isolate a Problem to a Particular Layer" on page 510
- ▶ "Drill Down to Sub-Layers" on page 510

1 Isolate a Problem to a Particular Layer

The Load view can be used to isolate a performance problem to a particular tier in an application. For example: Business Tier, Database, Web Tier, Web Services layers. This type of triage can result in handing off the problem to the appropriate group for further analysis.

2 Drill Down to Sub-Layers

You can drill down from a layer listed in the table by double-clicking or right-clicking the layer's row. Diagnostics displays the Load view with the sub-layers for the selected layer. If there are no sub-layers defined for the selected layer, double-clicking the row will display a detail layout view, with a note in the table indicating that no data is available.

Reference

Load User Interface

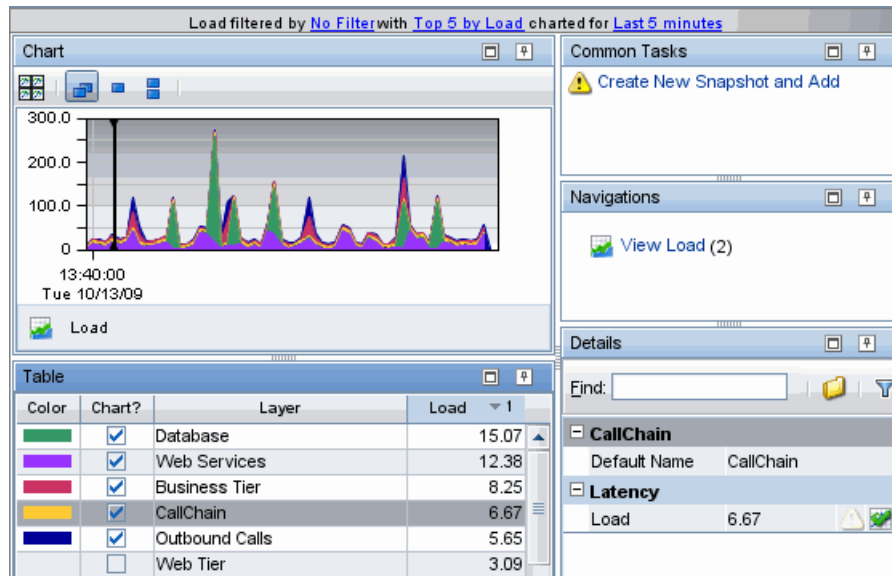
This section includes:


- ▶ Load View Description on page 511

Load View Description

The Load view shows the load calculated for each type of layer.

The following is an example of the Load view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click Entire Enterprise. The Diagnostics views open.</p> <p>In the Standard Views view group, click Load.</p> <p>If you select a single application context instead of Entire Enterprise, the Load view is not available in the Standard Views view group.</p> <p>You can access the Load view by drilling down from various other views where you've selected a probe.</p>
<p>Important information</p>	<p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Load View" on page 510 for information on how to analyze performance using the data in this view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<p><Common elements></p>	<p>For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."</p>
<p><View title and view filters></p>	<p>The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.</p>

Page Content

The following user interface elements are included:

UI Elements	Description
<p>Chart</p>	<p>Displays a graph of the load for each layer where processing has taken place in your application. The breakdown of the load across layers is shown in a stacked area graph. By default, the five layers that have the highest load during the previous five minutes are charted. You can use the view filters in the Title bar to filter the data.</p> <p>The x-axis of the graph shows actual chronological time in hours, minutes, and seconds (hh:mm:ss). The y-axis of the graph shows the scale for the calculated load values.</p> <p>Layer Details tooltip for the area graph (or the Layer column in the table) displays the layer (and sub-layer) name.</p> <p>The information included in the Layer Details tooltip depends on which views were accessed before drilling down to the Load view. For example, if the Load view was accessed by drilling down from a probe you would also see probe and probe group name in the tooltip.</p>
<p>Table</p>	<p>Lists the layer and the load for each layer.</p> <p>The table is customizable. You can specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Layer. Name of the layer.</p> <p>Load. Load value for the layer.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the hosts displayed in this view. The drill down selection based on pre-defined work flow is View Load. The Load view shows detail data for the sub-layers.
Details Pane	The Details pane just shows the load. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

23

Status View

This chapter describes the Status view.

This chapter includes:

Concepts

- ▶ Status Overview on page 516
- ▶ Status Propagation on page 517
- ▶ Status Indicators on page 517

Reference

- ▶ Status User Interface on page 520

Concepts

Status Overview

The Status view is a table that contains the probe groups that you defined when you installed the probes. Within each probe group in the Status view probe group table there are several sub-tables.

- ▶ **Probe Table.** Lists all of the probes in the probe group.
- ▶ **Host Table.** Lists the hosts for each of the probes in the probe group.
- ▶ **Monitors Table.** If you have configured an integration with SiteScope to forward data to Diagnostics you will see the Monitors table, which lists the SiteScope monitors in the probe group.

For details on the user interface, see "Status View Description" on page 520.

The Status view presents a status for the following:

- ▶ Each probe group
- ▶ Each probe in the probe table
- ▶ Each host in the host table
- ▶ Each SiteScope monitor in the monitors table

The Title bar contains the baseline control. You use this control to set the time that Diagnostics uses as the baseline performance level to compare the current performance level against.

The result of the comparison of the baseline performance and the current performance is shown using the metric trend indicators that are displayed following the metrics values in the metric columns of the Status view tables.

Precise information about how the current performance of a given metric compares against its baseline performance is available in the tooltip for the metric.

The following example of a Status view highlights the features and controls of the view.

The screenshot shows a hierarchical status view. At the top, a title bar reads "Status for Last 5 minutes baselined against Last 6 hours". Below this is a tree view on the left showing a hierarchy: Probe Group (Default) -> Probe (EnterprisePortal, Web-Services) -> Host (WLB1-WS) -> Host (t-ic8.lab.performant.com) -> Host (WAS_server1) -> Host (ALMAGRO). The main area displays a table of metrics for these entities. A red box highlights the CPU usage of 65% for the host t-ic8.lab.performant.com. Annotations with lines point to various UI elements: "Probe Group Status" points to the tree view; "Probe Group" points to the "Default" group; "Probe" points to "EnterprisePortal"; "Host" points to "WLB1-WS"; "Status View Title" points to the top title bar; "Baseline Control" points to the "Last 6 hours" link; "Probe Group Headers" points to the "Probe Group" header; "Host Status" points to the "Host" column; "Threshold Alert Indicator" points to the "65%" value; "Trend Indicator" points to the up/down arrows; "Host Table Header" points to the "Host" header; "Probe Table Header" points to the "Probe" header.

Entity	Metric	Value	Trend
Host: t-ic8.lab.performant.com	CPU	65%	↕
Host: t-ic8.lab.performant.com	Disk Bytes/S	13.6 KB	↕
Host: t-ic8.lab.performant.com	Network Bytes/S	926.9 KB	↕
Host: t-ic8.lab.performant.com	Page In/S	0	↕
Host: t-ic8.lab.performant.com	Memory	82%	↕
Host: WAS_server1	VM Heap Used	83.1 MB	↕
Host: WAS_server1	CPU	2%	↕
Host: WAS_server1	Disk Bytes/S	17.2 KB	↕
Host: WAS_server1	Network Bytes/S	21.7 KB	↕
Host: WAS_server1	Page In/S	2	↕
Host: WAS_server1	Memory	44%	↕
Host: ALMAGRO	CPU	12%	↕
Host: ALMAGRO	Disk Bytes/S	781.8 KB	↕
Host: ALMAGRO	Network Bytes/S	25.3 KB	↕
Host: ALMAGRO	Page In/S	165	↕
Host: ALMAGRO	Memory	97%	↕

Status Propagation

It is important to note the way in which status propagates through the Diagnostics application. For example, a probe with a good status will become critical if any of its server requests become critical. Similarly, a probe group will become critical if any of its probes become critical. However, Hosts are separate entities, and will only become critical if their host metrics exceed defined thresholds.

Status Indicators

In many of the Diagnostics views you will see Status Indicators in the first column of the graph entity table. The Status Indicator is described below:



- **Status.** This indicator shows how the listed entity or any of its child entities is performing relative to the thresholds that you set for their metrics.

For example, a probe can have a critical status even when none of the probe's own metrics have exceeded their thresholds. The probe's status will become critical when any of the server requests for that probe become critical.

The color of the indicator stands for the severity of the threshold violation. If you hold the mouse pointer over the status indicator in a row of the table, Diagnostics displays a tooltip with a description of the current status:

Status Indicator Color	Description
Green	Good – The entity is performing within defined thresholds.
Yellow	Warning – The component is occasionally but not consistently exceeding defined thresholds. If the metric that is exceeding the defined threshold is displayed in the entity table, the cell for the metric is also outlined in yellow.
Red	Critical – The component is consistently exceeding defined thresholds. If the metric that is exceeding the defined threshold is displayed in the entity table, the cell for the metric is also outlined in red.
Grey	No status information available. Either no recent data was received for the metric or no threshold was set.

If the metrics that are exceeding the threshold are displayed in the row of the graph entity table, they are highlighted with a threshold violation indicator.

The following describes, in general, how a status of RED or YELLOW is determined:

- ▶ Diagnostics compares the average value for the metric in the last 5 minutes to the threshold. If the threshold is crossed, the status is RED.

- ▶ If not, Diagnostics counts the total number of points (5-second averages from the probe) that violate the threshold. If the count is greater than three but the average is still under the threshold (current status is not RED), the status is YELLOW; otherwise the status stays at GREEN.
- ▶ If none of the above, the status is GREEN.

Status indicators can progress in severity from yellow and then to red as the metric's performance deteriorates. However, the indicators do not regress in the same way. Once a red indicator has been issued, it continues to be displayed as red until the metric value returns to normal levels.

For information on setting thresholds, see "About Metric Thresholds" on page 158.

Reference

Status User Interface

This section includes:

- ▶ [Status View Description on page 520](#)

Status View Description


The **Status** view displays a table that gives you a quick view of the status and performance of all the probes and hosts in your monitored environment.

You can drill down into the details for the probe groups, probes and hosts that are behind the status that you see in the view. And you can take actions that are relevant to the entity selected.

Right-clicking a row in the Status table displays a pop-up menu of suggested actions you can take to further analyze a performance problem and diagnose the cause. The available actions and drill downs vary according to the type of probe group, probe or host.

The following is an example of the Status view.

Status for Last 5 minutes baselined against Last 6 hours							
St. ▾ 1	Probe Group			Diagnostics Server	Info		
+	Sanity_CallChain_WebService_ovruxd10			server-ovruxd10.rose.hp.com			
+	Sanity_LR_8_1_ovrntt154			server-OVRNTT154			
-	Sanity_LR_9_1_hpswros018			server-HPSWROS018			
St. ▾ 1	Probe	VM Heap Used	Latency	Throughput	Timeouts	Exceptions	Info
⊗	L91_JavaTrader2.WebClien...	194.0...	4 m, ... ↑	132 / hr ↕	11	0	
⊗	L91_JavaTrader2.WebClien...	277.8...	898.1 ... □	179 / ... ↑	0	0	
⚠	L91_CallChain2_0.NET_OV...	184.6...	1.0 s ↕	12 / hr ↓	0	0	
⚠	L91_TestService2.WebClie...	186.1...					
⚠	L91_TestService2.WebSer...	190.2...					
⚠	L91_TestService2.WebSer...	187.4...					
●	L91_TestService2.WebSer...	274.0...	3.6 ms □	18 / min ↑	0	0	
●	L91_CallChain2_0.NET_OV...	277.1...	603.9 ... □	79 / min ↑	0	0	
●	L91_TestService2.WebSer...	277.1...	41.5 ms ↕	18 / min ↑	0	0	
●	L91_MSPetShop_1.1_NET_...	43.6 ...	1.6 ms ↕	12 / min □	0	0	
●	L91_WAS60_Plants_oxycyco...	136.6...	7.8 ms □	277 / ... ↑	0	0	
●	L91_TestService2.WebClie...	275.2...	16.5 ms □	72 / min ↑	0	0	
St. ▾ 1	Host	CPU	Disk Bytes/S	Network Bytes/S	Page In/S	Memory	Info
●	ovresx1-vm10	8% ↕	9.1 KB ↕	107.4... ↕	0	32% □	
●	oxycontin.rose.hp.com	2% □	4.8 KB ↕	495.9... ↕	51	96% □	
●	ovresx2-vm3.ovrtest.adapp...	7% ↕	10.3 KB ↕	56.6 KB ↕	0	32% □	

<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the Standard Views view group, click Status.</p>
<p>Important information</p>	<p>Changing the global time range has no affect on the Status view—it is always based on the last five minutes.</p> <p>You can use the standard Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the Server Summary view. For more information about the ways you can control how performance metrics are presented in this view, see Chapter 3, "Common Controls in the Diagnostics UI."</p> <p>The Probe Group Summary view and Probe Summary view are available navigations by right-click menu. These views are not available from the Standard Views, only by navigating from a status or summary view. See "Server Summary User Interface" on page 289 for details on these views.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Status Propagation" on page 517 ➤ "Status Indicators" on page 517

Common Elements

The following user interface elements are included:

UI Elements (A-Z)	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."

Page Content

The following user interface elements are included:

UI Elements	Description
<Title>	<p>The title of the status table contains the baseline control. You use this control to set the time that Diagnostics uses as the baseline performance level with which to compare the current performance level. The result of the comparison of the baseline performance to the current performance is shown by using the metric trend indicators (arrows that are displayed following the metric values in the status tables).</p>
Status	<p>By default, the first column in the Status table presents the status for each probe group, probe, and host. The status reflects how the performance of the items compares with their performance thresholds.</p>
<Probe group table>	<p>Lists the probe groups that you defined when you installed the probes.</p> <p>Probe Group Tooltip displays the probe group name.</p> <p>Within each probe group, there are several sub-tables for Probes, Hosts and Monitors. You can expand and collapse these sub-tables.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p>

UI Elements	Description
<Probe table>	<p>Lists the probes in the probe group along with the status. The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>Probe Details tooltip for the probe column displays:</p> <ul style="list-style-type: none"> ➤ Probe. The name of the probe. ➤ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ➤ Host. The host on which the probe is running. ➤ Probe Type. Java, .NET, Oracle, SAP, MQ, SQL Server. <p>By default, the following metrics are displayed for each probe (the displayed columns are customizable):</p> <p>VM Heap Used. The average amount of heap used by the applications being monitored by the probe.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>The result of the comparison of the baseline performance to the current performance is shown in the metric trend indicator arrows that are displayed following the metric values. A tooltip for each indicator gives a description of the current status. See "Status Indicators" on page 517 for details on status indicators.</p> <p>When you double-click a row in the probe table, the pre-defined workflow displays the Probes view for the selected probe.</p>

UI Elements	Description
<Hosts table>	<p>Host Table. Lists the hosts for each of the probes in the probe group.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>The Host Details tooltip for the host column displays the fully qualified host name.</p> <p>By default, the following performance information is displayed for each host:</p> <p>CPU. Average CPU utilization (system metric)</p> <p>Disk Bytes/Second. The average number of bytes read and written per second.</p> <p>Network Bytes/Second. The average number of bytes sent and received per second.</p> <p>PageIn/Second. The average number of virtual memory pages swapped in per second.</p> <p>Memory. The average amount of memory in use.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>The result of the comparison of the baseline performance to the current performance is shown in the metric trend indicator arrows that are displayed following the metric values. A tooltip for each indicator gives a description of the current status. See "Status Indicators" on page 517 for details on status indicators.</p> <p>When you double-click a row in the host table, the pre-defined workflow displays the Hosts view for the selected host.</p>

UI Elements	Description
<Monitors table>	<p>Monitors Table. If you have configured an integration with SiteScope to forward data to Diagnostics you will see the Monitors table, which lists the SiteScope monitors in the probe group. By default, the following performance information is displayed for each monitor:</p> <p>Type. Type of monitor.</p> <p>Collector. Collector responsible for this monitor (for example SiteScope).</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>When you double-click a row in the monitors tables, the pre-defined workflow displays the Monitors view for the selected monitor.</p>

Part IV

Understanding Diagnostics Specialized Views

24

Aggregate Views

This chapter describes the Aggregate Views view group.

This chapter includes:

Concepts

- Aggregate Views Overview on page 530
- Aggregate Request Overview on page 530
- Dependent Services Overview on page 531
- SQL Statements Overview on page 532
- Trended Methods Overview on page 532

Tasks

- How to Set SQL Trending Thresholds on page 534
- How to Configure Instrumentation Points for Method Trending on page 536

Reference

- Aggregate View Group User Interface on page 538

Concepts

Aggregate Views Overview

The **Aggregate Views** group contains a number of views showing you performance data that is aggregated by probe group. Data from probes within the same probe group is aggregated in these views. These views are useful in cluster environments where probes are installed on each cluster node and all cluster nodes represent a logical application.

The Aggregate Views group displays:

- ▶ **Aggregate Requests** - shows performance metrics for server requests aggregated across all probes in each probe group.
- ▶ **Dependent Services** - Shows performance metrics for dependent services (locations and types). This can be thought of as aggregate outbound calls.
- ▶ **SQL Statements** - If enabled, shows performance metrics for monitored SQL statements in your enterprise. SQL statements are aggregated by probe group.
- ▶ **Trended Methods** - Shows performance data by probe group for methods configured to be individually trended.

Aggregate Request Overview

When multiple probes monitor the same application, the **Aggregate Requests** view shows server requests aggregated to the probe group level.

You can assign probes that monitor the same application to a single probe group. For example, you could use the Aggregate Requests view to monitor the virtual machines that make up a distributed application that is deployed in a cluster. If you assign these probes to a “cluster” probe group, you can monitor this probe group’s server requests and dependent services as a group with the aggregation of status, latency, and other key metrics. When a problem at the probe group level is detected, you can drill down into the server requests and dependent service calls (or outbound calls).

For details on the user interface, see "Aggregate Requests View Description" on page 538.

By default, the Aggregate Requests graph charts the average latency for the five aggregated requests that have the highest latency during the previous hour.

Dependent Services Overview

The **Dependent Services** view contains performance metrics for dependent services (locations and types) aggregated to the probe group level. This can be thought of as aggregate outbound calls.

The Dependent Services view displays latency broken down by services that are utilized by a service request. For example, if a number of server requests use 'JDBC' and 'WebServices', by clicking on one of these services you can see how the time was used in each of the server requests that used this service. This is a reverse breakdown of time, normally you break a server request down to see where the time is spent. In this case you know some time was spent in certain services and you are interested in knowing which server requests used this service and are dependent on it.

For details on the user interface, see "Dependent Services View Description" on page 544.

If the metrics displayed for a service type or location raise any concerns, you can drill down to find more information from a view that depicts performance metrics at a lower level. For example, from the Dependent Services view, you can drill down to the Service Calls view showing individual probe group outbound calls.

SQL Statements Overview

The **SQL Statements** view displays performance metrics aggregated to the probe group level for SQL statements executed by certain methods within your monitored environment. You can right-click a statement in the table and select **View entire SQL statement**, which brings up the entire SQL statement in a separate dialog box.

Caution: The SQL Statements view should only be used when you are working with a Diagnostics probe from version 6.6 or later.

For details on the user interface, see "SQL Statements View Description" on page 554.

The SQL statements in this view are aggregated according to probe group. This means that if the same SQL statement is called from two different probe groups it will be displayed twice. However, if the same statement is called from two different probes within the same probe group, it will be displayed only once.

Since SQL statements are aggregated for a probe group this allows you to filter the data roughly by application.

Note: There is also an SQL Statements by probe view accessed from the Standard views.

Trended Methods Overview

The **Trended Methods** view displays latency aggregated to the probe group level for methods you've configured for trending. For details on configuring instrumentation for method trending, see "How to Configure Instrumentation Points for Method Trending" on page 536.

For details on the user interface, see "Trended Methods View Description" on page 559.

The information displayed on the Trended Methods view differs, depending on how you access the view.

- ▶ When you access the Trended Methods view directly from the view bar, each trended method represents the aggregation of all the times this particular method was executed on different servers within a particular probe group. The actual probe where the method was executed is not identified.
- ▶ When you access the Trended Methods view by drilling down from a probe, each trended method represents the aggregation of all the times this method was executed on servers within a particular probe.

Tasks

How to Set SQL Trending Thresholds

This task describes how to set SQL trending thresholds.

Trending of the SQL statements begins only after an SQL statement exceeds the predefined latency threshold. After the SQL statement exceeds this threshold, trending continues, even if it falls below the latency threshold. An exception to this is where an application monitored by a Java probe is restarted, then trending stops until the SQL exceeds the threshold again.

The default SQL trending latency threshold is **one second**. You can change this threshold in the property files for the Diagnostics components.

For Java probes you can change the threshold using the **minimum.sql.latency** property in the `<probe_install_dir>\etc\dispatcher.properties` file.

For .NET probes you change the threshold using the **sql.latency.trim** property in either the `<diag_server_install_dir>\etc\server.properties` file or the `<probe_install_dir>\probeagggregator\etc\probeagggregator.properties` file. If you have installed and are using the Probe Aggregator on the .NET Agent system then you set the threshold in the `probeagggregator.properties` file. If the Probe Aggregator is not in use then you set the threshold in the `server.properties` file.

Note: Lowering the SQL trending latency below one second might add significant overhead to the Diagnostics system.

To define the SQL trending latency threshold for Java probes in the `dispatcher.properties` file:

- 1 Open the `<probe_install_dir>\etc\dispatcher.properties` file.

- 2 Adjust the latency threshold by setting the property shown in the following example:

```
minimum.sql.latency = 1s
```

To define the SQL trending latency threshold for .NET probes in the server.properties file:

- 1 Open the **server.properties** file of the Diagnostics Server in Mediator mode to which the probe is connected (**<Diag_mediating_server_install_dir>\etc\server.properties file**).
- 2 Adjust the latency threshold by setting the property shown in the following example:

```
sql.latency.trim=1000ms
```

Note: The threshold is changed for all .NET probes connected to this Diagnostics Server in Mediator mode (unless using the Probe Aggregator).

To define the SQL trending threshold for .NET probes in the probeagggregator.properties file

- 1 Open the **<probe_install_dir>\probeagggregator\etc\probeagggregator.properties** file on the .NET Agent system.
- 2 Adjust the latency threshold by setting the property shown in the following example:

```
sql.latency.trim=1000ms
```

How to Configure Instrumentation Points for Method Trending

To see data in the Trended Methods view, configure the instrumentation point for the method so that the latency for this method gets captured. This data can then be displayed as a trend line in the Trended Methods view.

Caution: By default, this view is empty. For a method to be displayed in this view, it needs to be configured.

To configure a method to be displayed in the Trended Methods view:

- 1 Open the relevant capture points file.
 - ▶ By default, the Java Probe capture points file is located at `<probe_install_dir>\etc\auto_detect.points`.
 - ▶ For the .NET Probe, the capture points files are located in the `<.NET probe_install_dir>\etc\` directory.
- 2 In the capture points file, locate or add the point that includes the method you want to display in the Trended Methods view.
- 3 Add the following argument to the point definition:

```
layertype = trended_method
```

- 4 Restart the application server on which your probe is running.

Caution:

- ▶ Configuring methods to be displayed in the Trended Methods view might add significant overhead to the Diagnostics system.
 - ▶ If you configured the probe to capture arguments for a point that is also configured for method trending, each unique argument will result in a unique trended method. This can have a significant impact on memory overhead.
 - ▶ Trended methods are not latency trimmed. This means that any frequently-executing method with low latency could cause a noticeable performance impact on the application.
-

Reference

Aggregate View Group User Interface

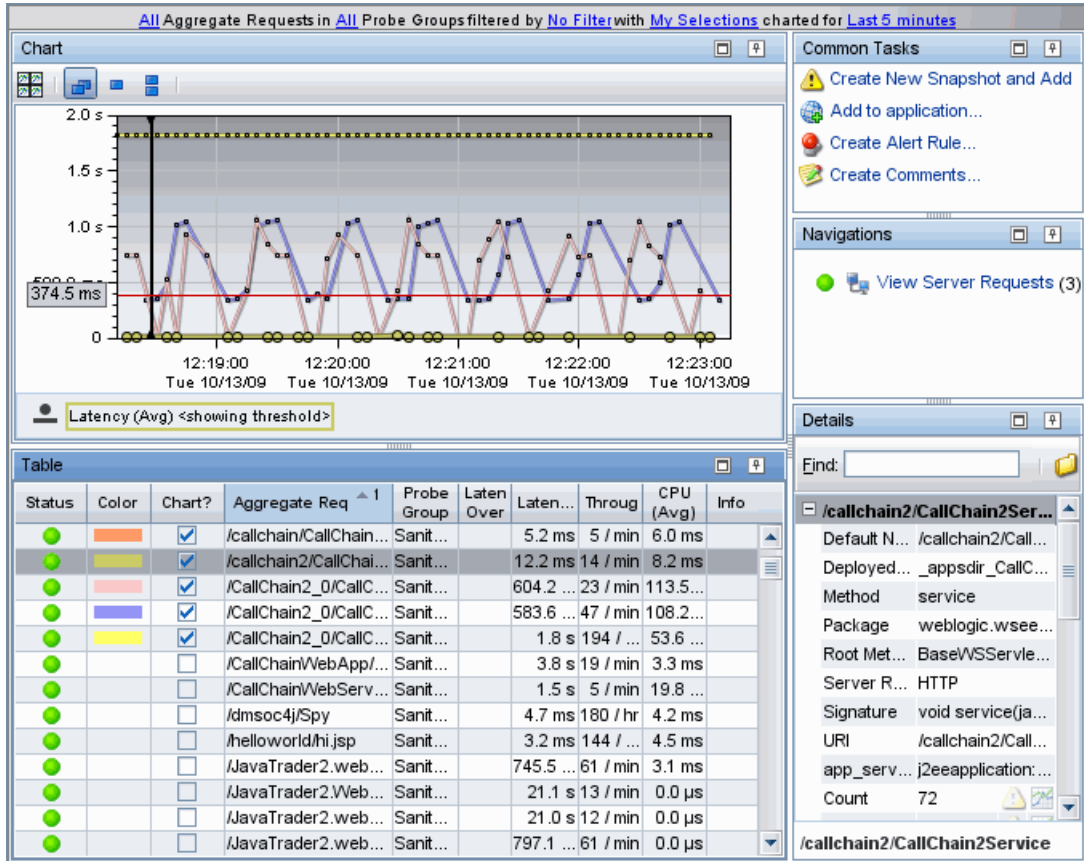
This section includes:

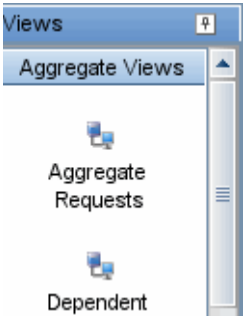
- ▶ [Aggregate Requests View Description on page 538](#)
- ▶ [Dependent Services View Description on page 544](#)
- ▶ [Service Calls View Description on page 549](#)
- ▶ [SQL Statements View Description on page 554](#)
- ▶ [Trended Methods View Description on page 559](#)

Aggregate Requests View Description

The Aggregates Requests view displays status and performance data for server requests aggregated at the probe group level. This view allows you to monitor a probe group and if a problem is found, you can drill down to the individual server requests within the aggregated item.

The following is an example of the Aggregate Requests view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). Then open the Aggregate Views view group and click Aggregate Requests.</p> <p>You can also access this view directly from the Applications window by selecting the Aggregate Requests link under Server Requests in the Application Contents pane.</p> <p>You can drill down to this view from a probe group.</p>
<p>Important information</p>	<p>You can use the Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>

Common Elements

The following user interface elements are included (unlabeled GUI elements are shown in angle brackets):

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View Title and View Filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average latency for each aggregate request, as a trend line. By default, the five requests with the highest average percent latency over threshold during the previous five minutes are charted. You can use the view filters in the Title bar to filter the data displayed.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Server Request Details tooltip for the trend line in the graph (or the Aggregate Request column in the table) typically displays the following detail data:</p> <ul style="list-style-type: none"> ▶ Server Request. The name of the selected server request. ▶ Probe. The name of the probe that captured the server request. This name should be the same name that is displayed in the Probe column of the Server Requests table. ▶ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ▶ Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself. ▶ Type. The Server Request type. Examples of server request types include: ADO, CICS, Corba, HTTP, JDBC, JMS, MSMQ, RMI, Remoting (.NET), SAP ABAP types, Web Services. ▶ Package. The name of the package that contains the class from which the method was called. ▶ Deployed Into. Tells where the web service was deployed into (usually an EAR file). ▶ Method Signature. The signature of the root method. <p>Other types of details tooltips may be displayed depending on the type of server request (for example, SAP ABAP Server Request Details or Web Service Details).</p>

UI Elements	Description
Chart	<p>In addition, you may see server request types named Pseudo and RootRename. Pseudo means the server request was not really a server request defined as something that 'hits' your server, like http. If you instrumented just a regular method, then the server request will show up as type pseudo.</p> <p>RootRename is also used for things that are not really server requests. Instead of pseudo the RootRename is based on a points file specification. These server requests will have a name that start with Background. The suffix is the value of the rootRenameTo property specified in the point or the name of the first sub-layer if this property is not specified. To mark a point as a root rename, the detail 'when-root-rename' must be specified in the point. Refer to the <i>HP Diagnostics Installation and Configuration Guide</i> Custom Instrumentation sections for details.</p> <p>The Server Request Details tooltip for the nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

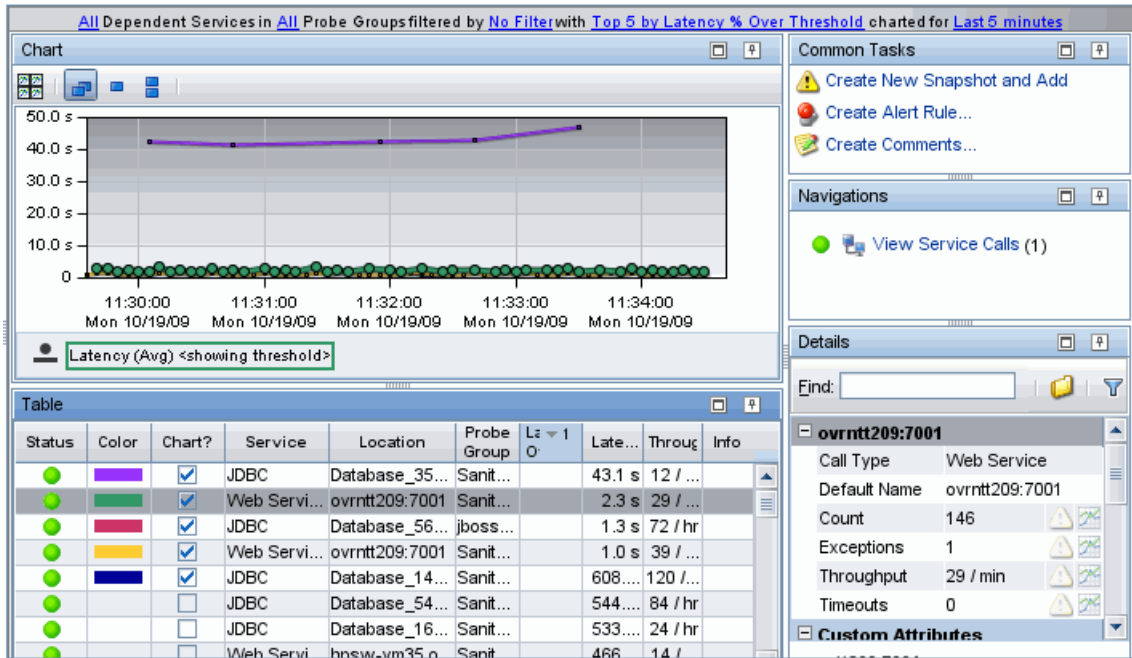
UI Elements	Description
Table	<p>Lists the aggregate requests that pertain to the context selected, either Entire Enterprise or a specific application. The metrics reported in the table are aggregated based on the time period select in the view filters in the title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed</p> <p>Status. By default the first column shows the status for the aggregate request. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Aggregate Request. Aggregated server request name.</p> <p>Probe Group. Probe group name.</p> <p>Latency Percent Over Threshold. The percentage over the threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU. Average CPU utilization (system metric)</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

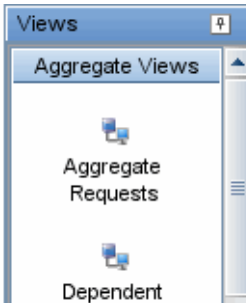
UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the aggregate requests displayed in this view. The drill down selection based on a pre-defined workflow is View Server Requests.
Details	The Details pane shows detail metrics. for the requests. The metrics vary depending on the type of request. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

Dependent Services View Description

The Dependent Services view displays latency and other performance metrics for the services used by a service request.

The following is an example of the Dependent Services view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise).</p> <p>Then open the Aggregate Views view group and click Dependent Services.</p>
<p>Important information</p>	<p>You can use the Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View Title and View Filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
<p>Chart</p>	<p>Displays a graph of the average latency for each dependent service, as a trend line. By default, the five services with the highest average latency during the previous five minutes are charted. You can use the view filters in the Title bar to filter the data for an individual service or an individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Dependent Service Details tooltip for the trend line in the graph (or the Service column in the table) typically displays the following detail data:</p> <ul style="list-style-type: none"> ▶ Location. The dependent service location. ▶ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ▶ Call Type. The type of outbound call. ▶ Call Name. The method call into the dependent service. <p>Dependent Service Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

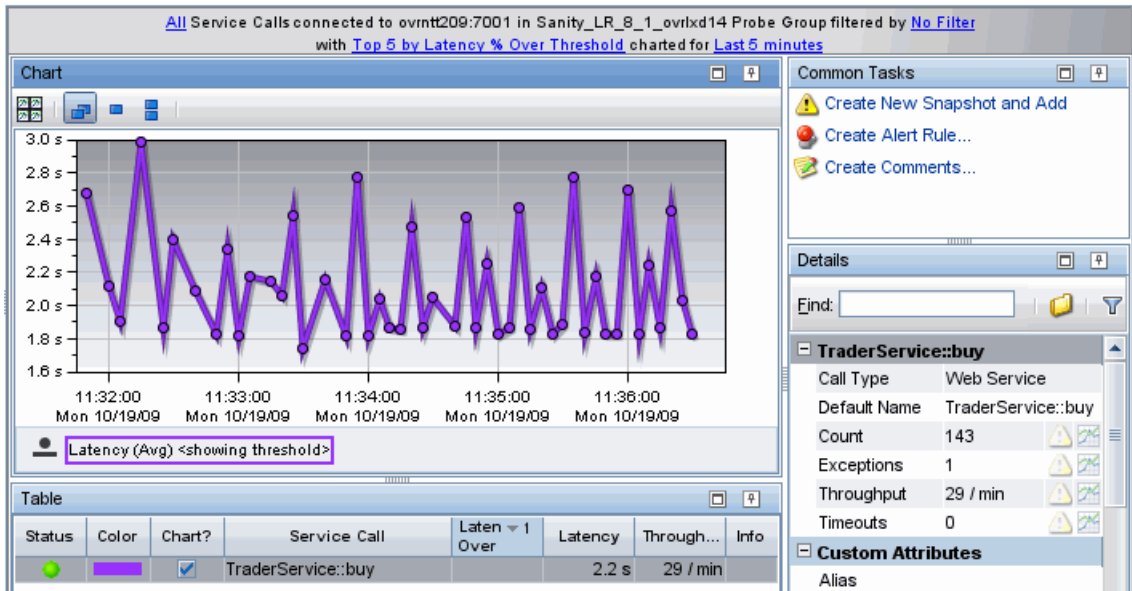
UI Elements	Description
Table	<p>Lists the services (service type and location) related to the context shown in the breadcrumb. The metrics for each service are aggregated and reported, based on the time period specified in the view filter.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the service or location. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Service. Dependent service name.</p> <p>Location. Dependent service location.</p> <p>Probe Group. Probe group name.</p> <p>Latency Percent Over Threshold. The percentage over the threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the dependent services displayed in this view. The drill down selection based on pre-defined workflow is View Service Calls. The Service Calls view shows the calls executed for a selected dependent service.
Details Pane	The Details pane shows detail metrics for the services. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

Service Calls View Description

The Service Calls view shows the calls that were executed for a selected dependent service during the specified time range.

The following is an example of the <name> report.



To access	Drill down from the Dependent Services view to the Service Calls view.
Important information	You can use the Diagnostics view controls to adjust the amount and type of data that Diagnostics displays in the view.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common report elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View Title and View Filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

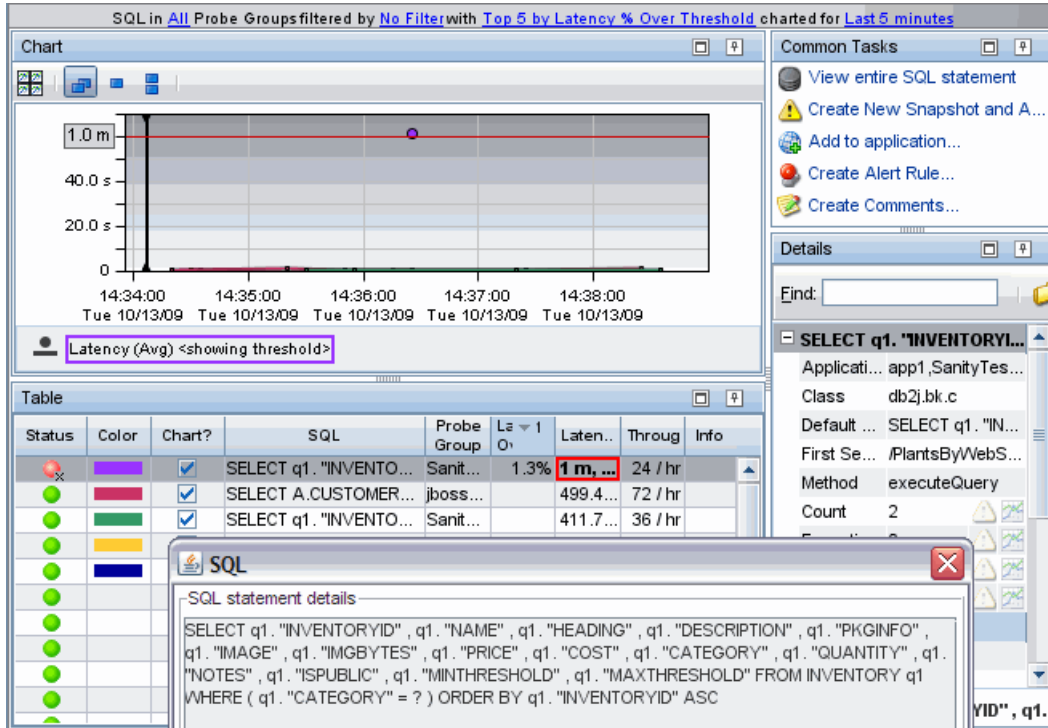
UI Elements	Description
Chart	<p>Displays a graph with trend lines of the average latency for service calls made to the dependent service you selected. By default, the five service calls for this dependent service with the highest latency percent over the threshold the selected time range are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the latency in seconds or milliseconds. Thresholds are displayed as a red horizontal line.</p> <p>Service Call Details tooltip for the trend line in the graph (or the Service Call column in the table) typically displays the following detail data:</p> <ul style="list-style-type: none"> ▶ Service Call. Service call name. ▶ Location. The dependent service location. ▶ Probe Group. The name of the probe group to which the probe was assigned when it was installed. ▶ Call Type. The type of outbound call. ▶ Call Name. The method call into the dependent service. <p>Service Call Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

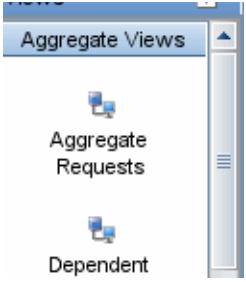
UI Elements	Description
<p>Table</p>	<p>Lists the service calls that are related to the dependent service you selected. (This context is shown in the breadcrumb at the top of the graph.)</p> <p>The metrics for each service call that is reported in the table are aggregated and reported based on the time period specified in the Time Range view filter.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed</p> <p>Status. By default the first column shows the status for the service call. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Service Call. Service call name.</p> <p>Latency Percent Over Threshold. The percentage over the threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
<p>Common Tasks and Navigations</p>	<p>Common tasks relevant to the service calls displayed in this view, for example, create a snapshot.</p>
<p>Details Pane</p>	<p>The Details pane shows detail metrics for the services. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

SQL Statements View Description

The **SQL Statements** view displays performance metrics for SQL statements executed by certain methods within your monitored environment.

The following is an example of the SQL Statements view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise).</p> <p>Then open the Aggregate Views view group and click SQL Statements.</p> <p>You can also access this view directly from the Applications window by selecting the SQL Statements link under Database Resources in the Application Contents pane.</p>
<p>Important information</p>	<p>The SQL Statements view should only be used when you are working with a Diagnostics probe from version 6.6 or later.</p> <p>Trending of the SQL statements begins only after an SQL statement exceeds the predefined latency threshold. The default threshold is one second but the value is configurable.</p>
<p>Relevant tasks</p>	<p>See "How to Set SQL Trending Thresholds" on page 534 for information on how to set the threshold to something other than the one second default.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
View title bar and view filters	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of the average latency over threshold. By default, the five statements that have the highest latency over the threshold during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>The threshold is shown as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>SQL Details tooltip for the trend line on the graph (or the SQL column in the table) displays the following detail data:</p> <p>SQL. The parsed SQL statement.</p> <p>Probe. The name of the probe.</p> <p>Probe Group. The name of the probe group.</p> <p>Class. The class that contains the SQL statement.</p> <p>First Seen In. The name of the method in which the SQL statement was first identified.</p> <p>The SQL Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

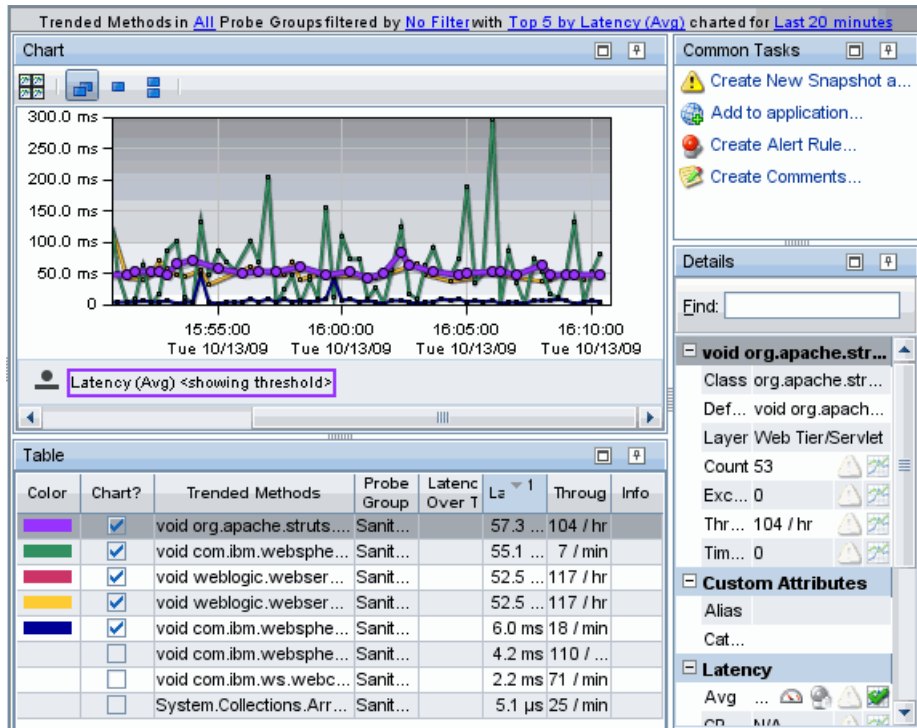
UI Elements	Description
Table	<p>Lists the SQL statements that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the SQL statement. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>SQL. The parsed SQL statement.</p> <p>Probe. The name of the probe.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

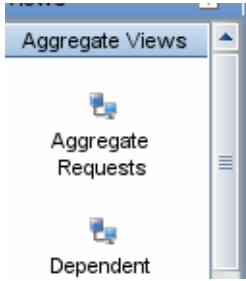
UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the SQL statements displayed in this view.</p> <p>The drill down navigation based on pre-defined workflow is to right-click and select View entire SQL statement, which brings up the entire SQL statement in a separate dialog box.</p>
Details Pane	<p>The Details pane in the SQL Statements view lists the metrics for the selected row of the graph entity table. This includes the First Seen From metric, which displays the name of the method in which the SQL statement was first identified.</p>

Trended Methods View Description

The **Trended Methods** view displays latency for methods you've configured for trending.

The following is an example of the Trended Methods view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise).</p> <p>Then open the Aggregate Views view group and click Trended Methods.</p> <p>You can also access this view directly from the Applications window by selecting the Aggregate Trended Methods link in the Application Contents pane.</p>
<p>Important information</p>	<p>By default, this view is empty. To see data in the Trended Methods view, configure the instrumentation point for the method so that the latency for this method gets captured.</p>
<p>Relevant tasks</p>	<p>See "How to Configure Instrumentation Points for Method Trending" on page 536.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
<p>Chart</p>	<p>Displays a graph of average latency, as a trend line. By default, the five trended methods that have the highest average latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Method Details tooltip for the trend line on the graph (or the Trended Methods column in the table) displays the following detail data:</p> <p>Method Name. The name of the trended method.</p> <p>Probe Group. The name of the probe group containing the probe.</p> <p>Class. The class associated with the executing method.</p> <p>Layer. The layer in which the method executed.</p> <p>The information displayed in the tooltip differs, depending on how you accessed the Trended Methods view. When accessed directly from the view bar the tooltip identifies the probe group, but not the probe, in which the method was executed. When accessed by drilling down from a probe the tooltip identifies both the probe and probe group in which the method was executed.</p> <p>The Method Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the trended methods that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Trended Methods. The name of the method.</p> <p>Probe Group. The name of the probe group.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	Common tasks and actions relevant to the trended methods displayed in this view.
Details Pane	Shows metrics collected on trended methods. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

25

System Views

This chapter describes the specialized views for system administrators.

This chapter includes:

Concepts

- ▶ System Views Overview on page 564

Reference

- ▶ System View Group User Interface on page 565

Concepts

System Views Overview

In large scale Diagnostics deployments you can use the System Views instead of the System Health Monitor. The specialized System Views allow you to quickly locate systems or groups of system based on various system attributes. Allowing you to more easily monitor system health and identify when systems are nearing capacity.

To access the System Views you must access the Diagnostics UI as the *Mercury System* customer.

The **System Views** view group displays information for system administrators in the following views:

- ▶ **System Capacity.**
- ▶ **System Health.**

For details on the user interface, see "System View Group User Interface" on page 565.

Reference

System View Group User Interface

This section includes:

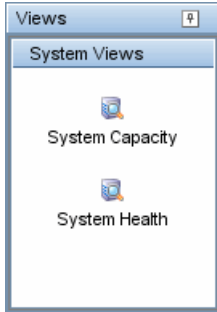
- ▶ System Capacity View Description on page 565
- ▶ System Health View Description on page 567

System Capacity View Description

The System Capacity view allows you to understand how each mediator is being used. The number of probes assigned to each mediator is given. So this view can allow you to easily check if you are nearing capacity on a server.

The following is an example of the System Capacity view.

System Capacity				
Name	Host	Port	Probe Gro... Count	Probe Co...
server-ovrntt100	ovrntt100.ovrtest.adapps.hp.com	2612	3	4
server-ovruxt32.rose.hp.com	ovruxt32.rose.hp.com	2612	2	2
server-ovrlxd14.rose.hp.com	opt.rose.hp.com	2612	2	11
server-ovresx1-vm5	ovresx1-vm5.ovrtest.adapps.hp.com	2612	2	2
server-OVRNTT154	OVRNTT154.ovrtest.adapps.hp.com	2612	3	9
CommandingServer	OVRNTT150.ovrtest.adapps.hp.com	2006	2	2
server-ovrsun28.rose.hp.com	ovrsun28.rose.hp.com	2612	1	9
server-ovruxd10.rose.hp.com	ovruxd10.rose.hp.com	2612	1	1
server-HPSWROS018	HPSWROS018.ovrtest.adapps.hp.com	2612	3	12

<p>To access</p> 	<p>To access the System Views you must access the Diagnostics UI as the Mercury System customer.</p> <p>Access the Mercury System customer in Diagnostics: <a href="http://<Diagnostics_Commanding_Server_Name>:2006/query/">http://<Diagnostics_Commanding_Server_Name>:2006/query/</p> <p>For the Mercury System customer, select the link to Open Diagnostics.</p> <p>Log in to Diagnostics and on the Applications window select Entire Enterprise and select any link to open the Diagnostics Views.</p> <p>In the View bar you'll see the System Views view group. Select the System Health view or System Capacity view.</p>
<p>Important information</p>	<p>This view group is only available for the Mercury System customer.</p>
<p>See also</p>	<p>See "Using the System Health Monitor" in the <i>HP Diagnostics Installation and Configuration Guide</i>.</p>

The following user interface elements are included:

UI Elements	Description
Table	<p>Provides information for managing capacity in your Diagnostics environment. Shows the number of probe groups and probes that are assigned to each Diagnostics mediator. This view is useful for managing a large Diagnostics deployment.</p> <p>Under each server there is a sub-table for Probe Groups. You can expand and collapse sub-table.</p> <p>The table is highly customizable. Right-click any column header to select Customize Columns and specify which metrics or system attributes you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Name. Diagnostics server name.</p> <p>Host. The system host where the Diagnostics component is installed.</p> <p>Port. The port the Diagnostics component communicates on.</p> <p>Probe Group Count. The number of probe groups configured on the server.</p> <p>Probe Count. The number of probes configured on the server.</p> <p>Probe Group Name. The name of the probe group.</p> <p>Probe Count. The number of probes configured in the probe group.</p>

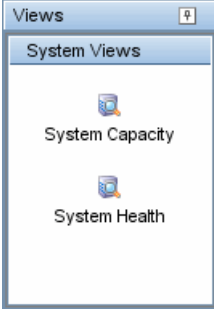
System Health View Description

The System Health view provides overall health information for the components you have installed in your Diagnostics environment.

The following is an example of the System Health view.

System Health									
Table									
Heartbeat	Name	Type	Host	Medi..	Port	Run	Eve... per s..	Vers..	
●	CommandingServer	Comman...	OVRNTT150.ovr...		2006		0 9.00...		
●	server-HPSWROS0...	mediator	HPSWROS018.o...	Commandi...	2612		30 9.00...		
●	server-ovrlxd14...	mediator	opt.rose.hp.com	Commandi...	2612		57 9.00...		
●	server-ovresx1-vm5	mediator	ovresx1-vm5.ov...	Commandi...	2612		0 9.00...		
●	server-ovrntt100	mediator	ovrntt100.ovrtes...	Commandi...	2612		0 9.00...		
●	server-OVRNTT154	mediator	OVRNTT154.ovr...	Commandi...	2612		0 9.00...		
●	server-ovrsun28.ro...	mediator	ovrsun28.rose.h...	Commandi...	2612		100 9.00...		
●	server-ovruxd10.ro...	mediator	ovruxd10.rose.h...	Commandi...	2612		0 9.00...		
●	server-ovruxt32.ro...	mediator	ovruxt32.rose.h...	Commandi...	2612		0 9.00...		
●	TOMCAT_hpswros...	probe	hpswros022.ov...	server-HP...	35001		0 9.00...		
●	TIBCO_BV_hpswr...	probe	hpswros022.ov...	server-HP...	35002		1 9.00...		
●	TIBCO_ESB_hpswr...	probe	hpswros022.ov...	server-HP...	35000		0 9.00...		
●	L91_CallChain2_0...	probe	ovresx1-vm10.o...	server-HP...	35001		5 9.0.2...		
●	L91_TestService2...	probe	ovresx1-vm10.o...	server-HP...	35004		2 9.0.2...		
●	L91_JavaTrader2...	probe	ovresx1-vm10.o...	server-HP...	35000	1	6 9.0.2...		
●	L91_TestService2...	probe	ovresx1-vm10.o...	server-HP...	35002	1	3 9.0.2...		
●	L91_TestService2...	probe	ovresx1-vm10.o...	server-HP...	35003	1	5 9.0.2...		
●	L91_MSPetShop_1...	probe	ovrntt121.ovrtes...	server-HP...	35000		2 8.1.2...		
●	P81_VLS81_MedR...	probe	ovrntt152.ovrtes...	server-HP...	35000		8 9.00...		
●	P81_WAS5_Plants...	probe	ovrntt152.ovrtes...	server-HP...	35001		22 9.00...		
●	L91_WAS60_Plant...	probe	oxycontin.rose...	server-HP...	35000	1	3 9.00...		
●	Collector_ovrntt15...	probe	ovrntt152.ovrtes...	server-HP...	35002		0 9.00...		

Details	
Metric	
BPM transactions	0
BPM transactions ...	0
Diagnostics sampl...	0
Diagnostics sampl...	0
Diagnostics sampl...	0
Diagnostics sampl...	5
Diagnostics sampl...	20912
ELT count since la...	0
Events per sec	0
Events since launch	<unknow
Last update (com...	Wed Ju
Load Runner tran...	0
Load Runner tran...	0
Probe events sinc...	0
Probe-data	ServerP
Up time	12 hour
WDE bytes relayed	189300
WDE bytes relaye...	114054
WDE samples failed	0
WDE samples fail...	0

<p>To access</p>  <p>The screenshot shows a window titled 'Views' with a sub-tab 'System Views'. Inside the window, there are two icons with labels: 'System Capacity' and 'System Health'.</p>	<p>To access the System Views you must access the Diagnostics UI as the Mercury System customer.</p> <p>Access the Mercury System customer in Diagnostics:</p> <p><a href="http://<Diagnostics_Commanding_Server_Name>:2006/query/">http://<Diagnostics_Commanding_Server_Name>:2006/query/</p> <p>For the Mercury System customer, select the link to Open Diagnostics.</p> <p>Log in to Diagnostics and on the Applications window select Entire Enterprise and select any link to open the Diagnostics Views.</p> <p>In the View bar you'll see the System Views view group. Select the System Health view or System Capacity view.</p>
<p>Important information</p>	<p>This view group is only available for the Mercury System customer.</p>
<p>See also</p>	<p>See "Using the System Health Monitor" in the <i>HP Diagnostics Installation and Configuration Guide</i>.</p>

The following user interface elements are included:

UI Elements	Description
Table	<p>Provides overall health information for the components you have installed in your Diagnostics environment. The information is similar to that provided in the System Health Monitor but allows you to filter, sort and select metrics and attributes of interest. This view is useful for managing a large Diagnostics deployment.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Heartbeat. The status of the connection or link to the system in question. It indicates if the entity is up and running and it's health. Green (good performance), Yellow (fair performance), Red (error), Grey (inactive).</p> <p>Name. The system name of the commander, mediator or probe system.</p> <p>Type. Can be the following type of Diagnostics component: Commander, mediator, probe.</p> <p>Host. Host system name where the Diagnostics component is installed.</p> <p>Mediator. For probe systems this is the mediator the probe is configured to communicate with. For mediator systems this is Commanding Server. For the commanding server this is blank.</p> <p>Port. Port number the component communicates on.</p> <p>Run. The run number of the process from LoadRunner or Performance Center that is driving the load.</p>

UI Elements	Description
Table (Continued)	<p>Events per second. An indication of how busy the components is how many events it is processing. This field is relevant for .NET probes and some Java probes that continue to use the event channel.</p> <p>Version. Indicates the installed version of the Diagnostics component.</p>
Details pane	<p>Provides detailed information for system administrator for the selected Diagnostics component.</p> <p>Provides metric (for example amount of time the component has been up), configuration (for example the vm and version), and Log file messages.</p>

26

Transactions Views

This chapter gives an overview of transaction monitoring in Diagnostics and describes the specialized views for synthetic transactions and business transactions.

This chapter includes:

Concepts

- ▶ Transactions View Group Overview on page 574
- ▶ Types of Transactions in Diagnostics on page 575
- ▶ Synthetic Transactions and Backend Data from Diagnostics on page 577
- ▶ Business Transaction Monitoring with Diagnostics on page 578
- ▶ Business Transaction Paths in Diagnostics on page 579

Tasks

- ▶ How to Create or Edit Business Transactions in Diagnostics on page 583
- ▶ How to Configure Business Transaction Discovery on page 587

Reference

- ▶ Transactions View Group User Interface on page 591

Concepts

Transactions View Group Overview

The Transactions view group contains the following views:

► **Business Transactions view.**

Business transactions are transactions executed in a Business Application based on real user input. These transactions are defined based on top level server requests which are the starting point of the business transaction.

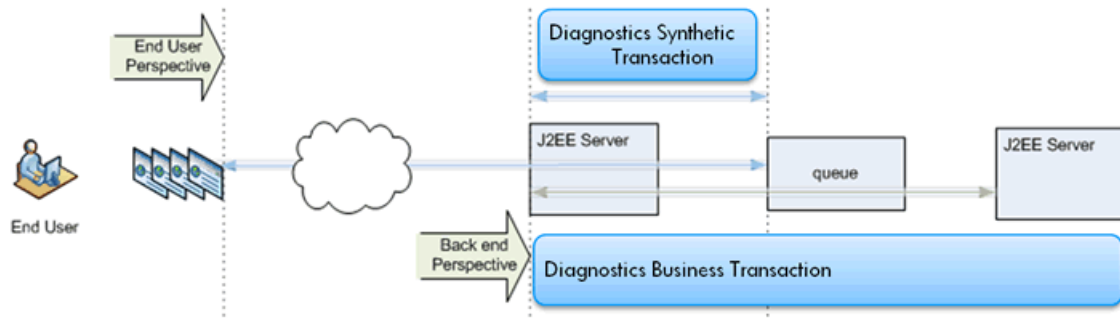
► **Synthetic Transactions view.**

Synthetic transactions are a script generated simulation of a transaction that an end user of the application might perform. These transactions can be generated by Business Process Monitor scripts.

For details on the user interface, see "Transactions View Group User Interface" on page 591.

Types of Transactions in Diagnostics

There are two types of transactions:



Synthetic Transactions

Synthetic monitoring (also known as active monitoring) is performance monitoring that is done using a simulation. Synthetic transactions are generated by Business Process Monitor scripts that simulate a transaction that an end user of the application would perform. These scripts are run periodically and transactions are monitored for availability and response time measures.

Diagnostics collects data generated by these scripts and displays the data in Diagnostics in the Synthetic Transactions view. For details on the **Synthetic Transactions** view in Diagnostics, see "Synthetic Transactions View Description" on page 607.

Synthetic monitoring does not require actual end-user traffic so it enables companies to test web applications 24x7, or test new applications prior to a live monitor launch.

Because synthetic monitoring is a simulation of typical user behavior or navigation through an application or transaction, it is often best used to monitor commonly trafficked paths and critical business processes. Synthetic tests must be scripted in advance, so it is not feasible to measure performance for every permutation of a navigational path an end-user might take. Synthetic testing is useful for measuring availability and response time of critical pages and transactions (how a site performs from all geographies) but doesn't monitor or capture actual end-user interactions.

Business Transactions

A business transaction is configured or discovered while a synthetic transaction is generated, typically, by a script. Business transactions represent real user input as opposed to a synthetic load.

You create business transactions in Business Service Management's Transaction Management application by selecting a Business Application CI and creating a new business transaction in it. After creating a business transaction you can configure it for monitoring and/or tracing depending on the type of information you need. A business transaction is configured for monitoring by selecting a server request to monitor. See the *Business Service Management online help* for details on tracing.

Diagnostics provides backend business transaction monitoring which is aggregate data. If you need transaction instance level data then you should use Transaction Management to configure tracing.

When configuring a business transaction for Diagnostics monitoring, you define the transaction based on top-level server requests known to Diagnostics. A **top-level server request** is one with no inbound calls. Diagnostics will automatically discover and add downstream server requests to the business transaction based on the top level server request you select. Diagnostics captures server requests that make up a business transaction and provides relevant performance metrics.

Business transactions created in Transaction Management and configured for monitoring are displayed in both Transaction Management reports and in the Diagnostics Business Transactions view. And drill downs from the Transaction Management reports to Diagnostics are available. For details on the **Business Transactions view** in Diagnostics, see "Business Transactions View Description" on page 591.

For stand alone Diagnostics users, business transactions can be created in the Business Transactions view group by selecting the Create Business Transactions task. For how to create business transactions in Diagnostics, see "How to Create or Edit Business Transactions in Diagnostics" on page 583.

Note: Business Transactions you create in Diagnostics are not shared with Business Service Management. But business transactions discovered in Diagnostics based on discovery rules can be shared with Business Service Management and populate CIs in the Run-time Service Model, if you specify an `appName` variable in the rule.

Synthetic Transactions and Backend Data from Diagnostics

When a synthetic transaction (generated from Business Process Monitor scripts) hits a server or application instrumented with a Diagnostics probe, Diagnostics detects the transaction. And Diagnostics collects data on all the backend activity related to the transaction.

Diagnostics creates a synthetic transaction in Diagnostics and adds the server requests that were part of the synthetic transaction.

Diagnostics provides two important data points that allow identification of the backend bottlenecks for a particular transaction:

- ▶ The response time of the transaction from the backend perspective
- ▶ All server requests that were executed during the transaction.

Diagnostics also provides data to Business Service Management for these synthetic transactions. Diagnostics populates the Run-time Service Model with application infrastructure (J2EE and .NET) data for detected synthetic transactions. This allows you to see front end and backend activity for a Business Process Monitor transaction.

Also in Business Service Management, for Business Transactions from Business Process Monitor, Diagnostics provides status for several synthetic transaction health indicators. This status is based on backend activity monitored by Diagnostics. With this information you can do quick triage in Business Service Management to see if the problem is in the backend. See "Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics" on page 1098.

Business Transaction Monitoring with Diagnostics

Managing the performance of business transactions can involve transaction monitoring and transaction tracing. The Transaction Management application allows you to configure a business transaction for monitoring and tracing. This section only deals with monitoring. For information on tracing see the *Transaction Management online help*.

In the Transaction Management application, transactions are flagged for monitoring. The common data model for business applications and business transactions across Business Service Management and Diagnostics means you can easily define a transaction in the Transaction Management application and monitor the same transaction using Diagnostics.

Transaction monitoring by Diagnostics is based on aggregate server requests. Metrics include latency, volume, exceptions and SOAP faults. Monitoring provides for transaction topology based on aggregate server request data and allows you to follow the outbound calls for a top level server request to other server requests.

A central requirement for managing business transactions is for IT to understand how transactions relate to the infrastructure components that support them. Diagnostics provides extensive system and application infrastructure data (deep diagnostics view into middleware) for the components that support each step in any given transaction.

Business Transaction Paths in Diagnostics

When Diagnostics monitors a business transaction it provides the aggregate latency of a chain of server requests that typically span multiple probes.

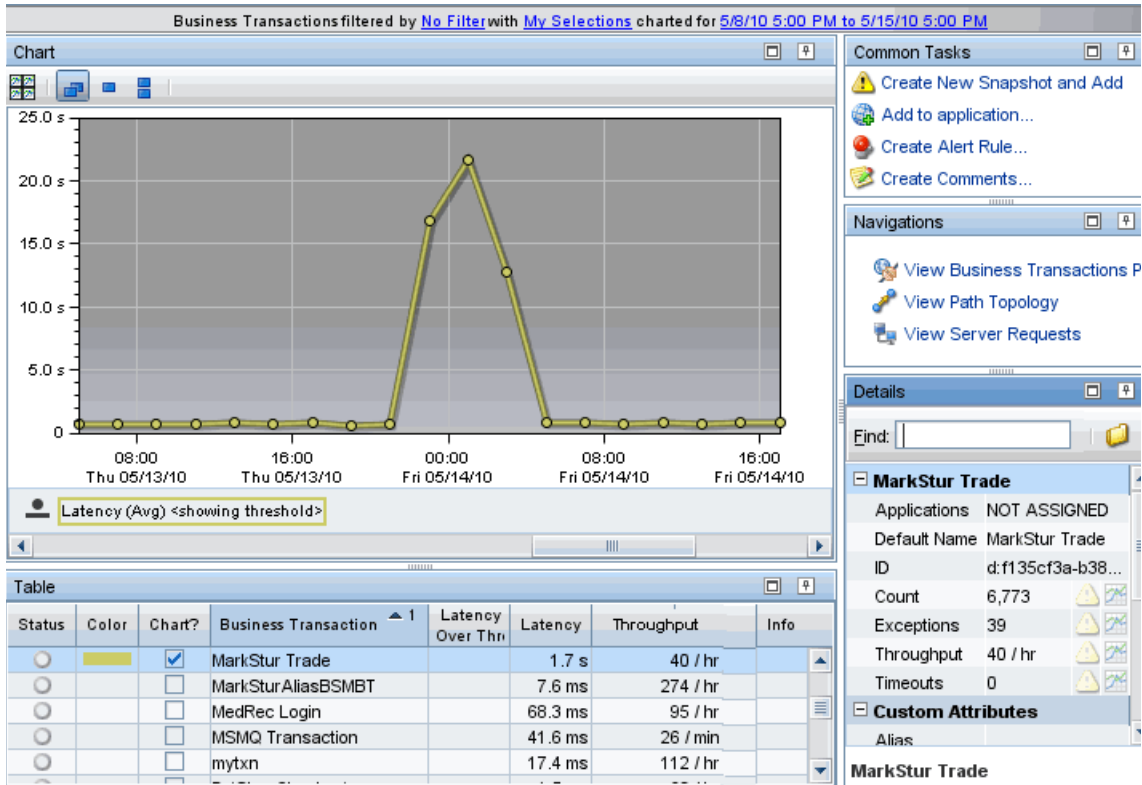
A business transaction is configured by selecting a **top-level (starting) server request** as an entry point into the transaction. A top level server request is one which does not have any inbound calls. The probes and server automatically identify all the other server requests that are reachable by this top-level server request.

A **Business Transaction Path** is defined by all server requests starting with the top level server requests to any other reachable server requests (including the top level server request).

When the business transaction is **synchronous**, the aggregate latency is calculated by just adding the latency for all the associated server requests.

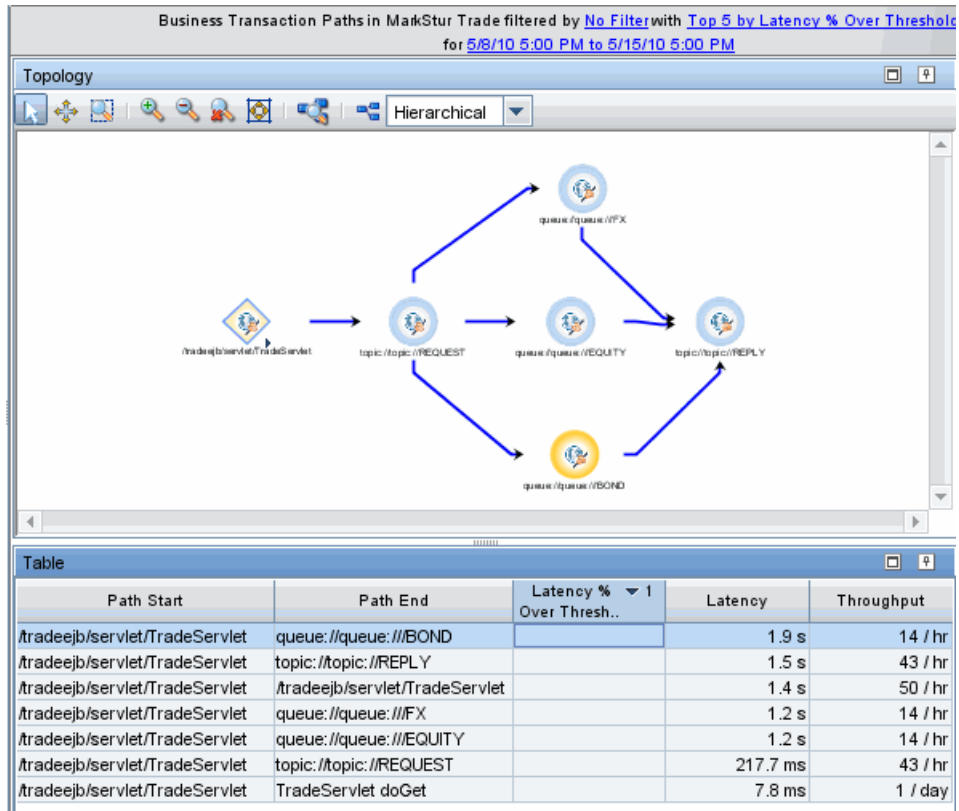
Aggregate latency can also be calculated for a business transaction that has **asynchronous** server requests. For example, if an item is put into a queue and processed at a later time, Diagnostics will take the wait time for this item into account when calculating latency.

In the example below you see the Trades business transaction with typically 1.0s average latency. But you see there is a spike up to 20.0s. So over the last hour the average latency is 1.7s. You can look at the Path Topology view to find out more about where the bulk of the time was spent.

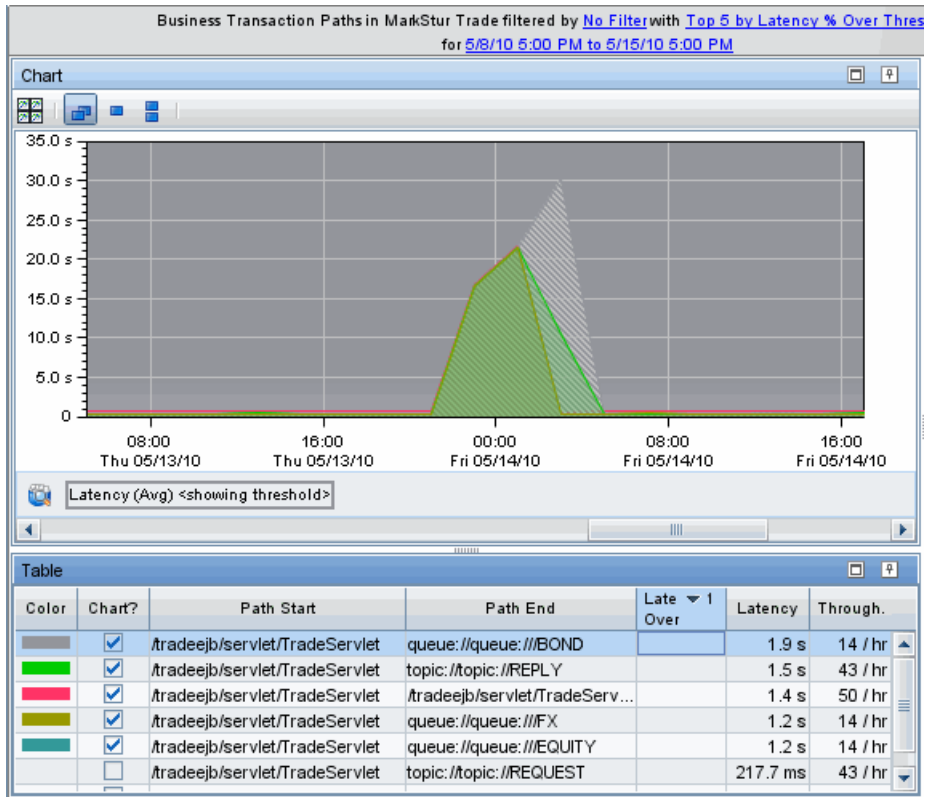


When the server requests make asynchronous calls, Diagnostics uses an algorithm to calculate business transaction times from the starting server request to the end of either that server request or one of its outbound calls (and chained calls) whatever takes the longest to finish.

The Path Topology example below shows the Path Start and Path End for each of 7 paths (including the top level server request). The latency value shown in the table is from top level server request (the left most icon in the topology) to one of the other server requests. The queue://queue:///Bond server request with latency of 1.9s is the longest path on average. This indicates the transaction is spending more time wait time in this path than in the others.



You can see the same information with the trend per path by looking at the Business Transaction Paths view as shown below:



To calculate the Business Transaction Paths, the probe collects the start information and start time across outbound calls so that the end fragments can record start-to-end time. The business transaction paths are aggregated by the probe and then sent to the server which analyzes them to decide which path best represents the business transaction time (this is based on a configurable 1m sample window). If the outbound calls were asynchronous, the server picks the path that averaged the longest amount of time and uses that to represent average business transaction latency. If everything was synchronous, then the average business transaction latency is the same as the latency for the root server request.

Tasks

How to Create or Edit Business Transactions in Diagnostics

The following steps describe how to create or edit business transactions in Diagnostics:

- "Prerequisites" on page 584
- "Access the Create/Edit Business Transaction Dialog Box" on page 584
- "Use the New Business Transaction Dialog Box" on page 585
- "Edit a Business Transaction" on page 587
- "Delete a Business Transaction" on page 587
- "Results" on page 587

Note: Business Transactions you create in Diagnostics are not shared with Business Service Management. But business transactions discovered in Diagnostics based on discovery rules can be shared with Business Service Management and populate CIs in the Run-time Service Model, if you specify an `appName` variable in the rule.

Prerequisites

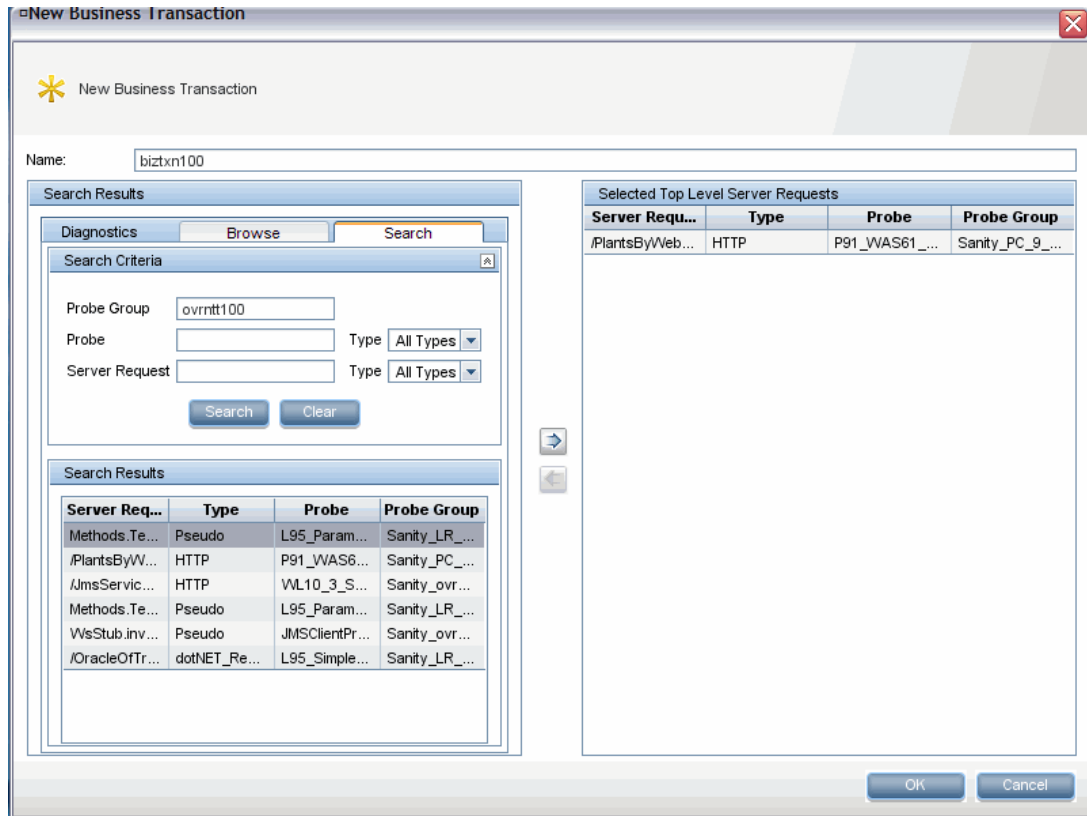
In order to create a new business transaction for monitoring you will need to be able to identify the top level server requests you want to assign to the new transaction. You can browse or search for server requests based on probe group, probe or server request type.

Access the Create/Edit Business Transaction Dialog Box

Access the Create Business Transaction dialog box from the Common Tasks menu in the Business Transactions view (in the Transactions view group).

Example:

An example of the Create/Edit Business Transactions dialog box is shown below:



Use the New Business Transaction Dialog Box

Create a new business transaction using the New Business Transaction dialog box as follows:

Name. Enter a name for the business transaction. This is a required field. For example, you could enter a name based on what the server request is doing (for example, login).

Search Results Pane. Top level server requests are selected to be used to define this business transaction. In general you select a server request that represents an important business transaction in your enterprise. See "How to Configure Business Transaction Discovery" on page 587 for an alternative way to define business transactions in Diagnostics.

You can use the Browse or Search tabs to locate the server request. Diagnostics will automatically discover and add downstream server requests to the business transaction based on the top level server request you select.

Search Tab. Select the Search tab and the left pane displays a set of filters that allows you to search by Probe Group, Probe or Server Request. Enter a string that the probe group, probe or server request contains. The string is case sensitive. Any entity name that contains the request string will be returned. You can also search based on probe type (Java or .NET) or server request type. The type filters use an equal function and will filter out all entities that are not of the type selected. To search over all types leave (or return) the drop down box selection to "All Types". Search finds top level server requests up to a week old.

Browse Tab. Select the Browse tab and the left pane displays a list of probe groups. Select + to open a probe group/probe folder and see a list of top level server requests.

Selected Top Level Server Requests Pane. Select a top level server request and use the arrow key in the center of the dialog box to move the item to the Selected Top Level Server Requests pane.

You can assign additional top level server requests to the transaction. When this is the case the assumption is that the transaction is load balanced. Therefore only same named server requests can be assigned to a transaction. If you attempt to assign different named transactions to a business transaction you will receive the following message:

"The following server requests cannot be added:
<list of server request user attempted to add that cannot be added>
Only multiple top-level server requests with the same name can be combined to define the business transaction; When this is the case the top-level server requests are assumed to part of the same cluster. "

Click **OK** when you are done.

The new business transaction will be displayed in the Business Transaction view. Business transactions created using this dialog box will have an ID (shown in the Details pane) starting with d:

Edit a Business Transaction

An existing business transaction can be edited by selecting the business transaction in the Business Transactions entity table and right-clicking to select **Edit Business Transaction**. You can change the business transaction name and the assigned server requests.

If you change the server request assigned to the business transaction, the metrics, topology and paths will be based on the new server request.

Delete a Business Transaction

An existing business transaction can be deleted by selecting the business transaction in the Business Transaction entity table and right-clicking to select **Delete Business Transaction**.

The business transaction continues to be monitored until it ages out of diagnostics. The data begins aging out right away. So for example if you are looking at five minute data, you will not see the old data after five minutes.

Results

Once you have created a business transaction in Diagnostics using the Create Business Transaction dialog box you can monitor transactions in the Business Transactions view, Business Transactions Path view and Business Transaction Topology view. You can also set alarm thresholds and other functions as you would with other entities in Diagnostics.

Diagnostics will automatically discover and add downstream server requests to the business transaction you create in Diagnostics based on the top level server request you select.

How to Configure Business Transaction Discovery

You can use rules based business transaction discovery for situations where capturing the server requests with rules is more practical than defining them individually. For example, when a high number of server requests are expected in the future and you want them mapped to business transactions.

If you want the business transactions discovered based on rules you set in Diagnostics to also be shared with Business Service Management and populate CIs in the Run-time Service Model, you must specify an `appName` variable in the rule.

- ▶ "Enable Business Transaction Discovery" on page 588
- ▶ "Configure Rules for Business Transaction Discovery" on page 588
- ▶ "Results" on page 589

Enable Business Transaction Discovery

The `biztxn.discovery.frequency` parameter in the `server.properties` file controls whether discovery is enabled and how frequently the Business Transaction Discovery Rules are evaluated.

By default business transaction discovery is not enabled, which means, `biztxn.discovery.frequency` is set to zero by default.

When enabled, a frequency of five minutes is suggested.

The Business Transaction Discovery program is run on a scheduled basis on the Diagnostics mediating servers.

Configure Rules for Business Transaction Discovery

Business transaction discovery is based on rules used to identify server requests that are expected to come in and map them to a business transaction name.

The rules are defined in the `/etc/bizTxnDiscoveryRules.properties` file located on the Diagnostics mediators.

If you want the discovered business transactions to also be shared with Business Service Management, you must specify an `appName` in the rule, without `appName` in the rule they are viewable only in Diagnostics. Refer to the example in `bizTxnDiscoveryRules.properties` with `appName` specified.

If the `appName` variable is also set, then the business transaction will be added to that application. The application will be created if necessary. Changing the rule to use a different `appName` would create new business transactions not modify existing transactions. Adding or changing application assignments in the Diagnostics UI will not have an affect on the Business Service Management integration.

The rule evaluation is done on the mediators to make the rules processing more efficient and spread the load off the commanding server. See the comments in the property file for details on how to write a query rule and some examples.

Tip: You can use `testmode=true` when testing new rules. Check the `biztxn.log` for results. Set `biztxn` logging to debug, if needed. Use `testmode=false` when the rules are ready to run and create/update Business Transactions.

When business transaction discovery is enabled this is what the default (example) rule does:

- ▶ Find top-level server requests that have outbound calls to JMS or Web services. These types of server requests are more likely to be "interesting" top-level server requests.
- ▶ Create business transactions for these with a name of **<server request name> added by rule default**.

Custom rules will be needed for your unique environment. The default limit is ten per mediating server. The limit is enforced by tracking discovered business transactions in `<mediator>/log/bizTxnAdded.properties`. If that file is cleared it will affect the limit tracking. To increase the limit, increase the `limit=` setting in `bizTxnDiscovery.properties` file.

Results

The business transactions discovered based on rules will be displayed in the Diagnostics Business Transaction view. Business transactions discovered by rules will have an ID (shown in the Details pane) starting with `r:`.

If the query rule you specified included an **appName** then these business transactions are also shared with Business Service Management and populate CIs in the Run-time Service Model.

Reference

Transactions View Group User Interface

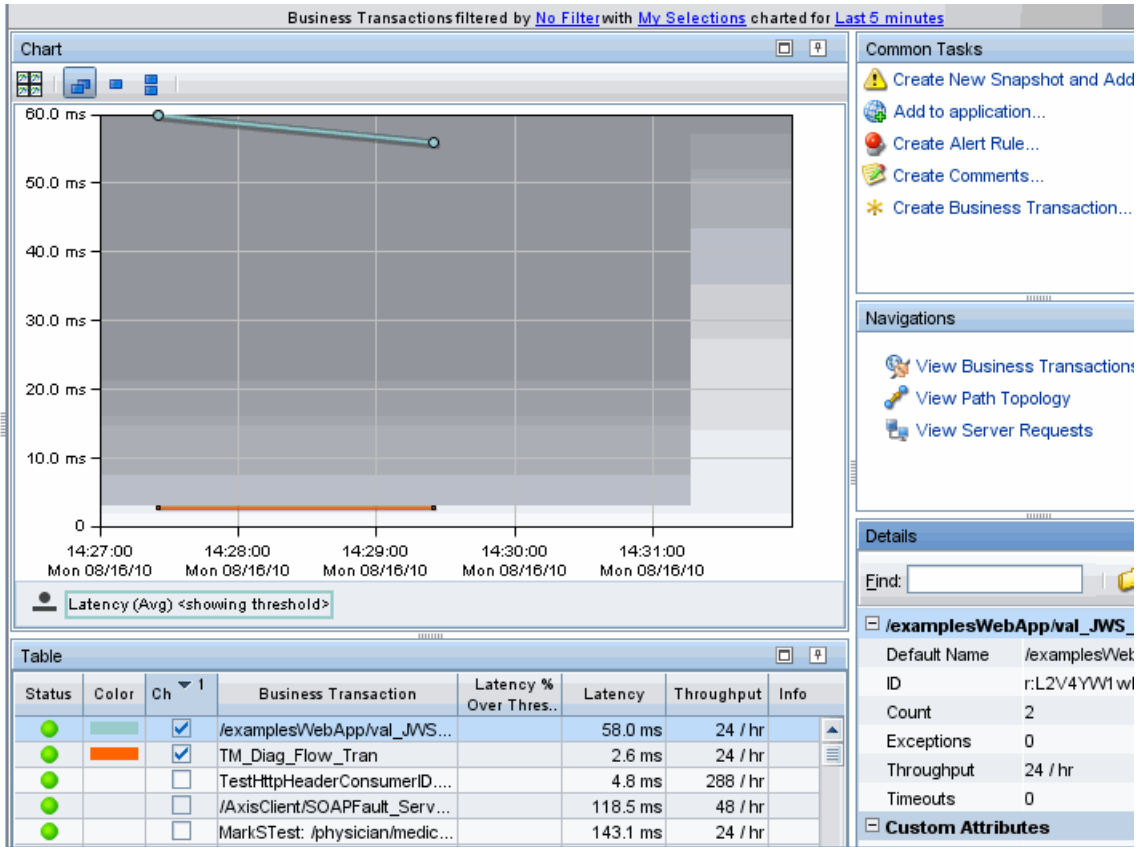
This section includes:

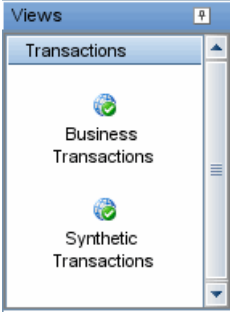
- ▶ Business Transactions View Description on page 591
- ▶ Business Transactions Paths View Description on page 597
- ▶ Path Topology View Description on page 602
- ▶ Synthetic Transactions View Description on page 607

Business Transactions View Description

The Business Transactions view allows you to see latencies and throughput, set thresholds and drill to see the business transaction paths, topology and server requests.

The following is an example of the Business Transactions view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click Transactions to open the Transactions view group.</p> <p>If the Transactions view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select Transactions and click OK. The Transactions view group is displayed in the View bar.</p> <p>Select the appropriate view from the Transactions view group.</p>
<p>Relevant tasks</p>	<ul style="list-style-type: none"> ▶ "How to Create or Edit Business Transactions in Diagnostics" on page 583 ▶ "How to Configure Business Transaction Discovery" on page 587

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five business transactions with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Transaction Details tooltip for the trend line on the graph (or the Business Transaction column in the table) displays the following detail data:</p> <p>Transaction. The name of the transaction.</p> <p>The Transaction Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
<p>Table</p>	<p>Lists the business transactions that pertain to the context selected. The metric reported in the table is aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column header to select Customize Columns and specify which metrics or system attributes you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the business transaction. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Business Transaction. The name of the business transaction.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

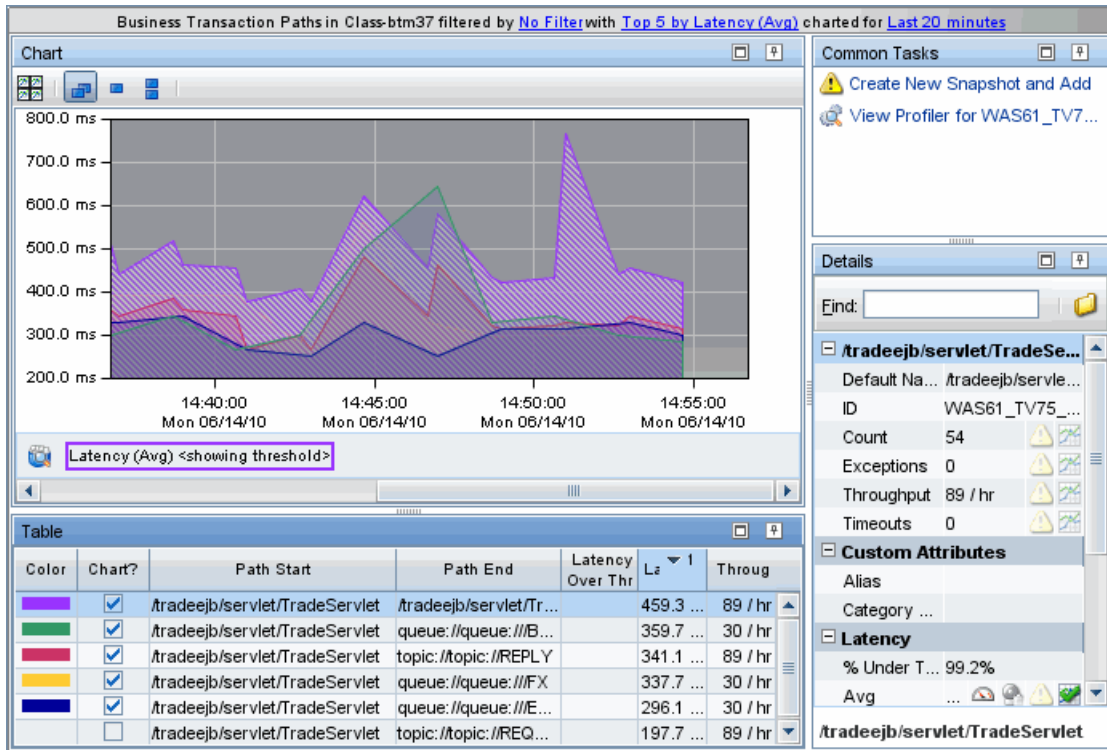
UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the business transactions displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Business Transactions Paths. Another common drill down is to the Server Requests view to see related information about the request.</p>
Details Pane	<p>Shows general latency metrics collected on the business transaction.</p> <p>Also the following information is shown, if available:</p> <p>SOAP Faults. Total number of sender SOAP faults.</p> <p>Business Transactions created in Diagnostics (as opposed to those created in Diagnostics based on rules or defined in the Transaction Management application) are only viewed in Diagnostics, not in Business Service Management.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Business Transactions Paths View Description

The Business Transactions Paths view displays a superimposed area chart of average latency for each path of the selected business transaction. The areas are slightly transparent and on top of each other as opposed to the Load or Layers views which are stacked area charts.

A Business Transaction Path represents the aggregate latency from the starting Server Request (Path Start) to an end point of the transaction (Path End). See "Business Transaction Paths in Diagnostics" on page 579 for details and examples of how the latency is calculated for the business transaction path.

The following is an example of the Business Transactions Paths view,



To access	Access the Business Transaction Paths view by drilling down from a business transaction in the Business Transactions view.
See also	"Business Transaction Paths in Diagnostics" on page 579

The following common user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency for a business transaction path, as a superimposed area chart.</p> <p>The path chart identifies all possible paths with each area chart visualizing the latency of that path.</p> <p>The path with the maximum area chart is used to determine the business transaction latency - thus the path used to determine that business transaction latency may vary over time. Think of this as if you drew a line along the topmost part of all the area charts (maximizing the area under the line).</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Transactions Path Details tooltip for the superimposed area chart (or the Path Start column in the table) displays the following detail data:</p> <p>Transaction Path. Path for the transaction. Based on the top level server request and automatically discovered outbound server requests. A Business Transaction Path is defined as a path from the top level server request to any other reachable server request.</p> <p>Transaction. The transaction name.</p> <p>Probe. The name of the probe that is executing the server request.</p> <p>Probe Group. The name of the probe group that contains the probe.</p> <p>And the tooltip can include additional information like the Server Request or Web Service and Operation Name depending on the type of server request.</p>

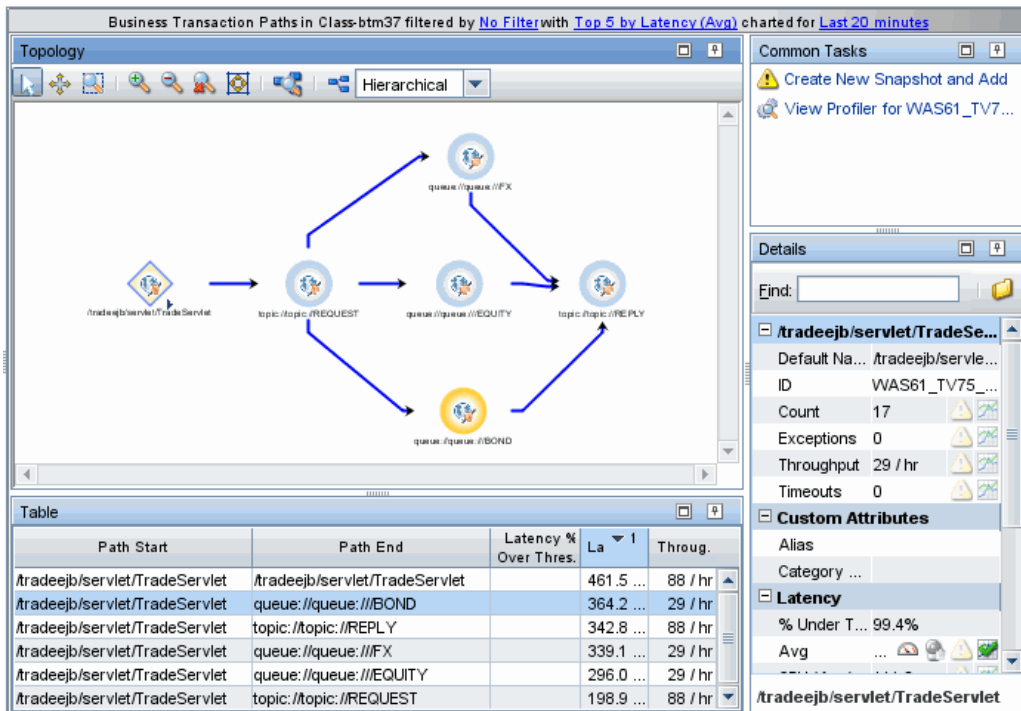
UI Elements	Description
Table	<p>The table identifies the start and end node of each path along with its latency. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Path Start. Server Request that started the business transaction.</p> <p>Path End. Server Request at the end of the business transaction.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p>
Common Tasks	Common tasks and actions relevant to the selected entity, for example, Create New Snapshot .
Details Pane	Shows metrics collected on the entity selected. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

Path Topology View Description

The Path Topology view provides a visual representation (topology) of the paths for a selected business transaction. In the topology, you see the path from the starting (Top-Level) server request to the server requests in the chain. The path topology diagram identifies possible paths for a business transaction with each node representing the end of a path.

A Business Transaction Path represents the aggregate latency from the starting Server Request (Path Start) to an end point of the transaction (Path End). See "Business Transaction Paths in Diagnostics" on page 579 for details and examples of how the latency is calculated for the business transaction path.

The following is an example of the Path Topology view.



To access	Access the Path Topology view by drilling down from a business transaction in the Business Transactions view.
See also	For details about the controls in a topology view, see Chapter 8, "Working with Topologies". "Business Transaction Paths in Diagnostics" on page 579

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.
Topology Diagram	<p>Displays a topology diagram of the business transaction paths based on the top level server request and automatically discovered outbound server requests.</p> <p>The path topology diagram identifies all possible paths with each node representing the end of a path.</p> <p>The topology display includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by changing diagram type or turning off and on the display of the legend. To find out how to use the diagram tool bar, see Chapter 8, "Working with Topologies."</p> <p>You can right-click items in the topology diagram to see a tooltip containing information the entity.</p> <p>The topology diagram and the table are integrated so that if you select an entity in the diagram then the relevant row in the table is highlighted.</p>

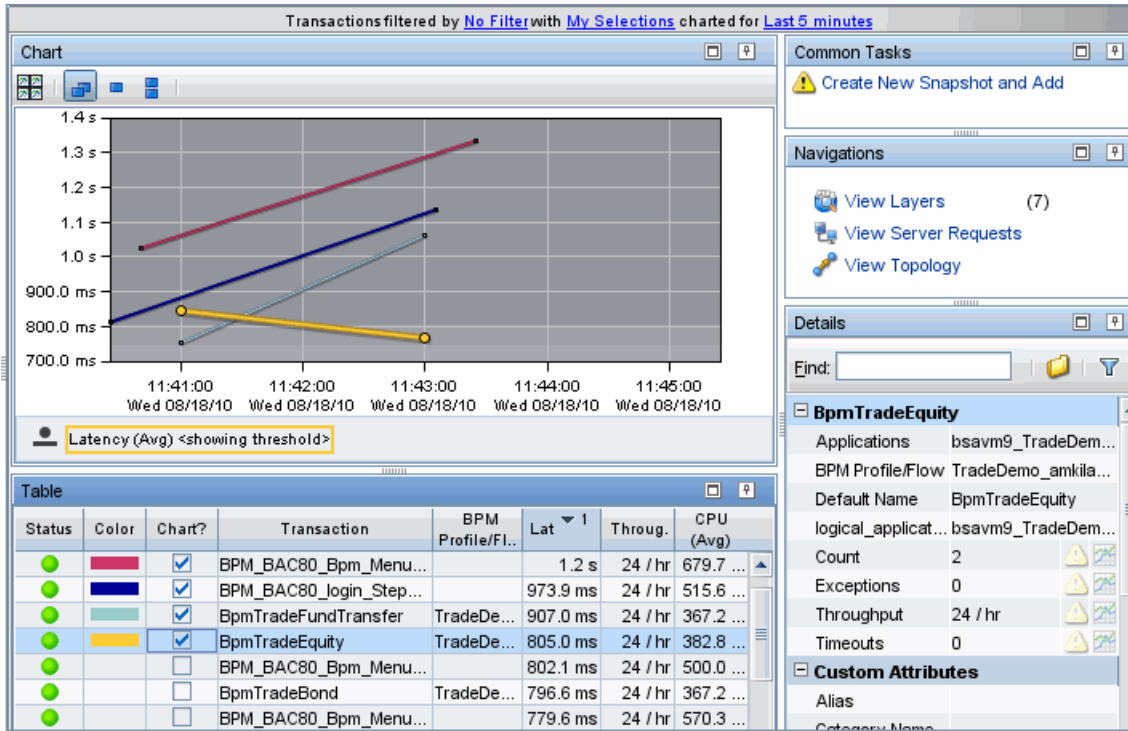
UI Elements	Description
<p>Table</p>	<p>Lists the start and end node of each path of the business transaction along with its latency.</p> <p>A Business Transaction Path is identified by the top level server request to any other reachable server request.</p> <p>The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Path Start. Server Request that started the business transaction.</p> <p>Path End. Server Request at the end of the business transaction.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p>

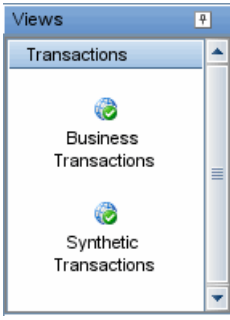
UI Elements	Description
Table (continued)	<p>Transactions Path Details tooltip for the Path Start column in the table displays the following detail data:</p> <p>Transaction Path. Path for the transaction. Based on the top level server request and automatically discovered outbound server requests. A Business Transaction Path is defined as all server requests starting with the top level server request to any other reachable server request.</p> <p>Transaction. The transaction name.</p> <p>Probe. The name of the probe that is executing the server request.</p> <p>Probe Group. The name of the probe group that contains the probe.</p> <p>And the tooltip can include additional information like the Server Request or Web Service and Operation Name depending on the type of server request.</p>
Common Tasks and Navigations	Common tasks and actions relevant to the business transaction displayed in this view, for example, Create a Snapshot.
Details Pane	<p>Shows metrics collected on the business transaction path starting server request.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Synthetic Transactions View Description

The Synthetic Transactions view displays performance metrics for the transactions Diagnostics monitors that are being executed from Business Process Monitor or from LoadRunner or Performance Center scripts.

The following is an example of the Synthetic Transactions view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click Transactions to open the Transactions view group.</p> <p>If the Transactions view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select Transactions and click OK. The Transactions view group is displayed in the View bar.</p> <p>Select the appropriate view from the Transactions view group.</p>
<p>See also</p>	<p>"Synthetic Transactions and Backend Data from Diagnostics" on page 577</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five synthetic transactions with the highest average latency during the selected time range are charted.</p> <p>Transaction data can come from Business Process Monitors or from LoadRunner or Performance Center scripts if integrated with those products.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Transaction Details tooltip for the trend line on the graph (or the Transaction column in the table) displays the following detail data:</p> <p>Transaction. The name of the synthetic transaction.</p> <p>The Transaction Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the transactions that pertain to the context selected. The metric reported in the table is aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column header to select Customize Columns and specify which metrics or system attributes you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the synthetic transaction.</p> <p>The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart? From the table you can select the entities that are to be charted on the graph.</p> <p>Transaction. The name of the transaction.</p> <p>BPM Profile/Flow. If the transaction is driven by a BPM monitor then the profile this transaction belongs to is indicated in this field.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the transaction displayed in this view. The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs include View Layers and View Topology.
Details Pane	Shows metrics collected on the synthetic transaction. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

27

Portals Views

This chapter describes the specialized views for Portals.

This chapter includes:

Concepts

- ▶ Portals Overview on page 614
- ▶ Configure Collection of CPU Time Metrics for Portlets on page 616
- ▶ Instance Tree Call Profiles for Portlets on page 618

Tasks

- ▶ How to Use the Portals View Group on page 621

Reference

- ▶ Portals View Group User Interface on page 626

Concepts

Portals Overview

The **Portals** view group displays performance metrics for portlets (and other portal components such as books and pages) that are managed in the following portals:

- ▶ BEA WebLogic Portal
- ▶ IBM WebSphere Portal
- ▶ SAP NetWeaver Portal
- ▶ Oracle Portal

For detailed information on supported versions, see the Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp or contact customer support.

If you are using Diagnostics on its own, the Portals view group contains the Server Summary view and the Portal Components view.

If you are using Diagnostics with Business Service Management, the Portals view group contains the Business Process Summary view.

If you are using Diagnostics with LoadRunner or Performance Center, the Portals view group contains the Scenario Summary view.

Portal Summary Views

The Portals view group contains a summary view that provides a high-level overview of portal related metrics. Depending on how you access Diagnostics (integrated with another HP product or on its own), the Portal summary view is presented differently and includes different concise views.

If you are using Diagnostics standalone, the Portals view group contains the Server Summary view that is made up of the following views:

- ▶ **Status.** A monitoring version of the standard Diagnostics Status view.

- ▶ **Server Requests.** A monitoring version of the standard Diagnostics Server Requests view.
- ▶ **Portal Components.** A monitoring version of the Portal Components view. For more information about the Portal Components view, see "Portal Components View Description" on page 636.

For details on the user interface, see "Portal Server Summary View Description" on page 626.

If you are using Diagnostics with Business Service Management, the Portals view group contains the Business Process Summary view which is made up of the following:

- ▶ **Status.** A monitoring version of the standard Diagnostics Status view. For more information about the Status view, see Chapter 23, "Status View."
- ▶ **Transactions.** A monitoring version of the standard Diagnostics Transactions view. For more information about the Transactions view, see "Transactions Views" on page 573.
- ▶ **Portals Components.** A monitoring version of the Portal Components view. For more information about the Portal Components view, see "Portal Components View Description" on page 636.

If you are using Diagnostics with LoadRunner or Performance Center, the Portals view group contains the Scenario Summary view which is made up of the following views:

- ▶ **Transactions.** A monitoring version of the standard Diagnostics Transactions view. For more information about the Transactions view, see "Transactions Views" on page 573.
- ▶ **Server Requests.** A monitoring version of the standard Diagnostics Server Requests view. For more information about the Server Requests view, see "Server Requests View" on page 379.
- ▶ **Portal Components.** A monitoring version of the Portal Components view. For more information about the Portal Components view, see "Portal Components View Description" on page 636
- ▶ **Load.** A monitoring version of the standard Diagnostics Load view. For more information about the Load view, see "Load View" on page 507.

Portal Components View

The Portal Components view displays average latency trends for portal components (portlets, pages, books, iViews) that are monitored by Diagnostics.

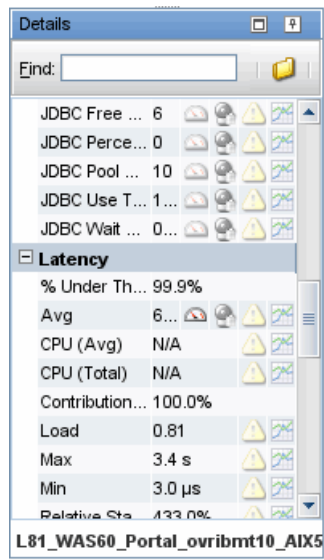
Caution: The Portal Components view can be used only with a Diagnostics probe from version 7.0 or later.

For details on the user interface, see "Portal Components View Description" on page 636.

Configure Collection of CPU Time Metrics for Portlets

The CPU time metrics are **CPU (Avg)** and **CPU (Total)**. If you want to have CPU time for portlets collected and displayed in the Details pane, CPU time collection needs to be configured for portlets in the relevant probe, or configured for All Methods.

If collection of CPU time metrics is disabled or not configured for portlets or all methods, you will see N/A in the Portal Components Details pane for these metrics.



Caution: In VMware, the CPU time metric is from the perspective of the guest operating system and is affected by the VMware virtual timer. See the VMware whitepaper on timekeeping at http://www.vmware.com/pdf/vmware_timekeeping.pdf and for more information, refer to the *HP Diagnostics Installation and Configuration Guide* section on Time Synchronization for probes running on VMware.

Note: Use caution when configuring the collection of CPU timestamps because of the increase in Diagnostics overhead. The increased overhead is caused by an additional call for each method that is needed to collect the timestamp.

You can use either of the following methods to enable collection of CPU time for portlet lifecycle methods or for all methods. Your changes take effect immediately. There is no need to restart the probe.

See "Configure Metrics" on page 168 for details on how to configure collection of CPU Time metrics.

- ▶ In the **dynamic.properties** file for the relevant probe set **cpu.timestamp.collection.method=3** to turn on CPU collections for portlet life cycle methods or **2 to turn on CPU collections** for All methods.
- ▶ Or, in the **Profiler** for the probe, select the **Configuration** tab (use the right arrow next to scroll to find this tab). The profiler does not need to be started to make this configuration change. In the Configuration screen set Collect CPU Timestamps to **For Server Requests and Portlet Methods** or **For Server Requests and All Methods** and apply changes.

Instance Tree Call Profiles for Portlets

The Life Cycle Methods view displays a trend line showing the average of all the instances for a selected method.

Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created for significant method calls that were made. The instance examples let you view, for example, the minimum, maximum and average user and compare this to the running average in the trend line.

Once you've determined which instance to analyze, you can drill down from the icon to see a call profile. You can then compare a 'good' performing call profile with a 'bad' performing call profile. In addition, if you see a pattern in the convergence of the min/max instances with the running trend line you can create a snapshot and look for other events that follow a similar pattern.

The call profile for the portlet is presented in the Call Profile view as a method call argument that depicts the method calls and their latency in a graph and in a table.

The Life Cycle Method call profile only shows the call tree from that method call down. It just shows that unit of work. To see the call tree for the whole server request, which may include other calls, other portlets, you would drill down to the portlet call profile from the Server Requests view. To access the Call Profile view from the Server Requests view, see "How to Use the Server Requests View" on page 385.

Note: You can identify the originating server requests of a portlet by right-clicking the relevant portlet in the Portal Components view and selecting **View Breakdown by Server Request**. After identifying the originating server requests, you can locate them in the Server Requests view and then drill down from those server requests to the Call Profile view.

For more information about instance trees, see Chapter 9, "Working with Instance Trees." For more information about the Call Profile view, see Chapter 16, "Call Profile View."

About Instance Tree Markers

Instance tree markers are displayed for the selected life cycle method. The markers are placed on the graph to indicate the total latency and the end time of the method for which the instance tree was created. For a detailed explanation of instance trees, and how Diagnostics captures them, see Chapter 9, "Working with Instance Trees."

Four instance tree marker icons are used to represent the different types of instance trees that are created for a method:



- ▶ **Maximum Instance Tree.** The maximum instance tree for a life cycle method represents an instance tree with the largest latency for the time period. This tree depicts one of the worst-performing invocations of the life cycle method. This is one tool used for diagnosing the root cause of poor application performance.



- ▶ **Minimum Instance Tree.** The minimum instance tree for a life cycle method represents an instance tree with the smallest latency for the time period. This tree depicts one of the best-performing invocations. It can be useful for comparison to instances that perform poorly.



- ▶ **Average Instance Tree.** The average instance tree for a life cycle method represents an instance tree that represents the average latency for the time period. This tree depicts a typical invocation. It can be useful for comparison to instances that perform poorly.

Tasks

How to Use the Portals View Group

A portal component (portlet) is not a single instrumented entry point. Portal components consist of the top-level methods of the layer type **portlet**. The statistics for these methods are aggregated by name and presented as a single portal component.

The information displayed in the Portal Components view allows you to see a breakdown of Web application server activity by portlet. As a result, you can see where portlet time is spent and you can assess latency (response time) trends.

You can see which portlet, or which layer or life cycle method under the portlet, is contributing the most to a performance problem. You can also view the server requests that contributed to the latency performance of a particular portlet.

From the Navigations pane in the Portal Components view or by right-clicking a Portal Component in the graph entity table you can drill down to the following views:

- View Life Cycle Methods
- View Call Profile of Portlets (Drill down to a call profile view for a portlet)
- View Breakdown by Server Requests
- View Layers

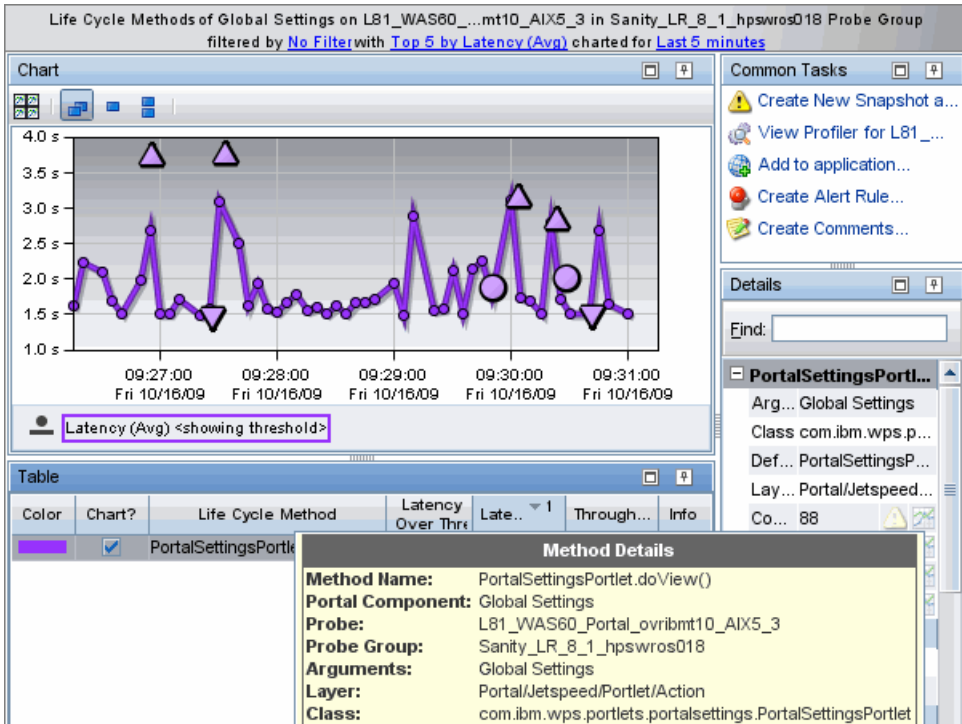
View Life Cycle Methods

You can drill down to the top-level life cycle methods for any portlet. To do this, right-click the relevant portlet in the graph entity table and select **View Life Cycle Methods**. The Life Cycle Methods view opens, displaying the top five life cycle methods with the highest average latency.

The trend line includes small icons to indicate where the minimum, maximum and average instance trees were recorded. You can select any of these icons to see the call profile for the instances.

Example:

An example of the Life Cycle Methods view is shown below:

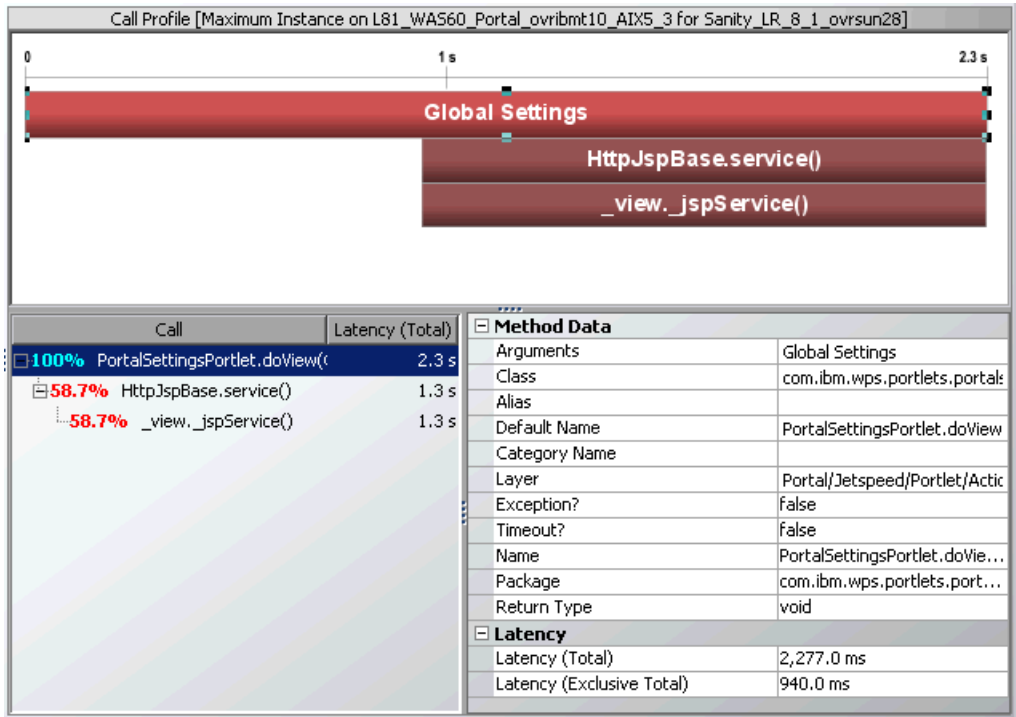


View Call Profile of Portlets

You can view the portlet in the Call Profile view as a method call argument. You access the Call Profile view by selecting an instance (Min, Max, Avg) on the trend line in the Life Cycle Methods view.

Example:

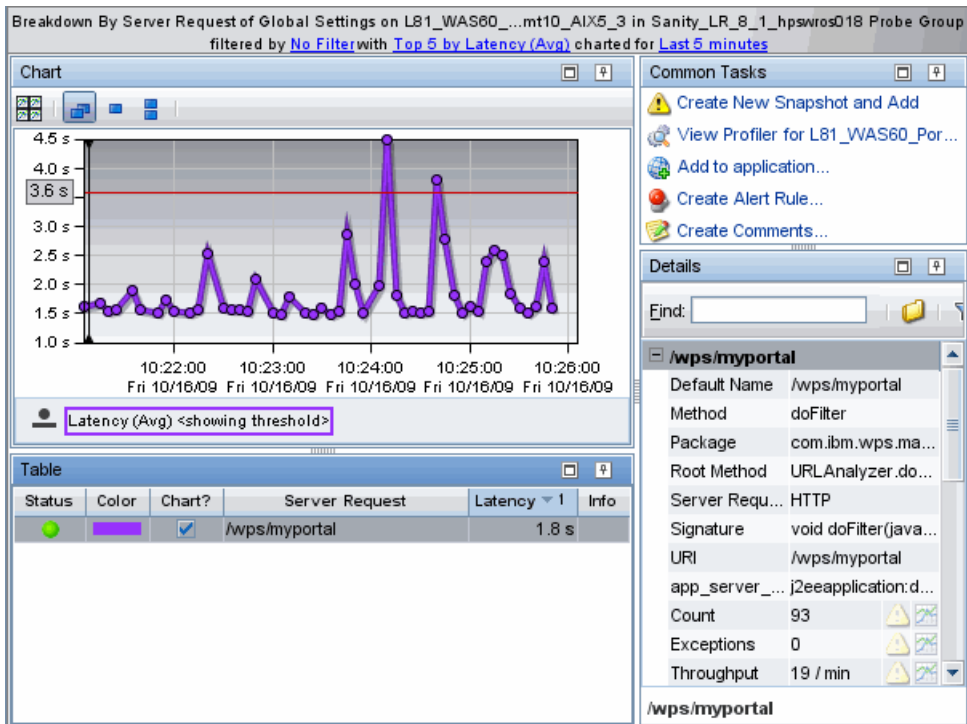
In the following example of an instance tree in the Call Profile view, a method call containing a portlet argument is displayed in the call tree table. The corresponding call box in the call profile graph is labelled by the name of the portlet (not the whole method). For more information about the Call Profile view, see Chapter 16, "Call Profile View." Data about the selected method call is displayed in the Details pane.

**View Breakdown by Server Requests**

You can drill down to view a breakdown of the server requests that contributed to the latency performance of a particular portlet. For example, you may want to view the server requests breakdown to determine whether the behavior of a portlet varied greatly from one server request to another.

Right-click the relevant portlet in the graph entity table and select **View Breakdown by Server Request** or select the view in the Navigations pane.

Example:

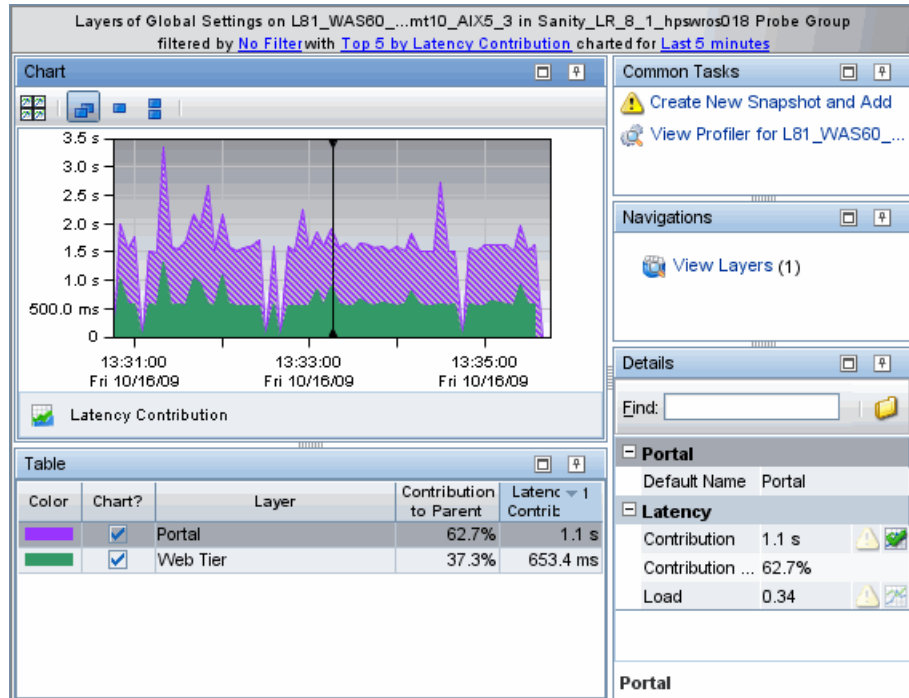


View Layers

You can drill down to the layers that contribute to a portlet's latency performance. Right-click the relevant portlet in the graph entity table and select **View Layers** or select this view in the Navigations pane. By default, Diagnostics displays the latency contribution for the five layers that have the highest percent contribution values during the currently selected time frame.

Example:

Diagnostics displays the latency contribution for each layer using a stacked area graph.



Reference

Portals View Group User Interface

This section includes:

- ▶ Portal Server Summary View Description on page 626
- ▶ Scenario Summary with Portlets View Description on page 630
- ▶ Business Process Summary with Portlets View Description on page 634
- ▶ Portal Components View Description on page 636
- ▶ Life Cycle Methods View Description on page 642
- ▶ Breakdown by Server Requests on page 645

Portal Server Summary View Description

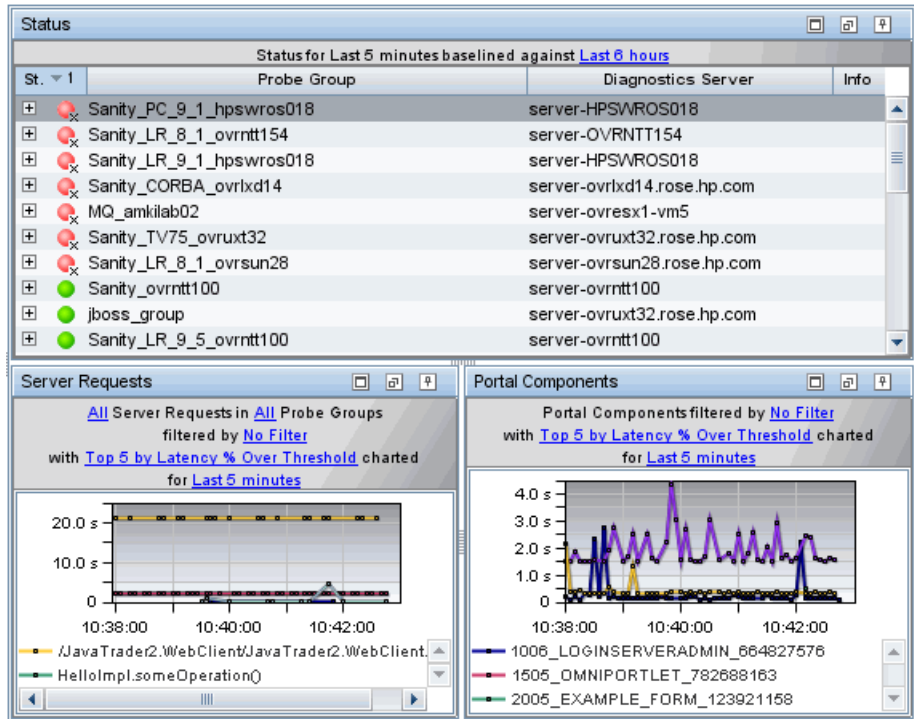
The Portal Server Summary view provides a high-level overview of portal related metrics.

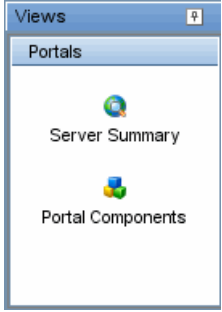
Depending on how you access Diagnostics (integrated with another HP product or on its own), the Portal summary view is presented differently and includes different concise views. See "Portals Overview" on page 614 for information on the Portal Server Summary view when accessed from Business Service Management, LoadRunner or Performance Center.

If you are using Diagnostics on its own, the Portals Server Summary view is a dashboard view that contains the following:

- ▶ Status table
- ▶ Server Requests graph
- ▶ Portal Components graph

The following is an example of the Portal Server Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click Portals to open the Portals view group.</p> <p>If the Portals view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select Portals and click OK. The Portals view group is displayed in the View bar.</p> <p>Select the appropriate view from the Portals view group.</p>
<p>Important information</p>	<p>Depending on how you access Diagnostics (integrated with another HP product or in standalone mode), the Portal summary view is presented differently and includes different concise views.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Portals View Group" on page 621.</p>
<p>See also</p>	<ul style="list-style-type: none"> ▶ "Status View Description" on page 520 ▶ "Server Requests View Description" on page 389

Status Table

The following user interface elements are included:

UI Elements	Description
<p>Status Table</p>	<p>A monitoring version of the standard Diagnostics Status view. All Diagnostics probe groups, probes, and hosts are displayed in this view, not just those that are portal-related.</p> <p>For more information about the Status view, see "Status View Description" on page 520.</p>

Portal Server Summary Graphs

The Portal Server Summary view contains monitoring versions (charts/ graphs) of the Server Requests view and Portal Components view,

UI Elements	Description
<p>Server Requests graph</p>	<p>By default, the graph displays the top five server requests with the highest latency percent over threshold during the previous five minutes.</p> <p>By default, the graph is based on all server requests for all probe groups. You can filter the data by server requests or individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>Server Request Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Method Signature. The signature of the root method.</p> <p>The Server Request Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Server Requests view.</p>

UI Elements	Description
<p>Portal Components graph</p>	<p>By default, the graph displays the top five portal components with the highest latency percent over threshold during the previous five minutes.</p> <p>By default, the graph is based on all server requests for all probe groups. You can filter the data by server requests or individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>Portal Component Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>The Portal Component Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Portal Components view.</p>

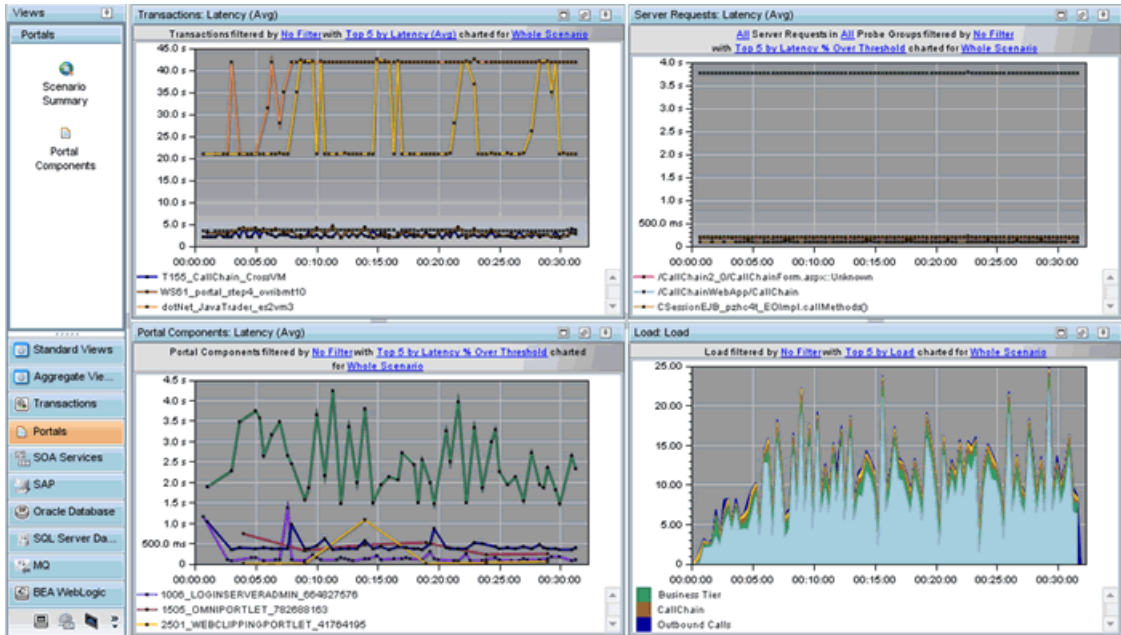
Scenario Summary with Portlets View Description

This Scenario Summary with Portlets view is displayed when you drill down for Diagnostics details, while running a scenario in HP LoadRunner or HP Performance Center that has portlets. You will see the Scenario Summary view in the Portals view group when drilling down from LoadRunner or Performance Center.

The Scenario Summary view is a dashboard view that contains a summary of Diagnostics detail data for the scenario you are running. Monitoring versions (charts/graphs) of the following views are displayed:

- Transactions
- Server Requests
- Portal Components
- Load

The following is an example of the Scenario Summary with Portlets view.



<p>To access</p>	<p>You access the Scenario Summary with Portlets view when drilling down from HP LoadRunner or HP Performance Center.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Integration Between Diagnostics and LoadRunner" on page 1122 ➤ "Integration Between Diagnostics and Performance Center" on page 1124

User interface elements are described below:

UI Elements	Description
<p>Transactions Graph</p>	<p>A monitoring version of the Synthetic Transactions view.</p> <p>By default, the graph displays the top five transactions with the highest average latency, charted for the whole scenario. You can filter the data by Transaction Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Synthetic Transactions view.</p> <p>For information on the Synthetic Transactions view see "Synthetic Transactions View Description" on page 607.</p>
<p>Server Requests Graph</p>	<p>A monitoring version of the Server Requests view.</p> <p>By default, the graph displays the top five server requests with the highest percentage latency over threshold, charted for the whole scenario. You can filter the data by server requests and probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency percent over threshold.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Requests view.</p> <p>For information on the Server Requests view see "Server Requests View" on page 379.</p>

UI Elements	Description
Portal Components Graph	<p>A monitoring version of the Portal Components view.</p> <p>By default, the graph displays the top five portal components with the highest average latency, charted for the whole scenario. You can filter the data by Application Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Portal Components view.</p> <p>For information on the Portal Components view see "Portal Components View Description" on page 636.</p>
Load Graph	<p>A monitoring version of the Load view.</p> <p>By default, the graph displays the load for the top five layers, charted for the whole scenario. You can filter the data by Layer Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average load for a layer.</p> <p>If you double-click the graph the pre-defined workflow displays the Load view.</p> <p>For information on the Server Requests view see "Load View Description" on page 511.</p>

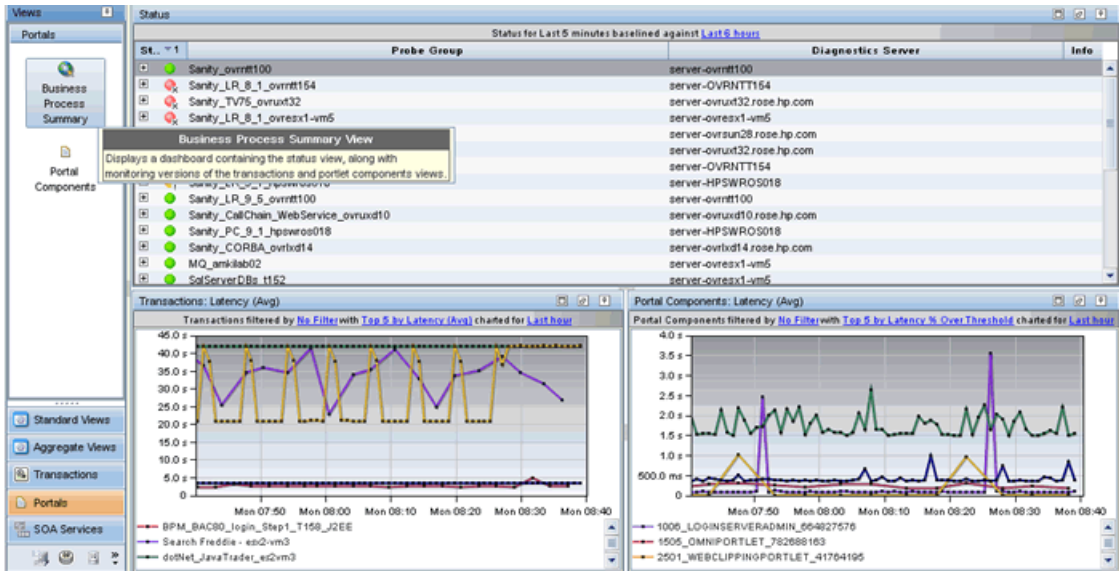
Business Process Summary with Portlets View Description

This Business Process Summary with Portlets view is displayed when you drill down for Diagnostics details from Business Process Monitor. You will see the Business Process Summary with Portlets view in the Portals view group you are drilling down from Business Process Monitor.

The Business Process Summary with Portlets view is a dashboard view that contains a summary of Diagnostics detail data for the business process you select in Business Process Monitor. Monitoring versions (charts/graphs) of the following views are displayed:

- ▶ Status view
- ▶ Transactions
- ▶ Portal Components

The following is an example of the Scenario Summary view.



To access	You access the Business Process Summary with Portlets view when drilling down from Business Process Monitor.
------------------	---

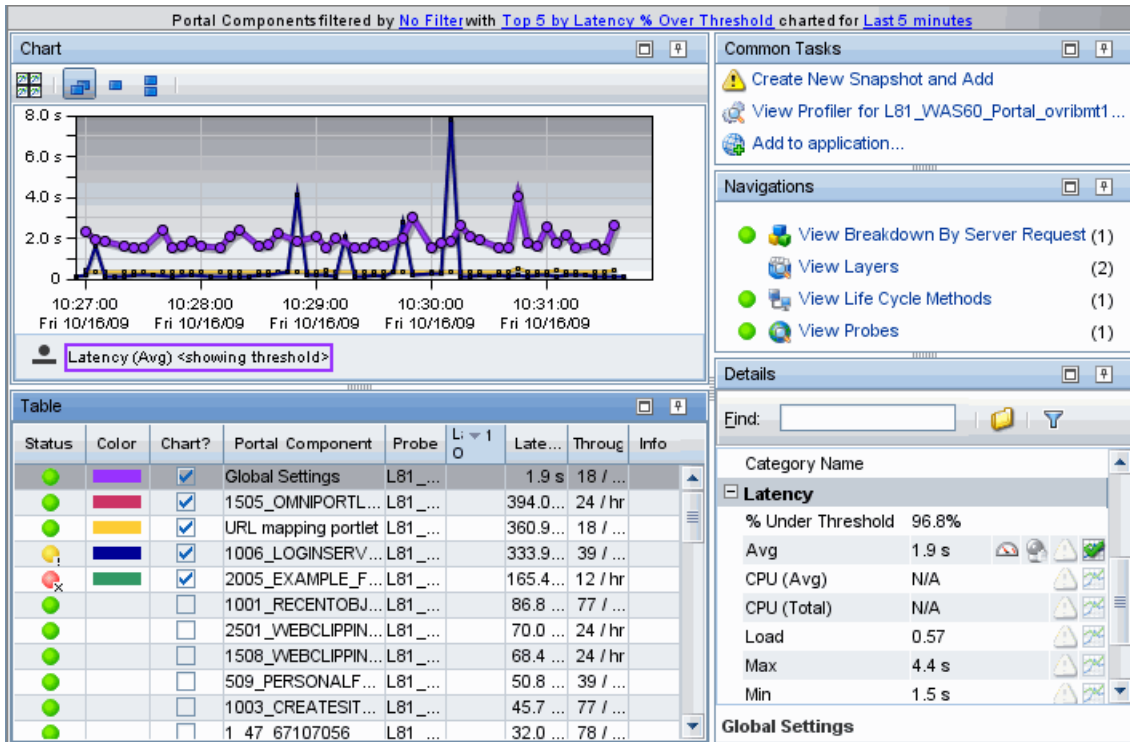
User interface elements are described below:

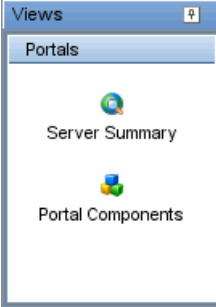
UI Elements	Description
Status View	<p>A monitoring version of the standard Diagnostics Status view.</p> <p>For more information about the Status view, see "Status View Description" on page 520.</p>
Transactions Graph	<p>A monitoring version of the Synthetic Transactions view.</p> <p>By default, the graph displays the top five transactions with the highest average latency, charted for the whole scenario. You can filter the data by Transaction Name.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Synthetic Transactions view.</p> <p>For information on the Synthetic Transactions view see "Synthetic Transactions View Description" on page 607.</p>
Portal Components Graph	<p>A monitoring version of the Portal Components view.</p> <p>By default, the graph displays the top five portal components with the highest average latency.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>If you double-click the graph the pre-defined workflow displays the Portal Components view.</p> <p>For information on the Portal Components view see "Portal Components View Description" on page 636.</p>

Portal Components View Description

In the Portal Components view you see performance metrics for portlets and other portal components (such as pages, books, iViews) that are monitored by Diagnostics.

The following is an example of the Portal Components view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click Portals to open the Portals view group.</p> <p>If the Portals view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select Portals and click OK. The Portals view group is displayed in the View bar.</p> <p>Select the appropriate view from the Portals view group.</p> <p>You can also drill down to the Portal Components view for a particular probe To do this, right click a probe and select View Portal Components. In this case, the Portal Components view displays the specific portlets that are being monitored by that probe.</p>
<p>Important information</p>	<p>The Portal Components view can be used only with a Diagnostics probe from version 7.0 or later.</p>
<p>Relevant tasks</p>	<p>See "How to Use the Portals View Group" on page 621 for information on how to analyze performance using the data in this view.</p>
<p>See also</p>	<p>"Configure Collection of CPU Time Metrics for Portlets" on page 616</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
<p>Chart</p>	<p>Displays a graph of average latency, as a trend line. By default, the five portal components with the highest average latency during the selected time range are charted.</p> <p>When Diagnostics is integrated with Performance Center or LoadRunner, the x-axis of the graph shows the elapsed time since the beginning of the current scenario. In all other cases, the x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Portal Component Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. The name of the probe that measured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>The Portal Component Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

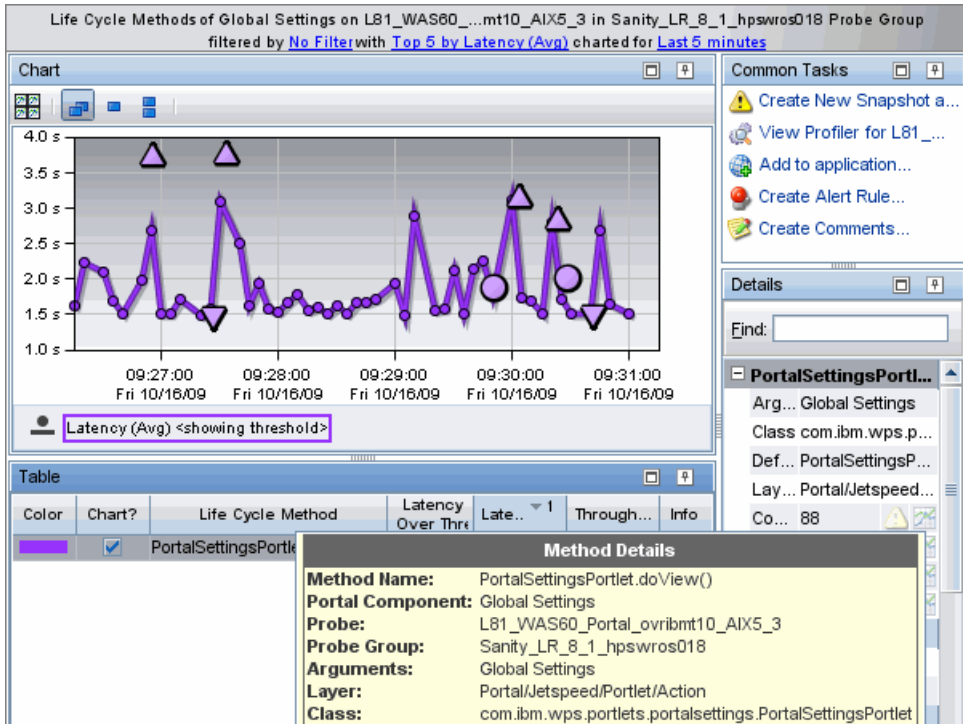
UI Elements	Description
Table	<p>Lists the portal components that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the portal component. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Life Cycle Methods. Other typical drill down navigations include: View Breakdown by Server Request, View Layers, View Probes.</p>
Details Pane	<p>Shows metrics collected on server requests.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p> <p>If CPU (Avg) and CPU (Total) values aren't shown (N/A), then collection of CPU Time metrics is not enabled for the relevant probe or is not configured for portlets or methods. See "Configure Collection of CPU Time Metrics for Portlets" on page 616.</p>

Life Cycle Methods View Description

The Life Cycle Methods view displays the top five life cycle methods with the highest average latency.

The following is an example of the Life Cycle Methods view.



To access	The Life Cycle Methods view cannot be accessed directly from the Standard View bar because lifecycle methods are always displayed in the context of a portlet. You can access the Life Cycle Methods view from the Portal Components view in the Portals view group. Select the portlet and then select View Life Cycle Methods.
Important information	Portlet Life Cycle instance trees are enabled by default. You can turn this off in the dispatcher.properties file with <code>enable.portlet.method.trees=false</code> .
See also	See "Call Profile View" on page 405

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

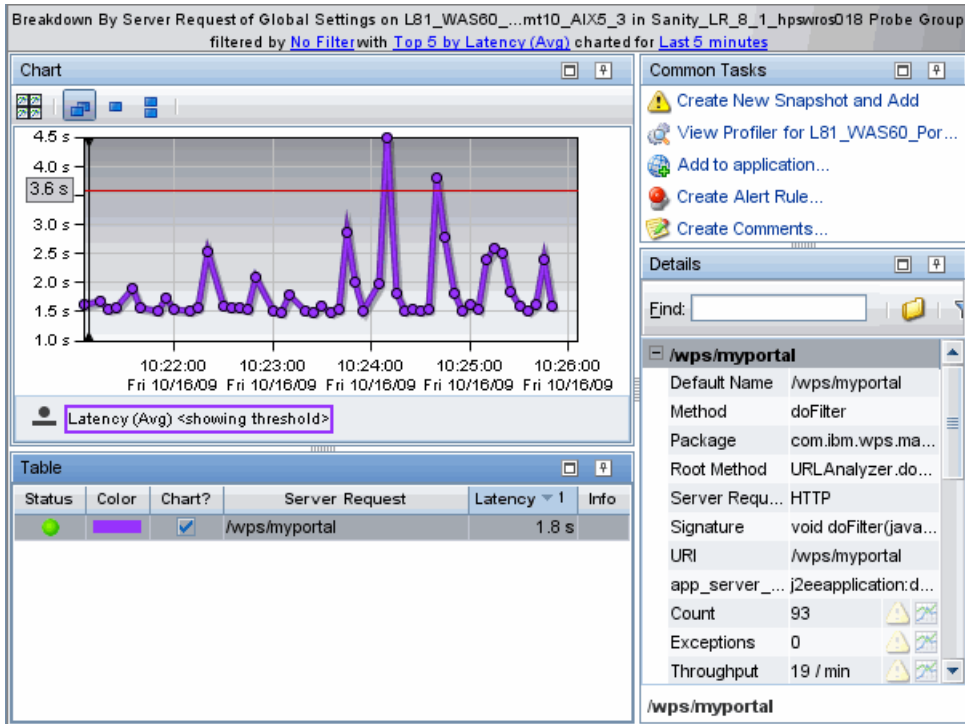
UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five life cycle methods that have the highest average latency during the selected time range are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Method Details tooltip for the trend line on the graph (or the Life Cycle Method column in the table) displays the following detail data:</p> <p>Method name. The name of the life cycle method.</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. The name of the probe that collected data on the method.</p> <p>Probe Group. The name of the probe group.</p> <p>Arguments. The arguments associated with the method.</p> <p>Class. The class associated with the method.</p> <p>Layer. The layer where the method is called.</p> <p>The Method Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created.</p> <p>You can drill down from the instance tree icon to see a call profile. See "Instance Tree Call Profiles for Portlets" on page 618 for more information.</p> <p>And see "Call Profile View" on page 405 for how to use the Call Profile view.</p>

UI Elements	Description
Table	<p>Lists the life cycle methods that pertain to the context selected. The metrics reported in the table are aggregated based on the time period select in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Life Cycle Method. The name of the method.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	Common tasks and actions relevant to the life cycle methods displayed in this view.
Details Pane	<p>Shows metrics collected on life cycle methods.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Breakdown by Server Requests

The Breakdown by Server Requests view displays information on the Server Requests that contributed to the portlet's latency performance.

The following is an example of the Breakdown by Server Request view.



To access

The Breakdown by Server Requests view cannot be accessed directly from the Standard View bar because the breakdown is always displayed in the context of a portlet. You can access the Breakdown by Server Requests view from the Portal Components view in the Portals view group. Select the portlet and then select View Breakdown by Server Request.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five portal components that have the highest average latency during the selected time range are charted.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Server Request Details tooltip for the trend line on the graph (or the Server Requests column in the table) displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL or, in the case of an RMI call, a class and method representing the name of the server request itself.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Method Signature. The signature of the root method.</p> <p>The Server Request Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the server requests contributing to the selected portal components latency during the time range.</p> <p>The metrics for the server requests displayed in this view do not represent the whole server request. Instead, they represent the part of the server request that worked on processing the relevant portlet.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the server request. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Server Request. Name of the server request that initiated this portion of the portal component statistics.</p> <p>Latency. The average response time across all instances.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the server requests displayed in this view.
Details Pane	<p>The metrics for the server requests displayed in this view do not represent the whole server request. Instead, they represent the part of the server request that worked on processing the relevant portlet.</p> <p>Under the Latency category in the Details pane you can view the contribution of the selected server request to the average latency of the portlet.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

28

SOA Services Views

This chapter describes the specialized views for SOA Services.

This chapter includes:

Concepts

- ▶ About SOA Services Monitoring on page 653
- ▶ Supported Web Service Platforms on page 654
- ▶ SOA Services Overview on page 655
- ▶ Instance Tree Call Profiles for SOA Services on page 656
- ▶ Cross VM and SOAP Faults for Services on page 659
- ▶ About Consumer IDs on page 663
- ▶ Monitoring REST Style Services on page 665
- ▶ About the Enterprise Service Bus Environment on page 666

Tasks

- ▶ How to Specify a Name for the Application Server Instance on page 667
- ▶ How to Change the Display Name of a Web Service on page 668
- ▶ How to Monitor a BEA AquaLogic Service Bus Environment on page 669
- ▶ How to Monitor the SOA Policy Enforcer Broker on page 675
- ▶ How to Monitor an IBM WebSphere Enterprise Service Bus Environment on page 680
- ▶ How to Monitor TIBCO ActiveMatrix Environment on page 684
- ▶ How to View BEA WLI Business Process Tracing Data on page 690

Reference

- ▶ SOA Services View Group User Interface on page 700

Concepts

About SOA Services Monitoring

A SOA service (for example, Web service or REST service) is a software component that provides functionality via a set of operations accessible over the network using various protocols.

For example, a currency conversion Web service offers a variety of Web service operations, such as returning the exchange rate for a specific currency or returning the currency symbol based on the specified locale.

Diagnostics monitors outbound Web service calls made from within your monitored environment, and inbound Web service calls received and processed in your monitored environment.

For example, a consumer requests an exchange rate from a currency conversion Web service. From the consumer's point of view, that request is defined as an outbound operation. The service provider receives the request, processes it, and sends a response (the exchange rate). From the service provider's point of view, the Web service request received from the consumer is defined as an inbound operation.

Outbound and inbound web service calls that use JMS as the transport are monitored as Web services in Diagnostics. In this case the JMS target (queue name or topic_name) is used as the target for the web service calls. In the Diagnostics Outbound service calls view and in the service topology the remote target is displayed as queue_name/web_service_name or topic_name/web_service_name.

Diagnostics monitors the following:

- ▶ SOAP over HTTP/HTTPS operations
- ▶ SOAP over JMS operations
- ▶ .NET Windows Communication Foundation (WCF) services are monitored as web services

In addition, the Diagnostics Java Agent monitors XML over HTTP/HTTPS operations when REST style services are configured to be monitored as Web services in Diagnostics. The .NET Agent does not support monitoring REST style services.

Key features for Web services monitoring provided by Diagnostics include:

- ▶ Creation of samples of Web services data that are sent to Business Service Management when the two products are integrated. See the Chapter 39, "Diagnostics Integration with Other HP Software Products" for details.
- ▶ The collection of performance data based on Web service operation and aggregation of the data into named services.
- ▶ Tracking of Web service consumers and breakdown views of consumer IDs by Web service operations and aggregated by Web Service (across all operations).
- ▶ Drill down to the call profiles for key instances includes details on SOAP faults and SOAP payloads captured on failures.
- ▶ Service Topology view providing information about the connection between Web services.
- ▶ Views for outbound call to Web services and outbound calls to Web service operations.

Supported Web Service Platforms

Diagnostics supports Web services tracking for the following:

- ▶ BEA WebLogic
- ▶ IBM WebSphere
- ▶ JBOSS
- ▶ Microsoft .NET and .NET Windows Communication Foundation (WCF)
- ▶ SAP NetWeaver

- Apache Tomcat
- AquaLogic Service Bus (ALSB)
- IBM WebSphere Enterprise Service Bus
- TIBCO ActiveMatrix Enterprise Service Bus and BusinessWorks
- HP SOA Policy Enforcer broker can be monitored by the Diagnostics Java Probe so that proxy services are seen as Web services and Diagnostics can monitor the operations on these services. See "How to Monitor the SOA Policy Enforcer Broker" on page 675.

For more information on supported versions and supported features for each version see the Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp or contact HP Customer Support.

SOA Services Overview

The **SOA Services** view group shows latency and other performance data for service oriented architectures.

Diagnostics automatically detects server requests that are Web services. Diagnostics aggregates Web service operations to Web services (by aggregating all Web service operations for a particular named Web service) to create a higher level aggregation of service performance. This data is used to provide a service topology showing which services connect to other services.

For details on the user interface, see "SOA Services View Group User Interface" on page 700.

The SOA Services view group includes the following views:

- **Service Summary.** A dashboard that displays the graphs from each of the following views.
- **Services.** Shows statistics for the services in your application. Services are made up of server requests that are operations aggregated into a named service.
- **Operations.** Displays server requests that are Web service operations.

- ▶ **Service Topology.** Displays a topology view of monitored Web services and service connections.
- ▶ **Services by Consumer ID.** Displays a breakdown by consumer for the Services in your application.
- ▶ **Operations by Consumer ID.** Displays a breakdown by consumer for the server requests that are Web service operations.
- ▶ **Outbound Service Calls.** Shows statistics for outbound calls to Web services from your application.
- ▶ **Outbound Operations Calls.** Shows statistics for outbound calls to Web service operations from your application.

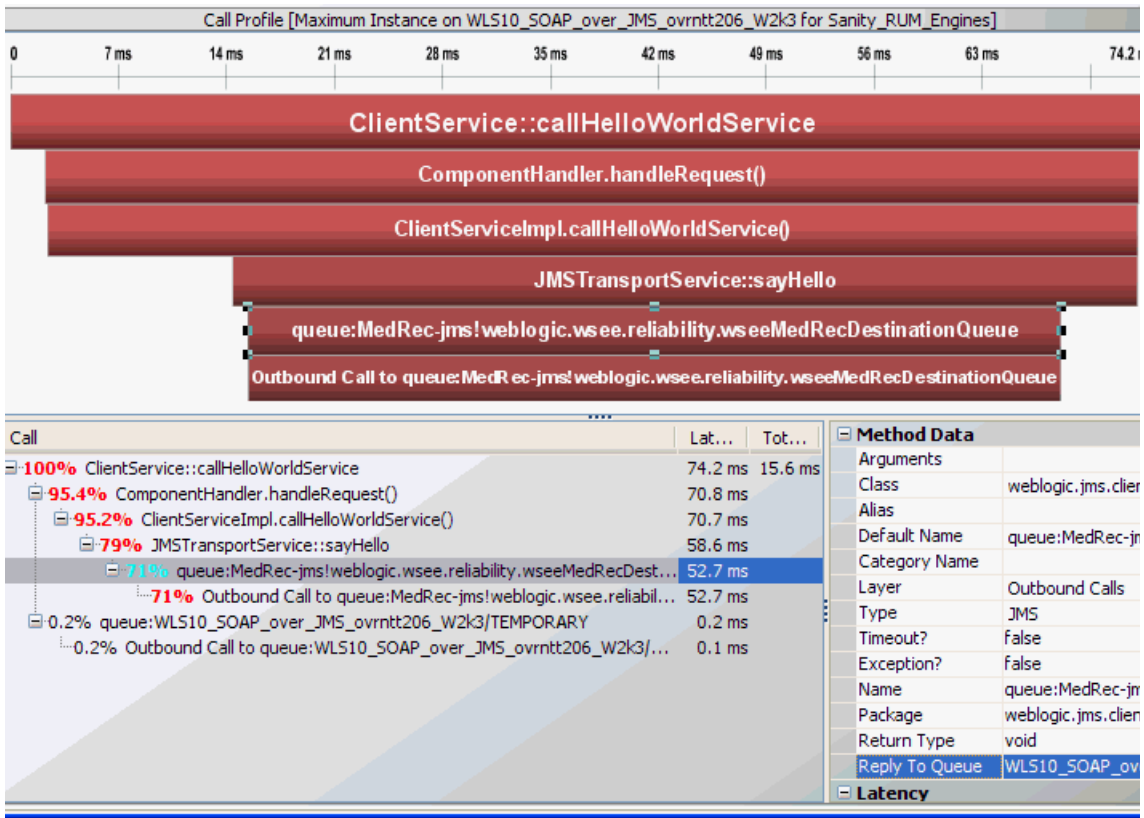
The Web service name and/or operation may appear as **Unknown**.

Instance Tree Call Profiles for SOA Services

Instance trees for the most interesting instances of services and operations are saved. Instance trees for the minimum, maximum and average instances are saved along with instance trees for exceptions, SOAP faults and cross VM correlations. You can select the appropriate marker to drill down to the Call Profile view for each of these types of instances.

In the call profile you can then compare a 'good' performing call profile with a 'bad' performing call profile. In addition, if you see a pattern in the convergence of the min/max instances with the running trend line you can create a snapshot and look for other events that follow a similar pattern.

An example of a Call Profile view for a service is shown below.



For outbound JMS calls, you will see the **Reply To Queue** name in the method data of the call profile (if the Reply To queue name is present on the outbound call). If the Reply To queue name is a temporary queue it will be renamed accordingly. You can use the Reply To queue name to help differentiate between producer and consumer on a JMS instance tree.

The web service call profile only shows the call tree from that method call down. It just shows that unit of work. To see the call tree for the whole server request, which may include other calls, other portlets, you would drill down to the portlet call profile from the Server Requests view. To access the Call Profile view from the Server Requests view, see "How to Use the Server Requests View" on page 385.

For more information about the Call Profile view, see "Call Profile View" on page 405

About Instance Tree Markers

Instance tree markers are displayed for the selected web services and operations. The markers are placed on the graph to indicate the total latency and the end time of the method for which the instance tree was created. For a detailed explanation of instance trees, and how Diagnostics captures them, see Chapter 9, "Working with Instance Trees."

Four instance tree marker icons are used to represent the different types of instance trees that are created for a web service request:



- ▶ **Maximum Instance Tree.** The maximum instance tree for a life cycle method represents an instance tree with the largest latency for the time period. This tree depicts one of the worst-performing invocations of the life cycle method. This is one tool used for diagnosing the root cause of poor application performance.



- ▶ **Minimum Instance Tree.** The minimum instance tree for a life cycle method represents an instance tree with the smallest latency for the time period. This tree depicts one of the best-performing invocations. It can be useful for comparison to instances that perform poorly.



- ▶ **Average Instance Tree.** The average instance tree for a life cycle method represents an instance tree that represents the average latency for the time period. This tree depicts a typical invocation. It can be useful for comparison to instances that perform poorly.



- ▶ **Cross VM Instance Tree.** The cross VM instance tree for a server request represents an instance tree that includes a call to another server request by using a technology such as RMI or Web services.

The following icons are used to represent exceptions and faults. Additional information is collected by the probe when exceptions or SOAP faults occur. You can drill down to the call profile graph and view exception or fault details for a method, including the stack trace.

See "Types of SOAP Fault Markers" on page 271 for details on the icons used to represent different types of faults.

See "Configure Collection of SOAP Faults and Payloads" on page 409 for details on how to configure SOAP fault and payload collection.



- **Exception.** The exclamation mark indicates where an exception occurred. When you click the icon, a call profile opens. This call profile has the methods that threw the exceptions marked with a yellow dashed outline and an **Exceptions** tab providing details on the exception. A stack trace for the exception is included with the details.



- **SOAP Faults.** The icon indicates where an soap fault occurred. When you click the icon, a call profile opens. This call profile has a SOAP Faults tab providing details on the fault, including the stack trace.

Cross VM and SOAP Faults for Services

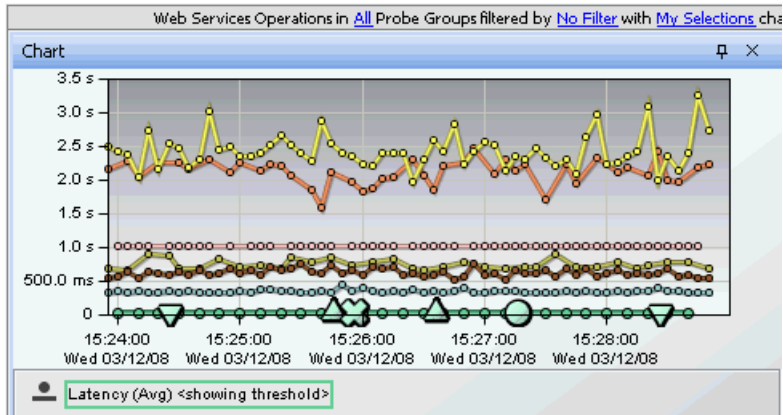
Instance trees for SOAP faults and cross VM correlations are saved for services and operations.

Cross VM Instances for Services

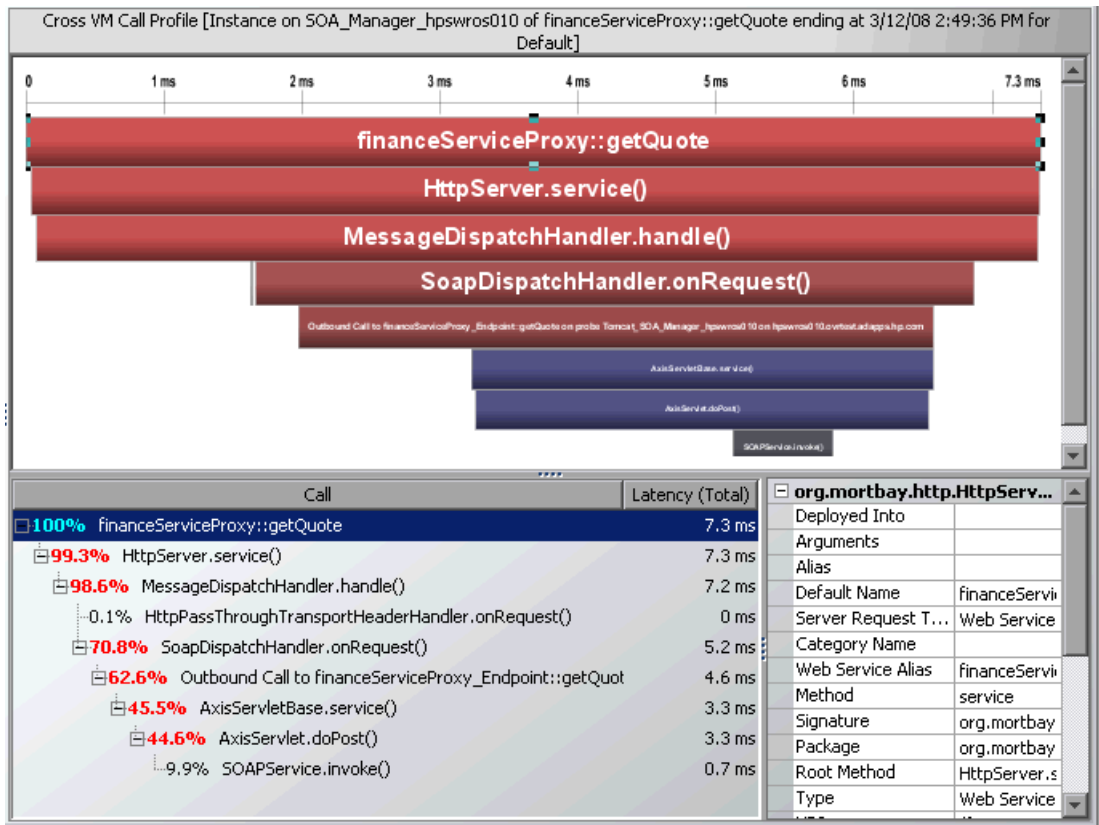
If both the outbound and inbound Web service calls for a particular Web service are monitored in your environment, you can view the correlation between outbound and inbound calls in an cross VM instance tree in the Call Profile view.



Click the X on the graph in the Operations view containing the outbound Web service call.



The Call Profile view opens, displaying a correlation instance tree. In the following example the call displayed at the top of the diagram and table contains an outbound Web service call and the inbound call for the same Web service is displayed in blue. Note that instance trees are not shown for the Outbound Operation Calls view or Outbound Service Calls views.



SOAP Faults for Services

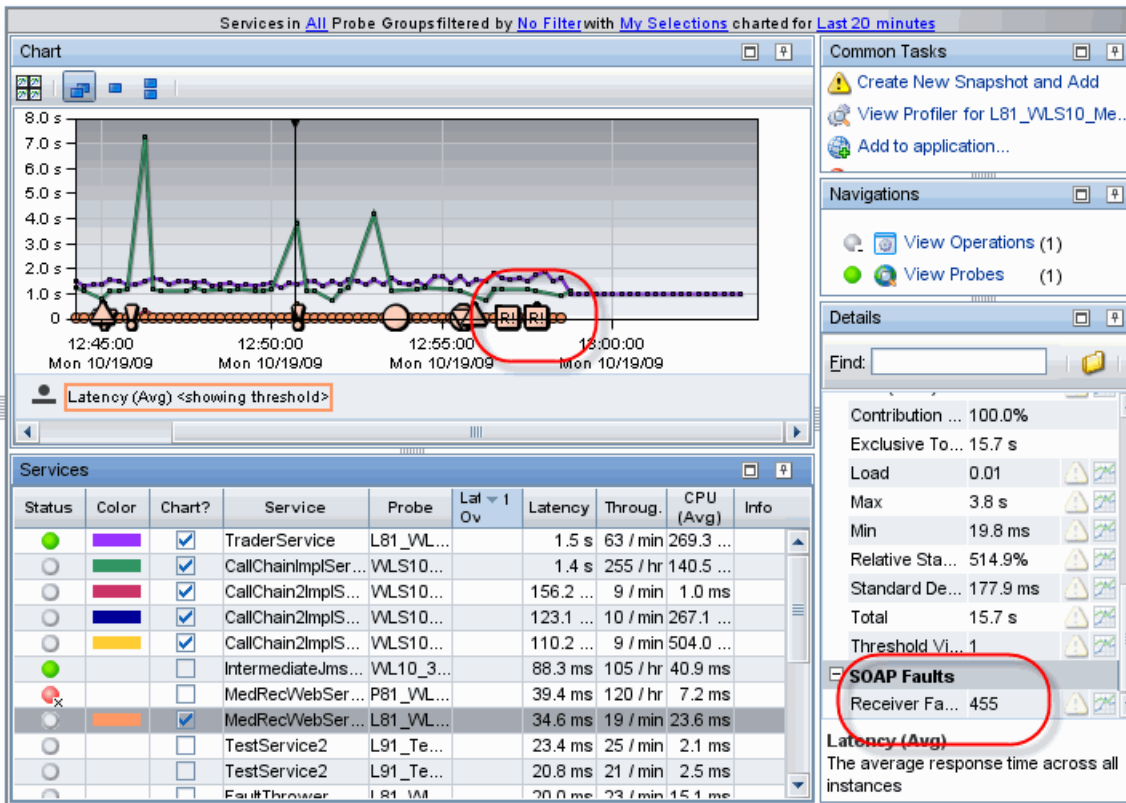
SOAP is an XML-based messaging protocol used in the exchange of Web services over a network. A SOAP fault is used to carry error information within a SOAP message.

Diagnostics captures SOAP faults and saves instance trees for the faults. By default, up to two SOAP faults per time period are captured.

See the *HP Diagnostics Installation and Configuration Guide* for details on configuring the probes to capture SOAP fault data.

Note: For the Java probe, SOAP faults are captured on both SOAP over HTTP and SOAP over JMS web services since the mechanism used to capture these faults is transport independent.

Diagnostics classifies SOAP faults into different types, based on their SOAP fault codes. Captured SOAP faults are represented in the SOA Services views by a unique icon on the graph. You can also see the number of SOAP faults in the details pane.



Each SOAP fault icon represents a different type of SOAP fault. When you click one of these icons, Diagnostics opens a call profile window that includes detailed information about the SOAP fault.

About Consumer IDs

The consumer represents calls to a Web service coming from a non-instrumented source (not monitored by a probe). These calls are grouped together in the service topology under the consumer node. You can configure consumer IDs to identify specific consumers.

Caution: For SOA customers, Business Service Management 7.50 or later is required to take advantage of Diagnostics Web services enhancements (such as configurable Consumer ID) in the Business Service Management integration.

Consumer ID is an arbitrary string that can be configured from any of the following:

- IP address or IP range.
- HTTP Header.
- SOAP request Headers, Body or Envelope. SOAP payload is the entire SOAP envelope.
- For Java probes, JMS queue name (or topic name) and JMS message header or JMS message property for SOAP over JMS web services.

IP address is used as the default consumer ID for SOAP over HTTP/S web services. For Java probes, inbound queue name (or topic name) is used as the default consumer ID for SOAP over JMS web services.

You can configure what you want in the Consumer ID field using probe configuration files. Note though that there can be a performance impact depending on what you configure for consumer ID.

You can also configure the number of consumer IDs that are monitored and displayed per probe. The default is five consumer IDs per probe. Assume for example that the probe was configured to display five unique consumer IDs. In this case, if there are more than five consumer IDs, the first five are displayed on separate rows in the graph entity table. The remaining consumer IDs are aggregated into a single row in the table with the consumer ID label **other**.

Configuration of consumer IDs is done on the .NET Probe and the Java Probe. See the *HP Diagnostics Installation and Configuration Guide* for details.

Note: Diagnostics creates samples for each consumer ID that calls a Web service. These samples are sent to Business Service Management when the two products are integrated. See the Chapter 39, "Diagnostics Integration with Other HP Software Products")

Increasing the number of consumer IDs captured by Diagnostics will therefore increase the amount of data sent to, and processed by, Business Service Management. This will also increase the amount of CPU/Memory/Disk space used by the Diagnostics Servers and probes.

You can stop Diagnostics creating these samples by going to the `<diag_server_install_dir>\etc\server.properties` file on the mediating server and changing the value of the following line to **false**:

```
bac.webservice.create.samples = true.
```

Caution: For Java probe, defining consumer ID based on SOAP header, envelope or body requires the Diagnostics SOAP message handler. For some application servers, special instrumentation is provided in Diagnostics to automatically load the Diagnostics SOAP message handler.

However, some manual configuration is required for WebSphere 5.1 JAX-RPC and Oracle 10g JAX-RPC, see the *HP Diagnostics Installation and Configuration Guide* for details on configuring the SOAP message handler as well as configuring the probes to capture SOAP fault data.

In addition, the Diagnostics SOAP message handler is not available for all application servers nor is custom instrumentation available to capture SOAP faults or consumer IDs from SOAP payloads, and therefore this feature is not available on all versions of all application servers. For the most recent information on Diagnostics SOAP message handler support refer to the Diagnostics Product Availability Matrix at http://support.openview.hp.com/sc/support_matrices.jsp.

Monitoring REST Style Services

If you have Web services that are implemented with the REST style technology these will only be monitored by the Java Probe as server requests unless you configure Diagnostics to monitor them as Web services. For a Java Probe (not supported for the .NET probe) you define which of these REST services (HTTP server requests) you want monitored as Web services. This is done using the **rest.properties** file. See the *HP Diagnostics Installation and Configuration Guide* for details and examples.

Note: Unlike SOAP based Web services which are not latency trimmed, REST services are latency trimmed by default. This can be changed by setting **minimum.fragment.latency = 0ms** in the **dispatcher.properties** file for the probe.

About the Enterprise Service Bus Environment

In some environments an Enterprise Service Bus (ESB) lies between business applications and enables communication among them. In an enterprise architecture making use of an ESB, a Web service application will communicate via the bus, which acts as a message broker between applications. The primary advantage of such an approach is that it reduces the number of point-to-point connections required to allow Web service applications to communicate.

Enterprise Service Bus software is open standards-based distributed synchronous or asynchronous messaging middleware that provides secure interoperability between enterprise applications via XML, Web services interfaces and standardized rules-based routing of documents. It can handle the incoming Web service calls and process them for validation, extraction, branching and routing through different sets of internal proxy and external business services.

Diagnostics allows you to monitor Web services in an Enterprise Service Bus environment by providing Java Probe instrumentation for the following:

- ▶ BEA AquaLogic Service Bus
- ▶ HP SOA Policy Enforcer broker.
- ▶ IBM WebSphere Enterprise Service Bus
- ▶ TIBCO ActiveMatrix Service Bus
- ▶ TIBCO ActiveMatrix BusinessWorks

Tasks

How to Specify a Name for the Application Server Instance

Diagnostics creates samples with Web services performance data. The samples are sent to Business Service Management when the two products are integrated. To differentiate between application servers when there are multiple application servers on the same host (potentially hosting the same Web service), Business Service Management uses the **end_point identifier**. By default, the endpoint in the samples from Diagnostics is the probe name. However, you can set it to a more meaningful name in the **Probes** view of Diagnostics.

This task includes the following steps:

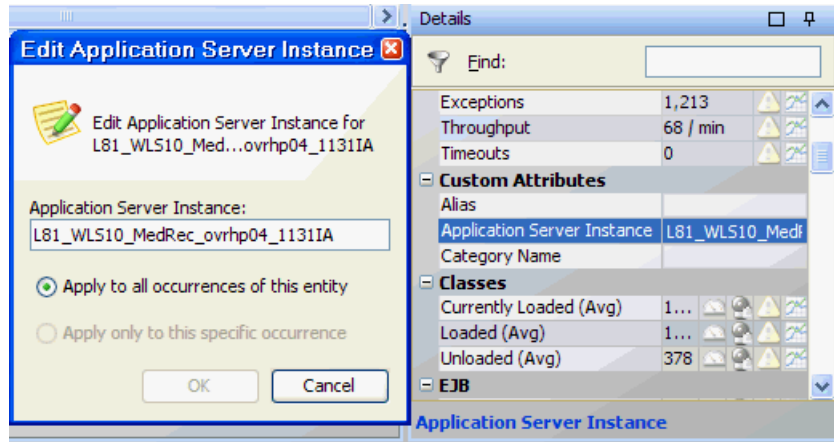
- "Navigate to the Probes View" on page 667
- "Enter an Application Server Instance Name" on page 667

1 Navigate to the Probes View

Navigate to the Probes view and select a probe that is monitoring your Web service.

2 Enter an Application Server Instance Name

Select Application Server Instance in the details pane. Enter the name you want to use for the application server instance. This is the endpoint that is passed to Business Service Management.

Example:

How to Change the Display Name of a Web Service

Diagnostics and Business Service Management use different ways of identifying the names of Web services and operations. As a result, there may be certain cases where Diagnostics and Business Service Management identify different names for the same Web service and operation.

In such cases, you can change the Web Service and operations names as displayed in Diagnostics to match the names used in Business Service Management.

This task includes the following steps:

- "Select a Service or Operation" on page 668
- "Use the Alias Field to Enter a Name" on page 669

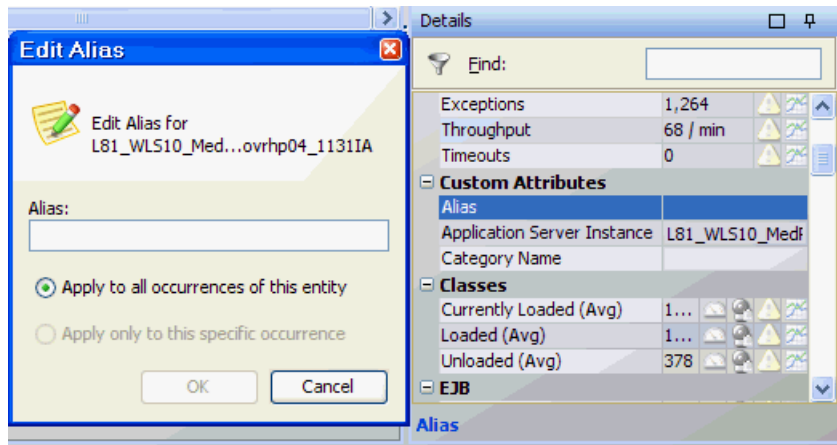
1 Select a Service or Operation

In the SOA Services views, select a service or operation to change the name.

2 Use the Alias Field to Enter a Name

Use the **Alias** field in the Details pane under Custom Attributes to change the service name or the service::operation name.

Example:



How to Monitor a BEA AquaLogic Service Bus Environment

The Java Probe includes instrumentation points for capturing the inbound proxy services and the outbound call to HTTP and JMS business services running in a BEA AquaLogic Service Bus (ALSB) environment. AquaLogic Service Bus is an enterprise service bus (ESB) for service-oriented integration, managing service interactions, and brokering messages across several heterogeneous IT environments.

This task includes the following steps:

- "AquaLogic Version Support" on page 670
- "ALSB Mechanism for Handling Server Requests" on page 670
- "Request and Response Pipelines" on page 672

- ▶ "Diagnostics Server Requests View" on page 672
- ▶ "Asynchronous or Synchronous Service Requests" on page 673
- ▶ "Cross VM Correlation" on page 673
- ▶ "Latency By Layers" on page 674

AquaLogic Version Support

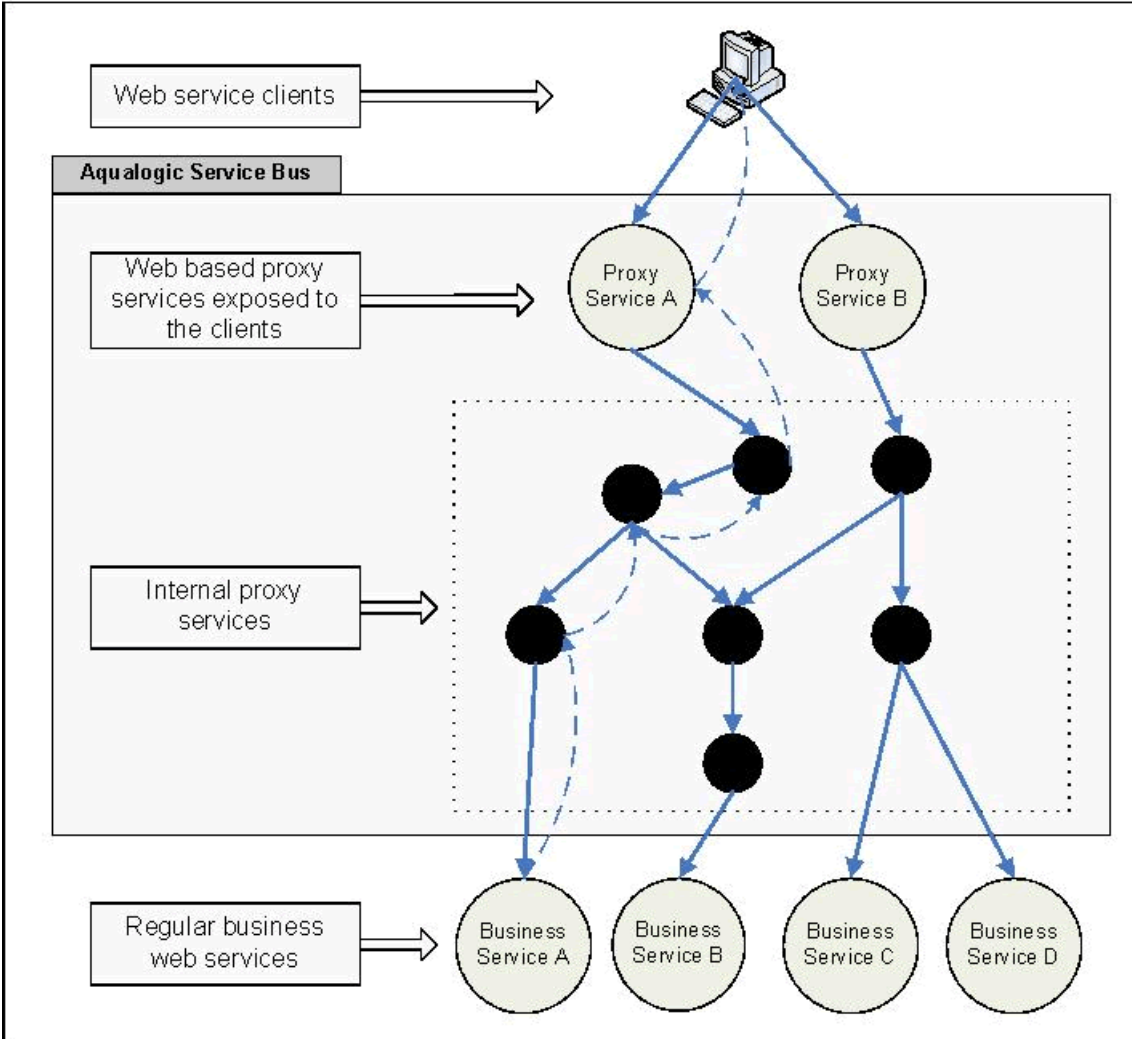
ALSB version 2.6 is supported with the Diagnostics Probe for Java.

ALSB Mechanism for Handling Server Requests

An incoming service request can flow through a series of proxy and business services before the response is returned to the user (client). The proxy and business services in ALSB can be based on several protocols (HTTP, local, JMS). The services based on local protocol are usually internal services which are not exposed to the web. These local services, therefore, are not usually defined using a WSDL and are not supported in Diagnostics.

Example:

The diagram below shows a general ALSB mechanism for handling server requests.

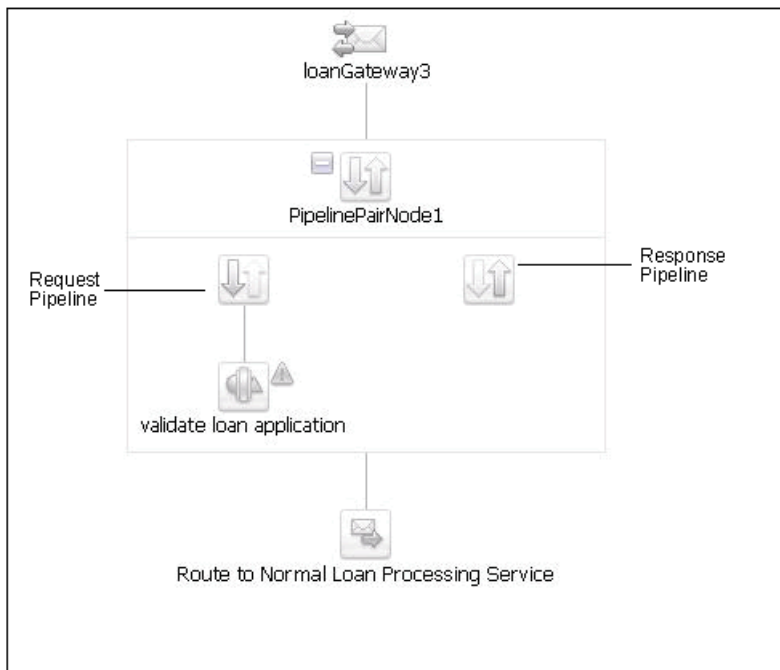


Request and Response Pipelines

In AquaLogic service bus, the incoming client request goes through a 'request pipeline' before it is eventually routed to a business service. The response from the business service then goes through a 'response pipeline'. The request and response pipelines work on different threads.

Example:

The request and response pipelines work on different threads as shown in the example ALSB message flow view below.

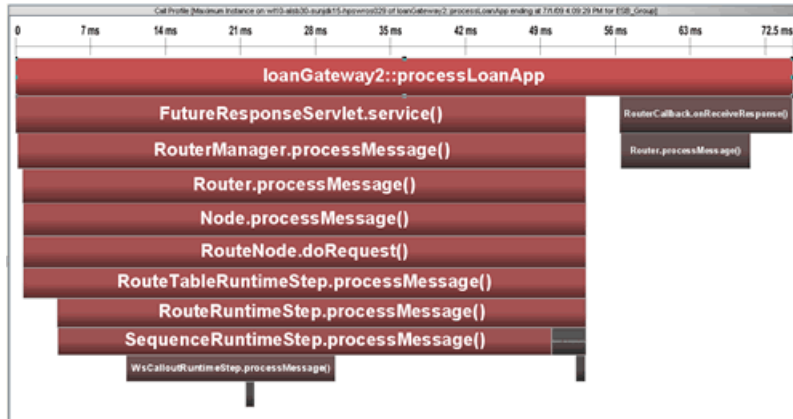


Diagnostics Server Requests View

In Diagnostics, the activities performed for both Request and Response Pipelines will show as one Server Request.

Example:

The proxy service request (for example, `loanGateway2::processLoanApp`) includes latency for processing in the request pipeline, `FutureResponseServlet.service()` and in the response pipeline, `RouterCallback.onReceiveResponse()` as shown below.

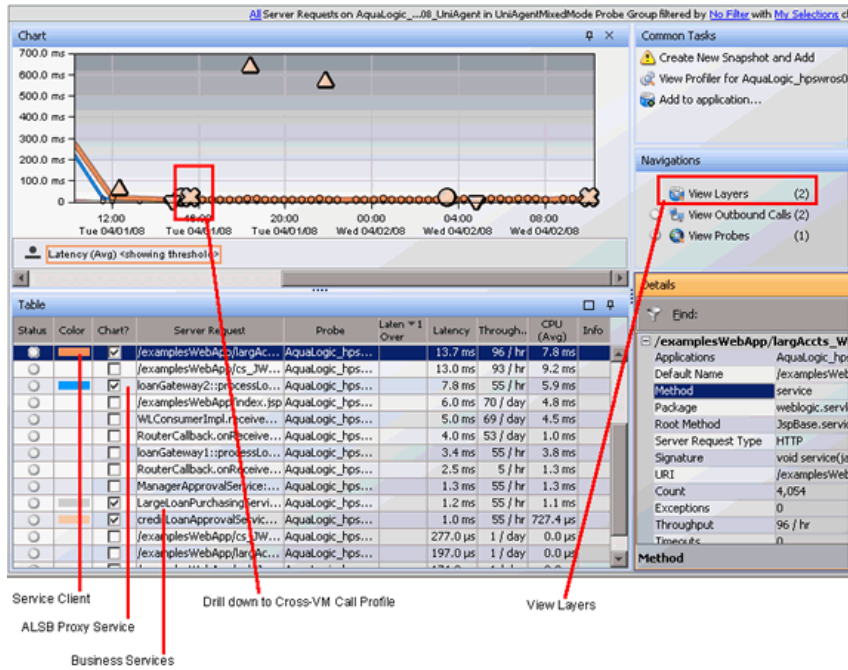
**Asynchronous or Synchronous Service Requests**

In AquaLogic, the route node can route the request to the business service either synchronously or asynchronously. This can be configured using the 'Quality of Service' attribute of the routing option. By default, the business services are invoked asynchronously.

Cross VM Correlation

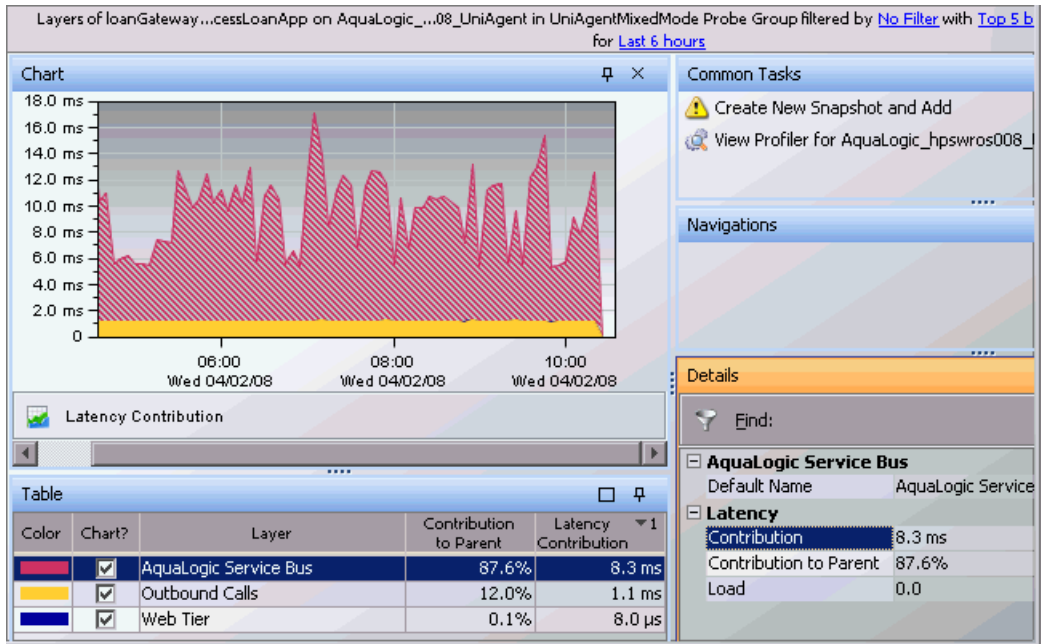
The cross-VM correlation for multiple levels of outbound service invocations is shown at the top level server request. It does not include the response from the business service.

Example:



Latency By Layers

From the Server Requests view you can select View Layers to see the latency by layer.

Example:

How to Monitor the SOA Policy Enforcer Broker

The Diagnostics Probe for Java integrates with the HP SOA Policy Enforcer broker and can be used to monitor proxy web services deployed in the policy enforcer environment. Please note that for SOA Broker 2.5 you would need to manually set up the integration with Diagnostics (see the `auto_detect.points` file).

This task includes the following steps:

- "Java Probe Instrumentation for SOA Policy Enforcer Broker" on page 676
- "Web Service Request/Response Monitoring" on page 676
- "Diagnostics Server Request View" on page 677
- "Call Profile View" on page 678

- "View Layers" on page 679

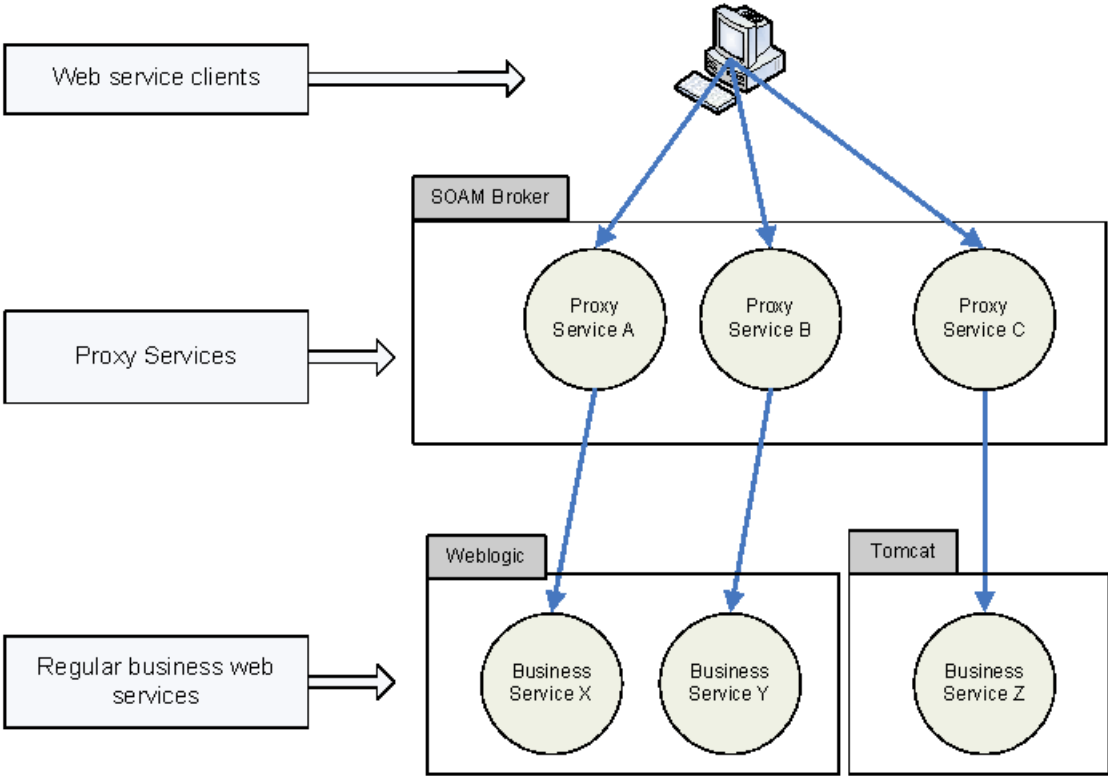
Java Probe Instrumentation for SOA Policy Enforcer Broker

The Java Probe includes instrumentation for SOA Policy Enforcer broker inbound and outbound services as well as all the service handlers.

Web Service Request/Response Monitoring

Diagnostics monitors Web services request/response from the client to SOA Policy Enforcer broker through proxy services and then to the business service and the return to the client.

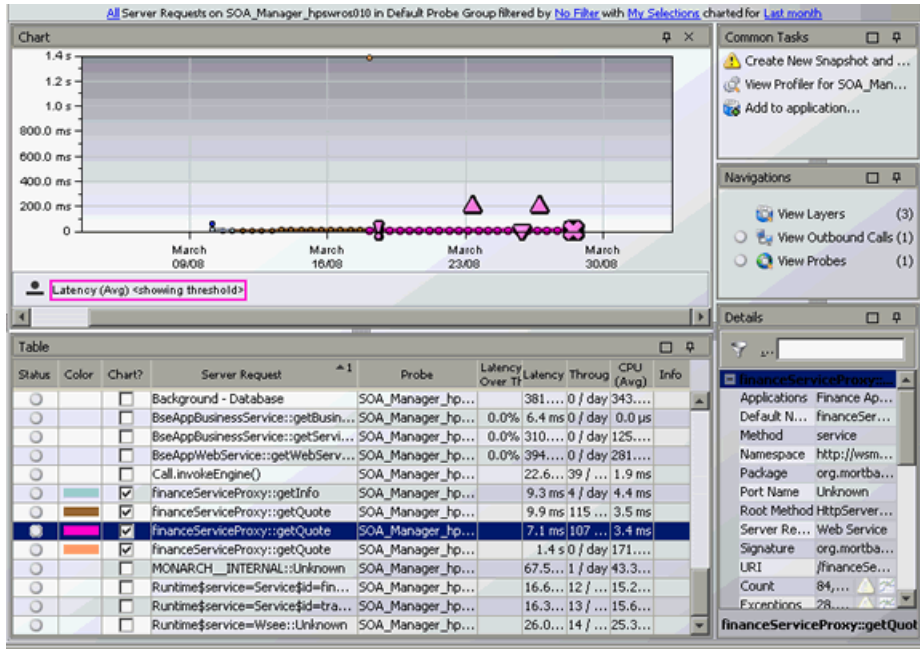
Example:



Diagnostics Server Request View

You can see the server request data from your SOA Policy Enforcer environment in the Diagnostics UI.

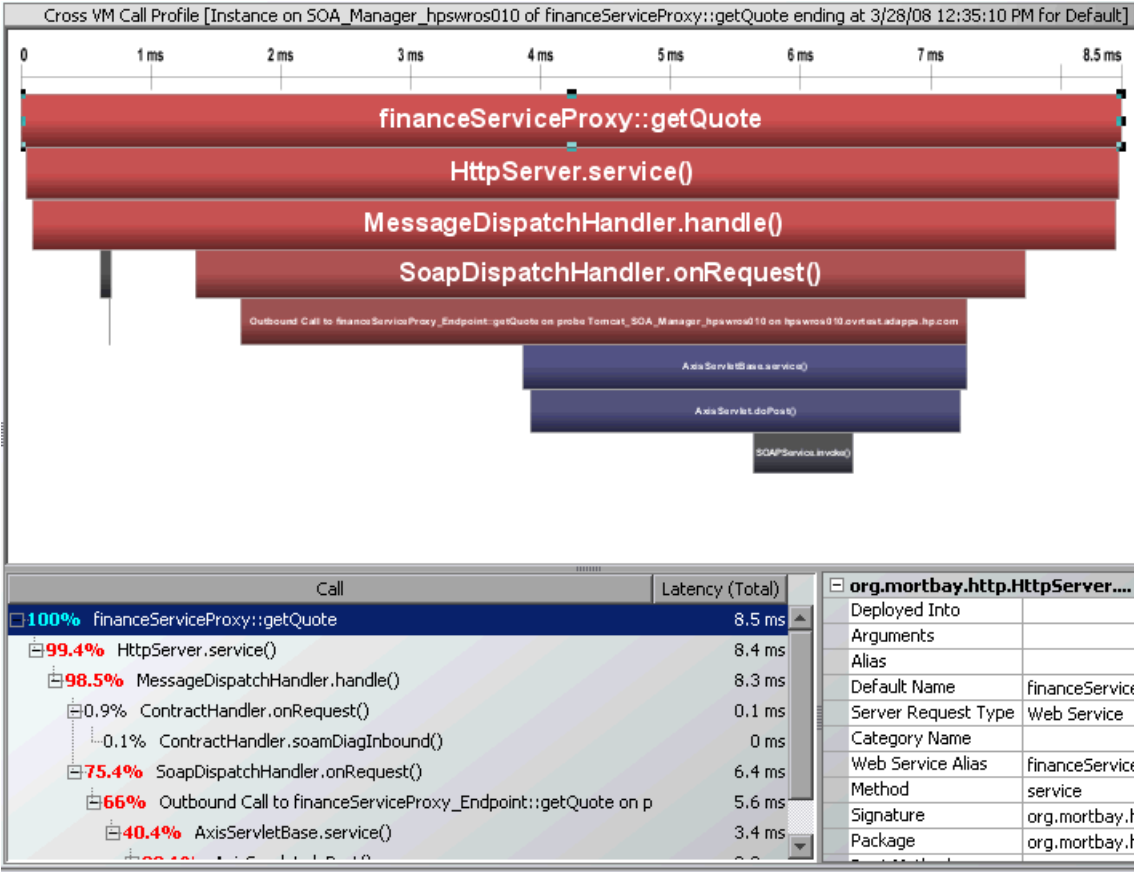
Example:



Call Profile View

From the Server Requests view you can drill down to see detailed call profiles for minimum/maximum instances, cross-VM instances, exceptions and SOAP faults.

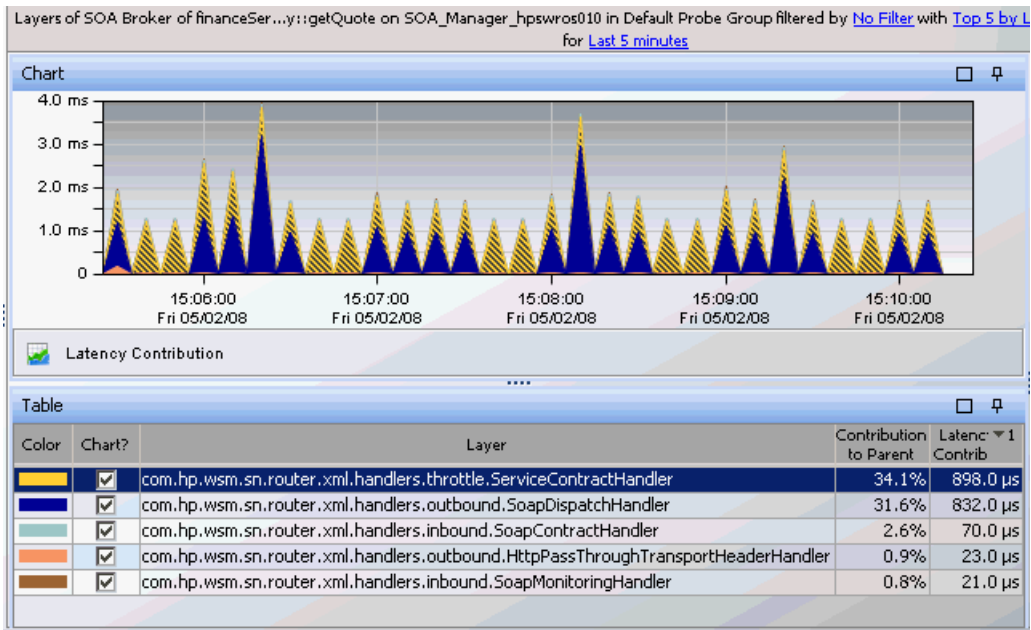
Example:



View Layers

From the SOA Services Operations view or the Server Requests view you can select View Layers and then drill down to the sub-layers for the SOA Policy Enforcer broker.

Example:



How to Monitor an IBM WebSphere Enterprise Service Bus Environment

IBM WebSphere Enterprise Service Bus (ESB) provides an ESB for IT environments built on open standards, SOA, messaging and Web services technologies of WebSphere Application Server.

WebSphere ESB manages the flow of messages between Service Component Architecture (SCA)-described interaction endpoints.

An ESB lets you reuse assets in a flexible way, by passing service interactions through logic called mediations. Mediations operate on messages in transit between requesters and providers; they allow facilities such as message filtering, protocol conversion, database logging, and dynamic routing.

In an SCA-based solution mediation modules are a type of SCA module. Usually mediation modules contain a specific type of SCA component called a mediation flow component. Mediation flow components define mediation flows. A mediation module can contain only one mediation flow component.

The mediation modules perform a special role, and therefore have slightly different characteristics from other components that operate at the business level.

Mediation components operate on messages exchanged between service endpoints. In contrast with regular business application components, they are concerned with the flow of the messages through the infrastructure and not just with the business content of the messages. Rather than performing business functions, they perform routing, transformation, and logging operations on the messages. Also mediation modules within the ESB handle mismatches between requesters and providers, including protocol or interaction-style mismatches and interface mismatches. The information that governs the behavior of mediation modules is often held in headers flowing with the business messages. The IBM SOA programming model introduces the service message object (SMO) pattern for Service Data Objects (SDOs) to support this pattern.

Unlike BEA's Aqualogic Service Bus that has proxy services within the ESB, WebSphere has Mediation flows, which are not web services and have no associated WSDL.

The way Diagnostics shows these mediation flows is by layers. A layer is displayed in Diagnostics for the Mediation Flow. The Mediation Flow component name prefixed by Web Services/WebSphere ESB/ is used as the layer name. See the example on the following pages that includes the Layers views.

This task includes the following steps:

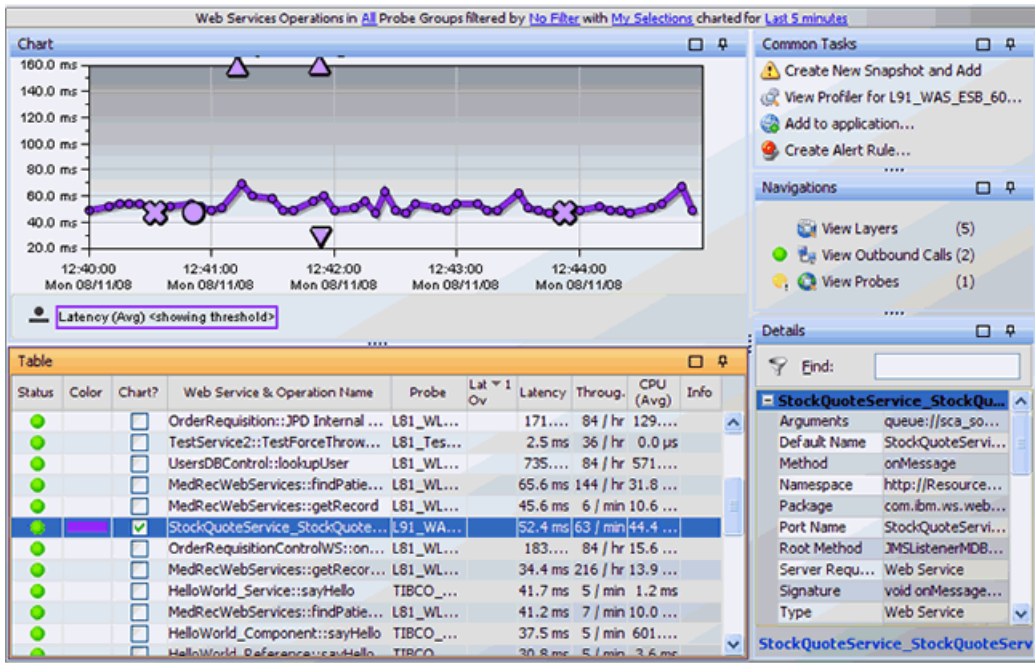
- "Web Service Operations View" on page 682
- "Call Profile View" on page 682
- "Layers View" on page 683

Web Service Operations View

The existing WebSphere Inbound and Outbound web service instrumentation in Diagnostics collects performance data on the web services.

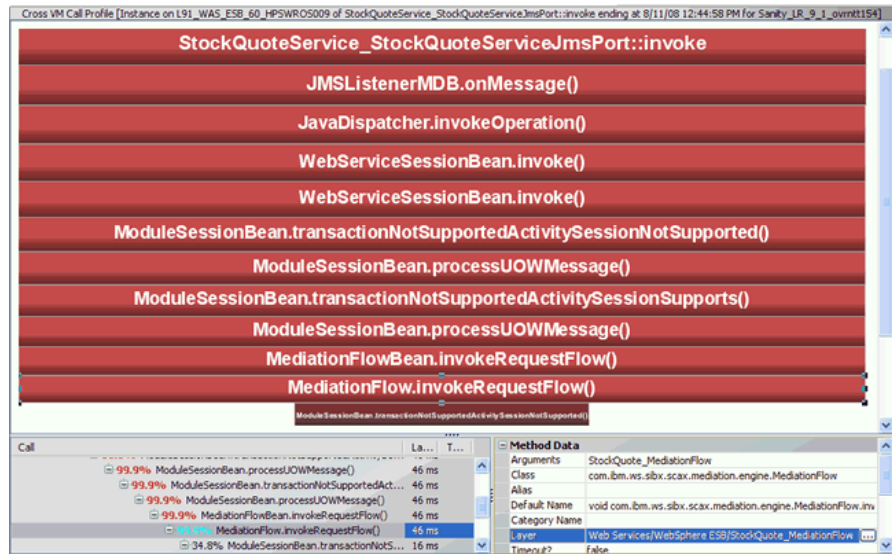
A Stock Quote example of Web Services Operations view for an environment with a WebSphere ESB is shown below.

Example:



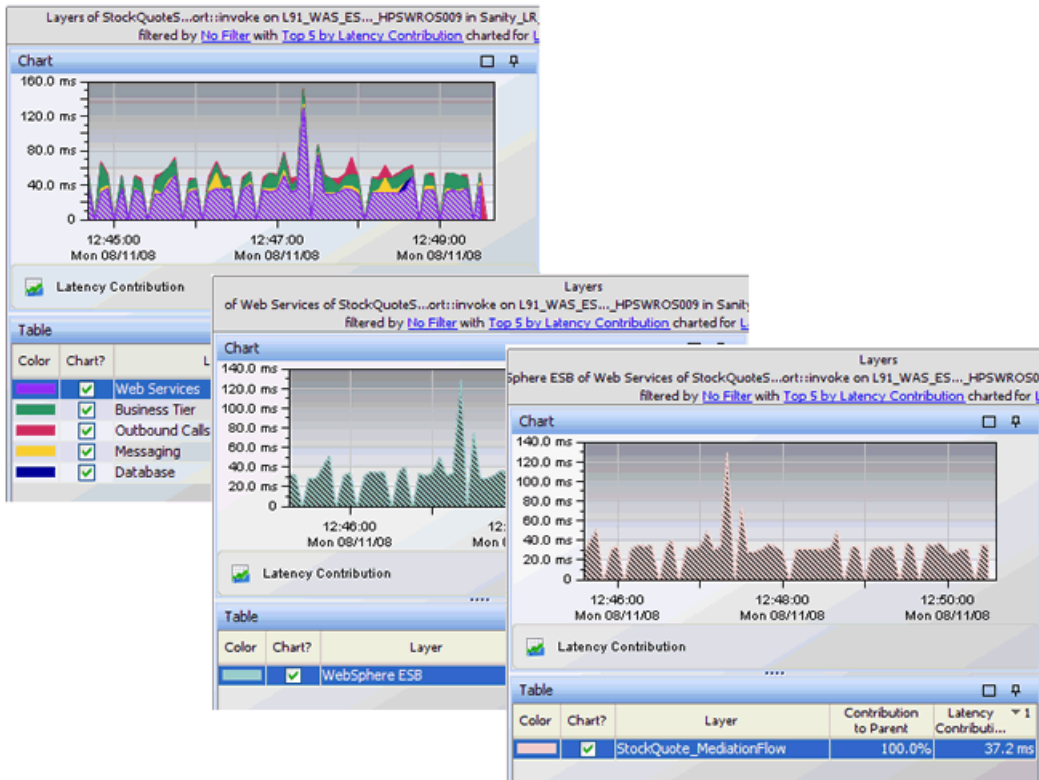
Call Profile View

Call Profile view for a WebSphere ESB environment. In the Stock Quote example both the invokeRequestFlow and invokeResponseFlow methods are instrumented for capturing the mediation flows as layers. And you see the Web Services/WebSphere ESB/StockQuote_MediationFlow layer.

Example:**Layers View**

You can also see the breakdown of the layers in the screen shots above from Web Services layer - WebSphere ESB layer to StockQuote_MediationFlow layer.

Example:



How to Monitor TIBCO ActiveMatrix Environment

Diagnostics monitors TIBCO ActiveMatrix Service Bus environments and TIBCO ActiveMatrix BusinessWorks environments.

This task includes the following steps:

- ▶ "TIBCO ActiveMatrix Service Bus Service" on page 685
- ▶ "Diagnostics Monitors Front-end and Internal Services" on page 685
- ▶ "Operations by Consumer ID View" on page 686
- ▶ "Topology Views" on page 687

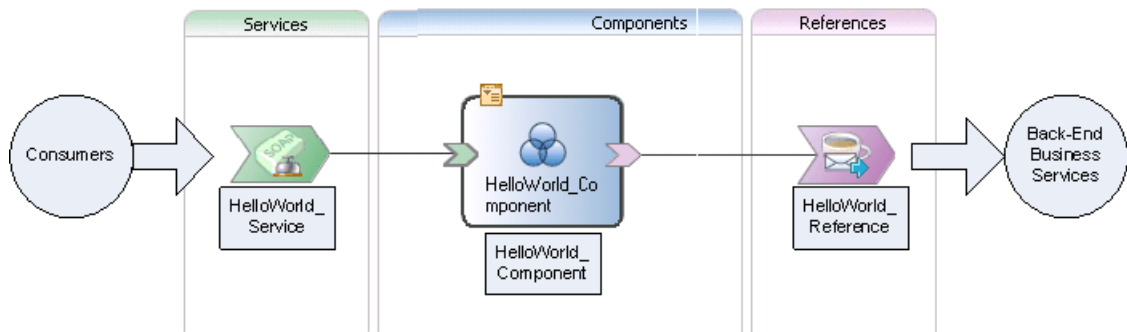
- "View Additional JMX Metrics" on page 689
- "TIBCO ActiveMatrix BusinessWorks Environment" on page 689

TIBCO ActiveMatrix Service Bus Service

TIBCO ActiveMatrix Service Bus is a lightweight enterprise service bus.

The figure below shows a typical scenario for a TIBCO ActiveMatrix Service Bus based web service. The consumers invoke the front-end "HelloWorld_Service" web service. This front-end service invokes an internal "HelloWorld_Component" mediation service. The mediation service then calls the "HelloWorld_Reference" reference service which in turn invokes the actual back-end business service.

Example:

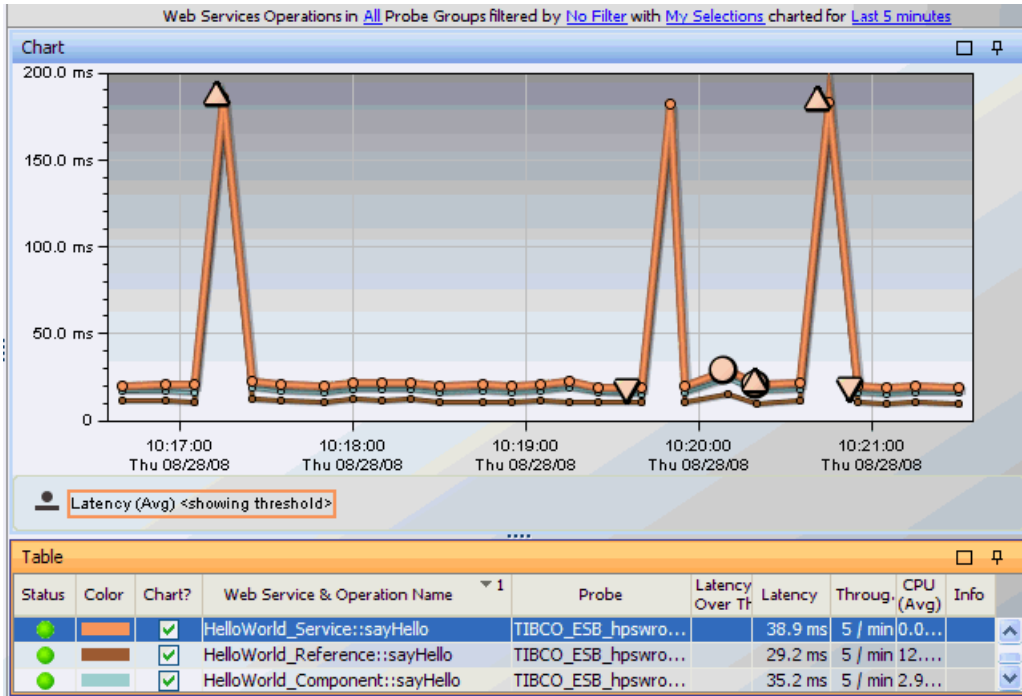


Diagnostics Monitors Front-end and Internal Services

In this type of example, Diagnostics monitors and displays performance data for the front-end "HelloWorld_Service" and the internal services "HelloWorld_Component" and "HelloWorld_Reference". The internal services are shown so that any performance bottlenecks within these services can be identified. It can also help in throughput analysis of the internal services when the mediation component is designed to make routing decisions to route to different reference services.

Example:

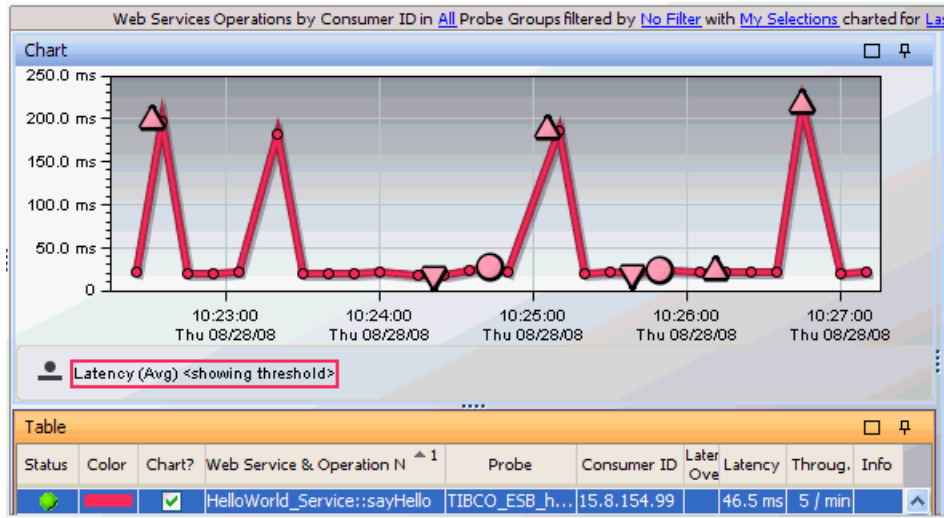
An example is shown below:



Operations by Consumer ID View

In the Web Services Operations by Consumer ID view, Diagnostics only shows the front-end service(s) because the consumer only interacts with the front-end "HelloWorld_Service" web service.

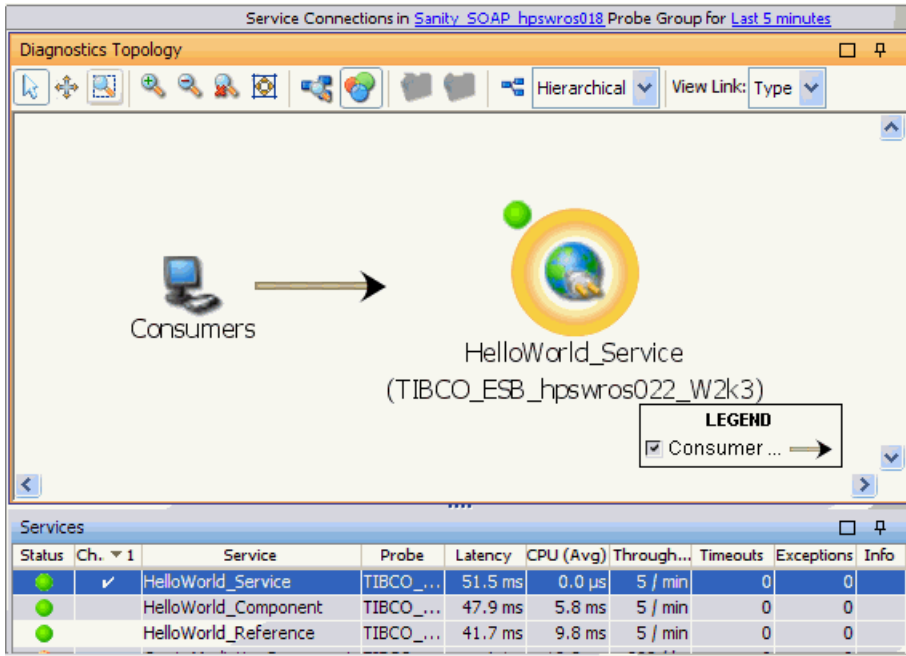
Example:

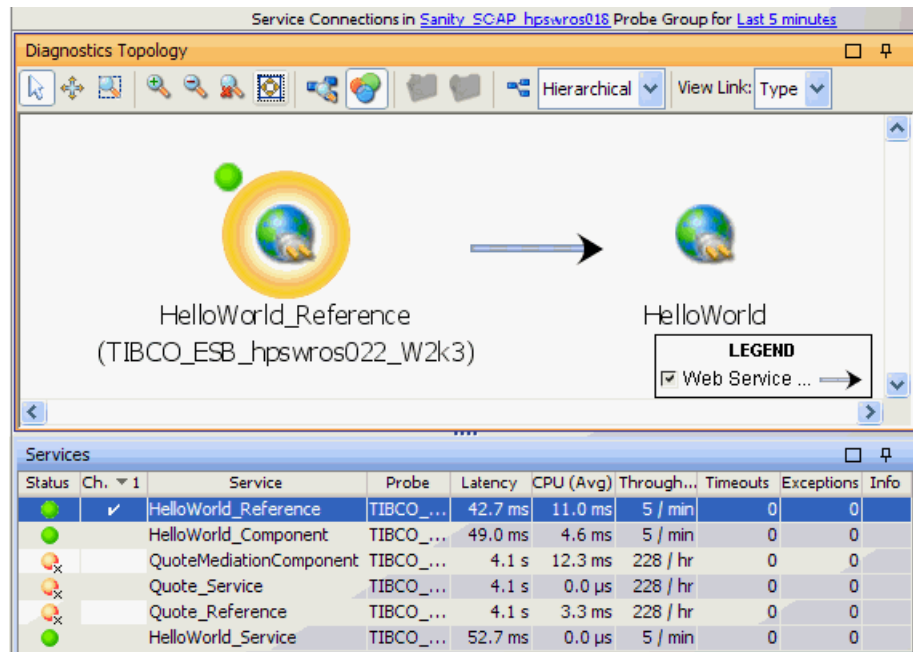


Topology Views

In the two examples below of the Service Topology view; the first shows the consumer interaction with the front-end service while the second shows the reference service invoking a back-end business service.

Example:



Example:**View Additional JMX Metrics**

Additional JMX metrics have been added as part of TIBCO Active Matrix Platform support. These metrics are exposed by the Apache Tomcat and Pramati J2EE server components used by the ActiveMatrix platform. Some of these metrics are inactive by default. They can be activated by uncommenting them in the **metrics.config** file. The reason to make them inactive is that they are mainly performance tuning configuration parameters and should rarely change during the lifetime of an application.

TIBCO ActiveMatrix BusinessWorks Environment

Diagnostics monitors the web services in a TIBCO ActiveMatrix BusinessWorks environment. For BusinessWorks you can see the typical web services monitoring data in the SOA Services views **except** the following:

- ▶ captured soap faults
- ▶ captured soap payload on faults
- ▶ web service topology

Note: To view outbound calls, cross VM correlation and the probe topology for the SOAP/HTTP calls, enable the [**Apache-HttpClient-Outbound**] point in the probe's `etc/auto_detect.points` file.

How to View BEA WLI Business Process Tracing Data

With BEA WLI's out of the box instrumentation, you can view your JPD, JWS and Controls within the Diagnostics console through many of the standard views, including Server Requests, Outbound and SOA Services screens.

In the context of Business Process Management Suites (BPMS), transaction tracing is the correlation of Business Processes with application actions. For example when a Business Process is invoked, corresponding events are instrumented on the database, application server and any ancillary systems such as messaging and legacy systems. You can use Diagnostics to make this correlation between business processes and the systems with which they interact.

These versions of BEA WLI are supported in this release: BEA Weblogic Platform v8.1.4, v8.1.5, v8.1.6.

This task includes the following steps:

- ▶ "BEA WLI View of Example Calls" on page 691
- ▶ "BEA WLI Drilldown View for MyProcess Example" on page 692
- ▶ "BEA WLI Drilldown View for StockService Example" on page 693
- ▶ "BEA WLI Test Browser View" on page 694
- ▶ "Diagnostics SOA Services Operations View of MyProcess" on page 696

- "Diagnostics SOA Services Operations View of StockService" on page 697
- "Call Profile Cross VM View" on page 698

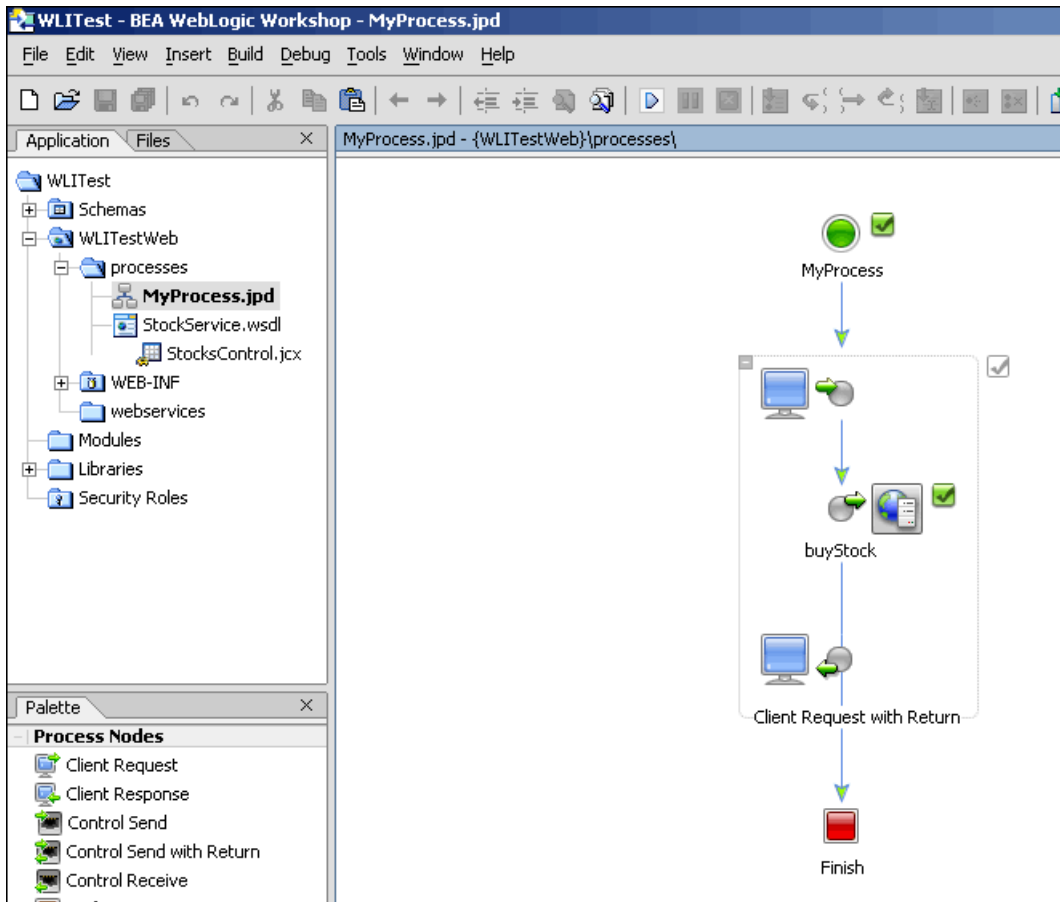
1 BEA WLI View of Example Calls

The following example demonstrates synchronous JPD calls made from MyProcess to a JWS called StockService. The example shows the BEA WLI views of these calls and the Diagnostics views of the same calls.

- MyProcess is running in wl815-jvm142-ovrntt1-wli1 WLI server (on port 7001).
- StockService is running in wl815-jvm142-ovrntt1-wli2 WLI server (on port 7003).

Example:

A typical view in Workshop is shown below:



2 BEA WLI Drilldown View for MyProcess Example

Drilling into the code for MyProcess JPD reveals the outbound call to the StockService WPS Web service. MyProcess code is shown below calling out to stocks.buyStock() Web service.

Example:

```

MyProcess.jspd - {WLItestWeb}\processes\
public void stocksBuyStock() throws Exception
{
    ##START: CODE GENERATED - PROTECTED SECTION - you can safely add code above th.
    // input transform
    // method call
    stocks.buyStock();
    // output transform
    // output assignments
    ##END : CODE GENERATED - PROTECTED SECTION - you can safely add code below th.
}

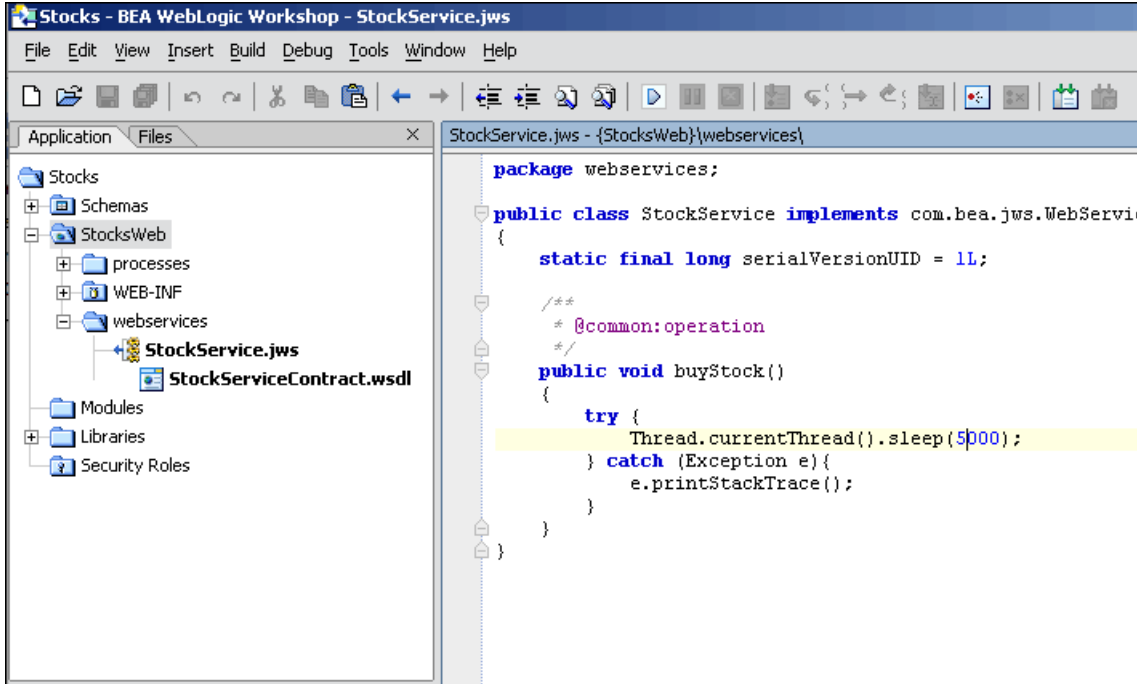
```

3 BEA WLI Drilldown View for StockService Example

On the receiving end, the StockService receives the request and simply sleeps for five seconds. The StockService Web service application is shown below.

Note: This application is deployed to another domain called **wl815-jvm142-ovrntt1-wli2**. You can see this cross-JVM call in the Diagnostics Call Profile view shown later in this section.

Example:

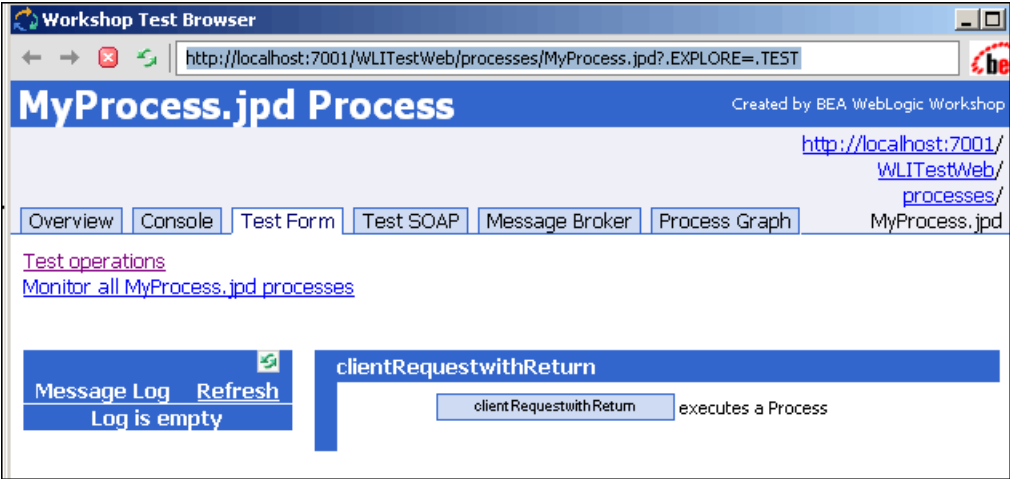


4 BEA WLI Test Browser View

The Workshop Test Browser is used to initiate the MyProcess JPD request to the StockService WPS.

The result of the request is then viewed in the Test Browser. As you can see in the second example below, the method called on the StockService is the **buyStock** method. You can also see in the left pane that the call **MyProcess.cilentRequestwithReturn** was called.

Example:



Example:

Workshop Test Browser

http://localhost:7001/WLITestWeb/processes/MyProcess.jpd?.EXPLORE=.TEST&.LOGENTRY=1

MyProcess.jpd Process Created by BEA WebLogic Workshop

http://localhost:7001/WLITestWeb/processes/MyProcess.jpd

Overview Console Test Form Test SOAP Message Broker Process Graph

Test operations
Monitor all MyProcess.jpd processes

Message Log Refresh

clientRequestwithReturn

stocks.buyStock

Clear Log

External Service Request
Submitted at Wed Apr 25 13:24:23 PDT 2007

```
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:buyStock xmlns:ns="http://www.openuri.org/"></ns:buyStock>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

External Service Response
Submitted at Wed Apr 25 13:24:31 PDT 2007

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:buyStockResponse xmlns:ns="http://www.openuri.org/">
      </ns:buyStockResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

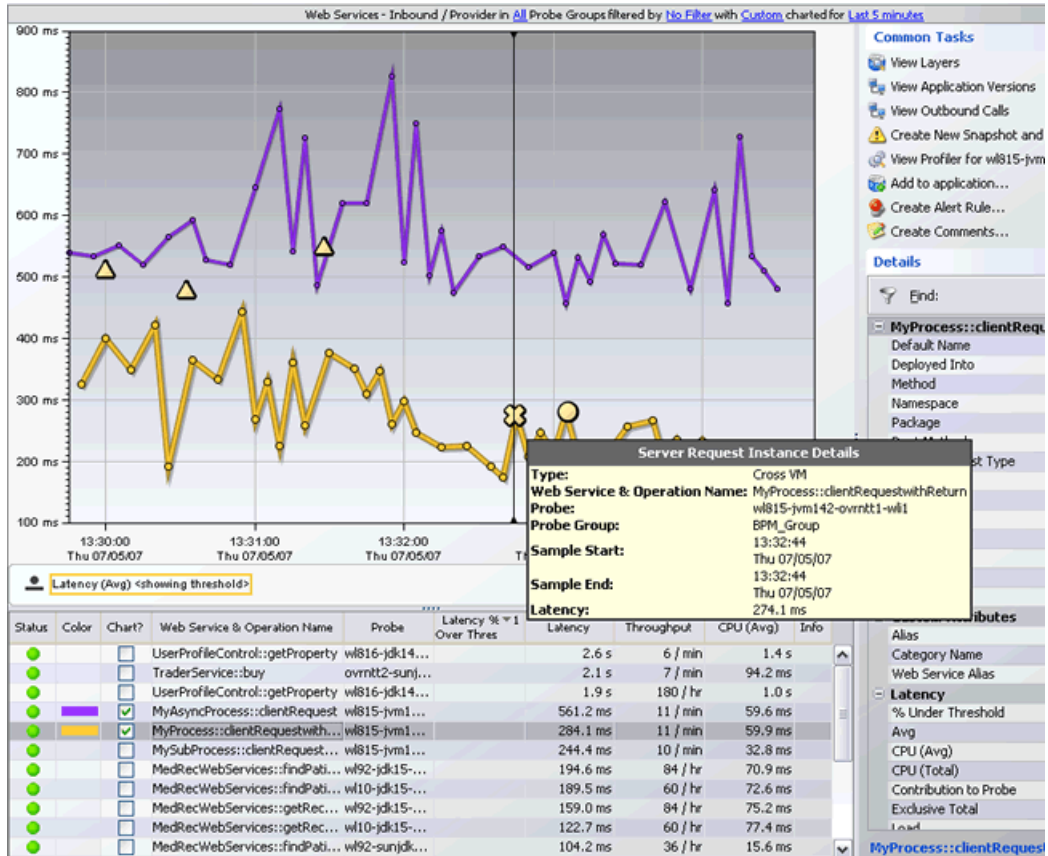
5 Diagnostics SOA Services Operations View of MyProcess

Next, open the Diagnostics server to the SOA Services - Operations view. This shows the call **MyProcess::clientRequestwithReturn** as a Cross-JVM call to the **StockService::buyStock** method.



The X icon in the example below indicates a cross-jvm call between the 'MyProcess' to the 'StockService' Web service.

Example:

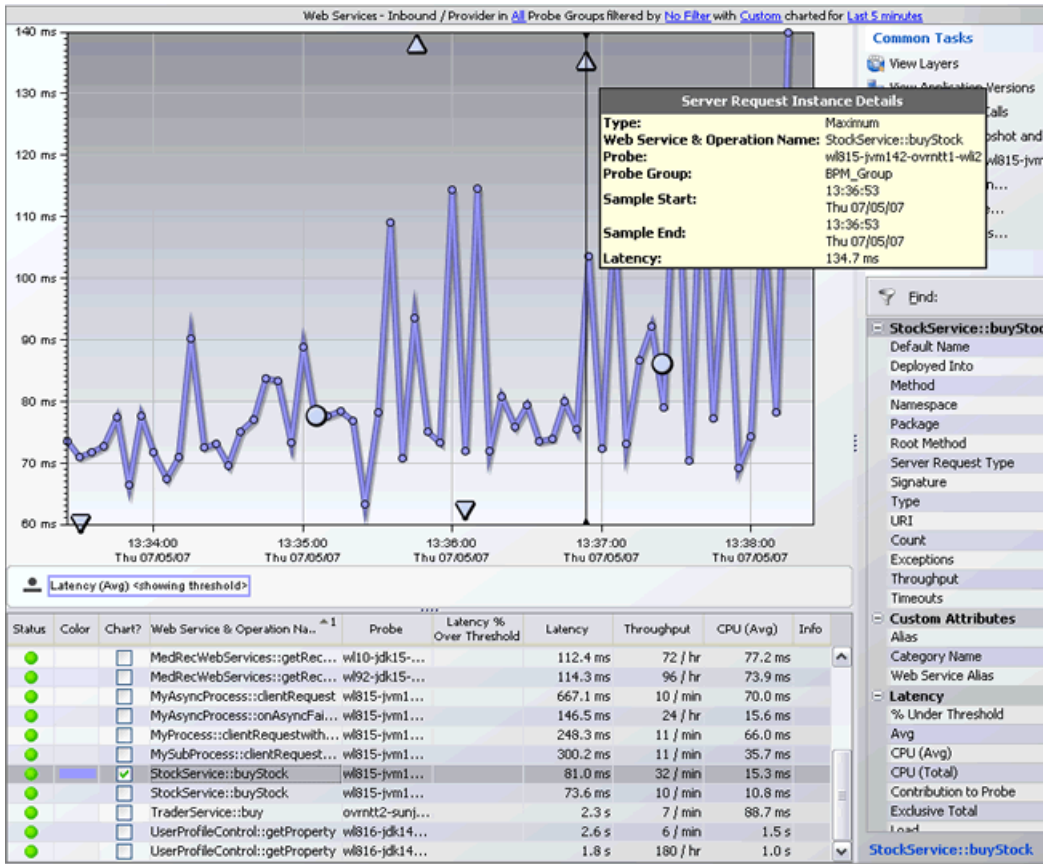


6 Diagnostics SOA Services Operations View of StockService

Since the StockService::buyStock operation is also a single server request received by the w1815-jvm142-ovrntt1-wli2 WLI server, it appears as an inbound request. Hovering over the call reveals a time of 134.7 milliseconds.

In the example below, StockService::buyStock server request is also captured.

Example:



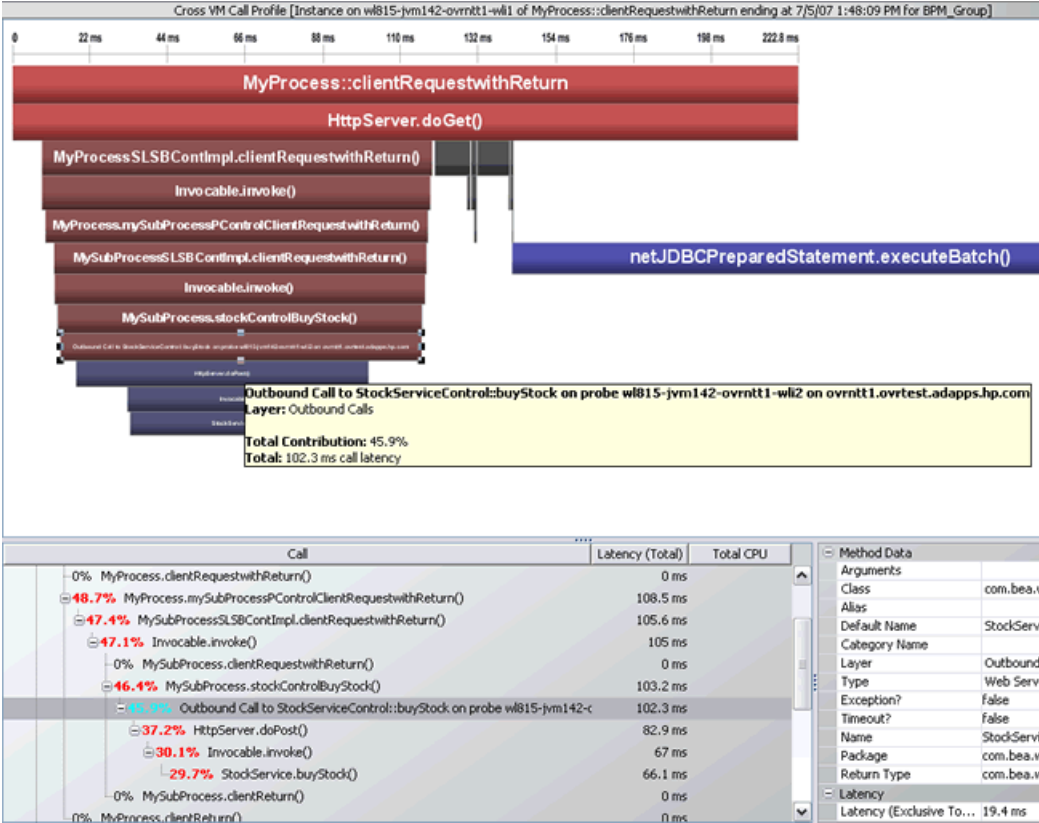
7 Call Profile Cross VM View



Clicking on the Cross-JVM X icon in the SOA Services - Operations views displays the resulting server request made from **MyProcess::clientRequestWithReturn** (on integration1 WLI server) out to the **StocksControl::buyStock** method.

Example:

Drill down into the X icon demonstrates the cross-JVM Web service call from the 'MyProcess' JPD to the 'StockService' Web Service.



Reference

SOA Services View Group User Interface

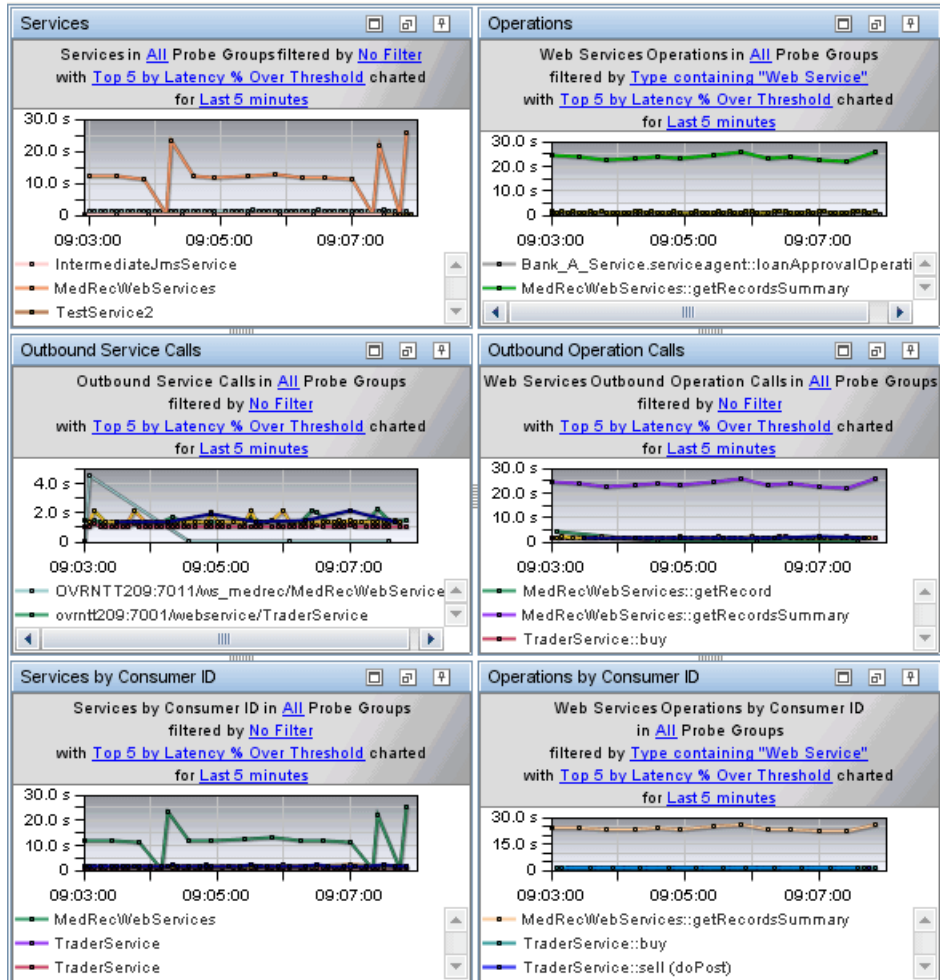
This section includes:

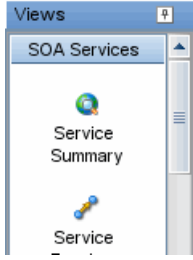
- ▶ Service Summary View Description on page 701
- ▶ Services View Description on page 704
- ▶ Operations View Description on page 709
- ▶ Service Topology Description on page 715
- ▶ Services by Consumer ID View Description on page 721
- ▶ Operations by Consumer ID View Description on page 726
- ▶ Outbound Service Calls View Description on page 733
- ▶ Outbound Operation Calls View Description on page 737

Service Summary View Description

The Service Summary view gives you a dashboard displaying the graphs from each other the other views in the SOA Services view group (except Service Topology).

The following is an example of the SOA Services - Service Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
<p>Important information</p>	<p>For SOA customers, Business Service Management 7.50 or later is required to take advantage of Diagnostics Web services enhancements (such as configurable Consumer ID) in the Business Service Management integration.</p>
<p>Relevant tasks</p>	<p>See "How to Specify a Name for the Application Server Instance" on page 667 for information on how to analyze performance using the data in these views.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Supported Web Service Platforms" on page 654 ➤ "About SOA Services Monitoring" on page 653

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

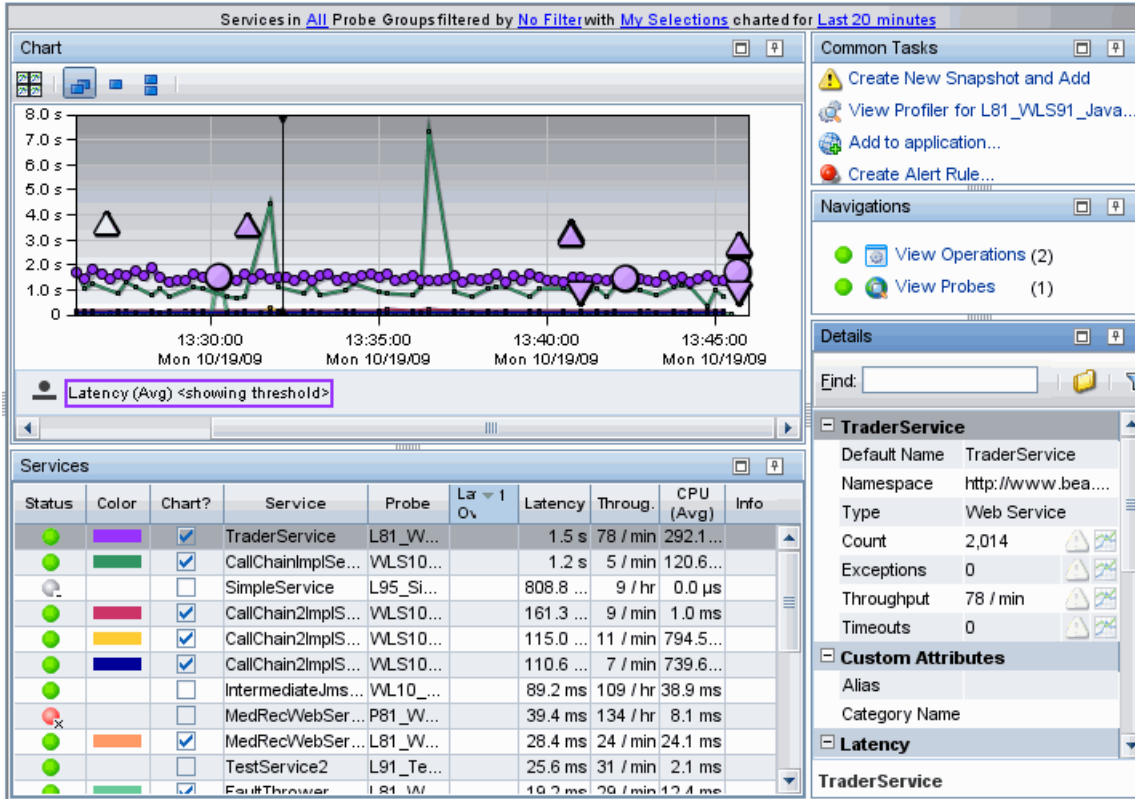
The Service Summary view is a dashboard that displays the graphs from each of the following views. The graphs show top 5 latency over threshold. You can double-click any of the graphs to drill into the related view.

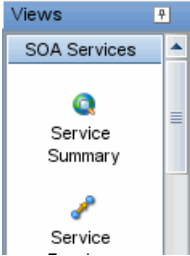
- Services view
- Operations view
- Outbound Service Calls view
- Outbound Operations Calls view
- Services by Consumer ID view
- Operations by Consumer ID view

Services View Description

The Services view gives you detailed information for the services being monitored.

The following is an example of the SOA Services - Services view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
---	--

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five services with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Web Service Details tooltip for the trend line on the graph (or the Service column in the table) displays the following detail data:</p> <p>Web Service. The name of the service. Server requests that are Web service operations are aggregated by the service name.</p> <p>Probe. The name of the probe that measured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Type. The server request type of Web Services.</p> <p>The Web Service Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created.</p> <p>You can drill down from the instance tree icon to see a call profile. See "Instance Tree Call Profiles for SOA Services" on page 656 for more information.</p> <p>And see "Call Profile View" on page 405 for how to use the Call Profile view.</p>

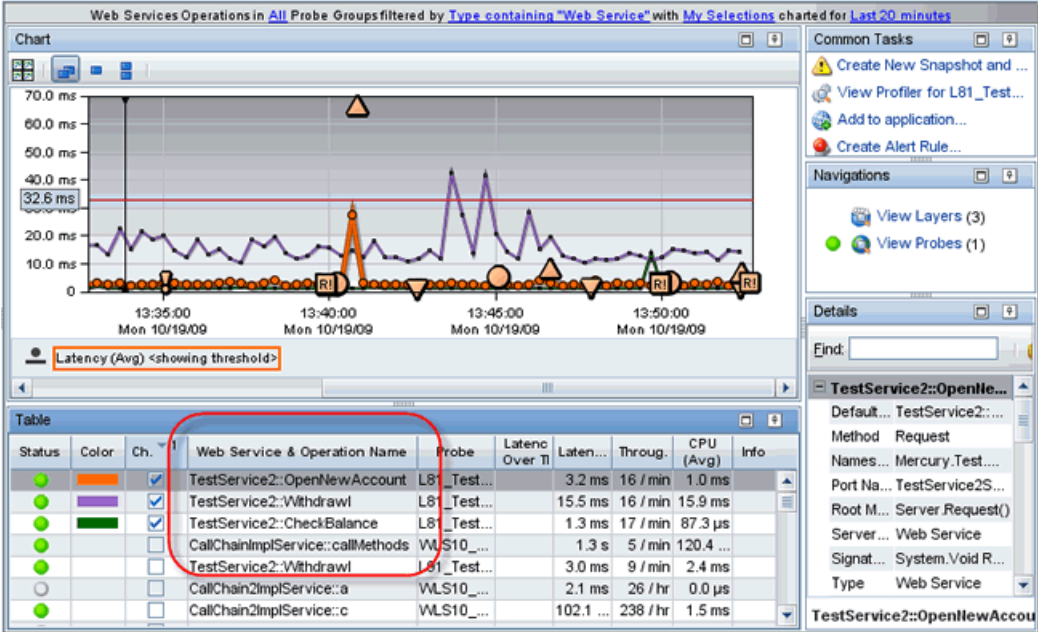
UI Elements	Description
Table	<p>Lists the services that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the service. From a service you can also see the status of its operations because any "red" status from an operation will be rolled-up to its services.</p> <p>The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Service. The name of the service.</p> <p>Probe. Name of the probe monitoring the service.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Probe Details tooltip for the Probe column in the table displays the probe, host and probe type.</p>

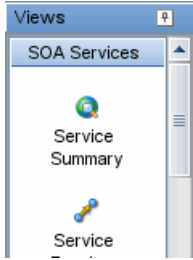
UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the web services displayed in this view.</p> <p>See "How to Specify a Name for the Application Server Instance" on page 667.</p> <p>The drill down navigation based on a pre-defined workflow is View Operations. Another common drill down is to the Probes view to see related information from the probe for the service.</p>
<p>Details Pane</p>	<p>Shows metrics collected on the service.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Operations View Description

The Operations view displays all inbound Web service operations.

The following is an example of the SOA Services - Operations view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
<p>Important information</p>	<p>If the same Web service operation was called by multiple consumers, the data from all these calls is aggregated and presented as a single Web service operation in the table.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five operations with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Web Service Operation Details tooltip for the trend line on the graph (or the Web Service and Operation Name column in the table) displays the following detail data:</p> <p>Web Service & Operation Name. The name of the operation. They appear in the following format: <Web-service-name>::<operation-name>. For example, TestService2::Withdrawal.</p> <p>Probe. The name of the probe that measured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Port Name. The port name of the Web service.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL.</p> <p>Type. The server request type of Web Services.</p> <p>Namespace. The namespace of the Web service.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Deployed Into. Tells where the web service was deployed into (usually an EAR file).</p> <p>URI. The HTTP URI (Uniform Resource Identifier).</p> <p>Method Signature. The signature of the root method.</p> <p>The Web Service Operations Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
	<p>Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created.</p> <p>You can drill down from the instance tree icon to see a call profile. See "Instance Tree Call Profiles for SOA Services" on page 656 for more information.</p> <p>And see "Call Profile View" on page 405 for how to use the Call Profile view.</p>

UI Elements	Description
Table	<p>Lists the unique Web service operations that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns and the order in which they should appear. You can also specify which columns to use to sort the rows. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the service. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Web Service & Operation Name. The name of the operation. They appear in the following format: <code><Web-service-name>::<operation-name></code>. For example, <code>TestService2::Withdrawal</code>.</p> <p>Probe. Name of the probe monitoring the service.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Probe Details tooltip for the Probe column in the table displays the probe, host and probe type.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the operations displayed in this view.</p> <p>See "How to Specify a Name for the Application Server Instance" on page 667.</p> <p>The drill down navigation based on a pre-defined workflow is View Layers. Another common drill down is to the Probes view to see related information from the probe for the operation.</p>
<p>Details Pane</p>	<p>Shows metrics collected on the operation</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

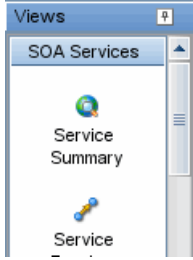
Service Topology Description

The Service Topology view provides a visualization of how Web services that communicate with each other are connected together. This is particularly useful in large complex Web service environments.

The following is an example of the SOA Services - Service Topology view.

The screenshot displays the 'Service Connections in All Probe Groups for Last 20 minutes' window. The main area shows a 'Diagnostics Topology' diagram with nodes representing services and arrows indicating connections. A legend indicates that a green circle represents a 'Web Service'. Below the diagram is a table of services with columns for Status, Chart?, Service, Probe, Late.., CPU (Avg), Throug, Timeo., Except, and Info. The table lists several instances of 'CallChain2ImplService', 'DelayedServiceS...', and 'FaultThrower'. On the right side, there are panels for 'Common Tasks' (including 'Create New Snapshot and Add', 'View Profiler for WLS10_WebSvc...', 'Add to application...', 'Create Alert Rule...', and 'Create Comments...'), 'Navigations' (with 'View Operations (5)' and 'View Probes (1)'), and 'Details' for the selected 'CallChain2ImplService' (showing Default Name, Namespace, Type, Count, Exceptions, Throughput, and Timeouts).

Status	Chart?	Service	Probe	Late..	CPU (Avg)	Throug	Timeo.	Except	Info
●	✓	CallChain2ImplSer...	WLS...	108...	787...	7 / min	0	0	
●	✓	CallChain2ImplSer...	WLS...	164...	1.4 ms	11 / ...	0	0	
●	✓	CallChain2ImplSer...	WLS...	112...	754...	11 / ...	0	0	
●	✓	CallChainImplServi...	WLS...	1.2 s	121...	6 / min	0	0	
●		DelayedServiceS...	L91_...	11.2...	11.2...	22 / hr	0	0	
●		FaultThrower	1.81	29.0	13.9	30 /	0	0	

<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
<p>See also</p>	<p>See "The Topology Toolbar" on page 249</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
<Title bar filters>	<p>You can filter the data by service connections in a particular probe group or time range. The filter selections are located above the Topology toolbar.</p> <p>If you select a filter that results in no matching data the diagram will be replaced by the initial topology message.</p>
Diagnostics Topology	<p>The topology diagram is a mapping of services and service-consumer connections. It provides you with a visualization of the web service.</p> <p>If you see a message in place of the topology diagram, you must select an entity in the table to generate a topology.</p> <p>The topology display includes a toolbar with controls that allow you to pan, zoom, and alter the way the diagram is drawn by grouping entities, changing diagram type, or changing connection (link) information. To find out how to use the diagram tool bar, see Chapter 8, "Working with Topologies."</p> <p>Note that the Service Topology view does not provide the ability to group by host and probe group.</p> <p>The topology diagram and the tables are integrated so that if you select an entity in the diagram then the relevant tab and entity in the table are highlighted. And if you select an entity in the table a topology is generated for that selection.</p> <p>The selected entity is highlighted in the diagram: services are highlighted with a yellow halo, connections are highlighted with a striped line.</p> <p>A probe icon is displayed in the diagram when you select a service arc that originates from a probe. This happens when the call to a Web service originates from an instrumented application that is itself not a web service. A probe icon is displayed so you have some information about where the call originated from.</p> <p>You can right-click items on the diagram to see a tooltip containing Name, Type, Latency, and Throughput.</p>

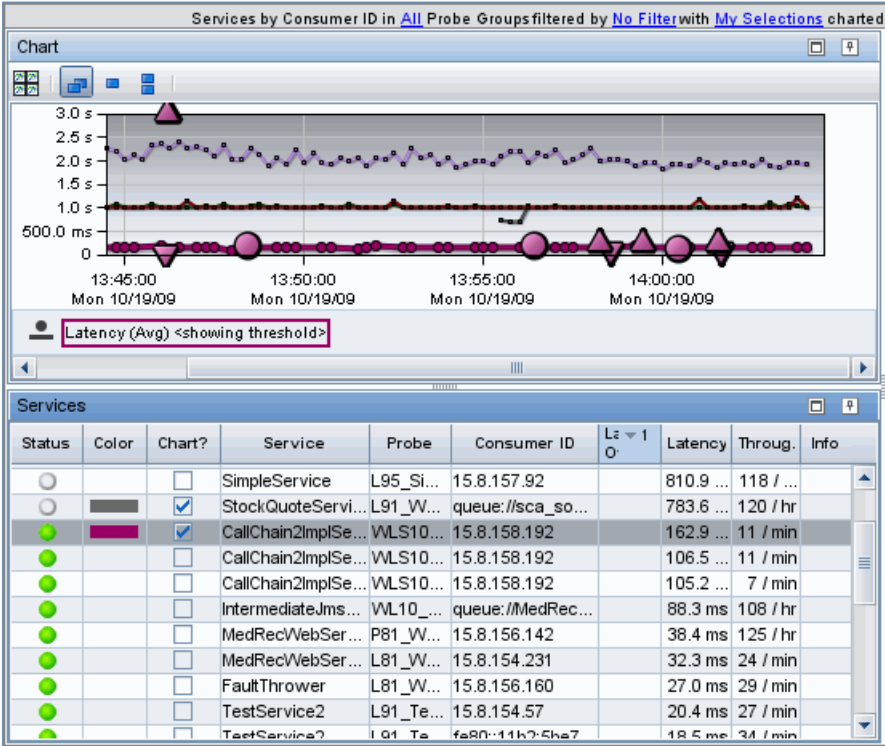
UI Elements	Description
<p>Table</p>	<p>Contains the following tabs:</p> <ul style="list-style-type: none"> ➤ Services ➤ Service Connections ➤ Consumer Connections <p>Once you generate a topology, each tab in the table will have check marks in the Chart column for those items that are represented in the currently displayed topology.</p> <p>The Services tab in the table contains the following columns:</p> <p>Status. By default the first column shows the status for the service. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Chart?. From the table you can select the entities that are to be included in the topology diagram.</p> <p>Service. The name of the service.</p> <p>Probe. The name of the probe.</p> <p>Latency. The average response time across all instances.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Web Service Details tooltip for the Service column displays Web service name, probe name, probe group name, and server request type.</p> <p>Probe Details tooltip for the Probe column displays probe name, host name and probe type.</p>


UI Elements	Description
	<p>The Service Connections tab provides information on how services are 'stitched' together. You can see the origin and destination of the calls to and from web services (inbound and outbound) as well as status, latency and throughput.</p> <p>The Origin columns for JMS web service calls will show the queue_name or topic_name/web_service_name. For example:</p> <pre>queue:// MedRec-jms!weblogic.wsee.reliability.wseeMedRecDestinationQueue/JMSTransportService</pre> <p>The Consumer Connections tab provides information on service calls from the consumer node. The consumer node represents calls to a web service coming from a non-instrumented source (not monitored by a probe). These calls are grouped together and the Consumer Connections displays allows you to identify specific consumers calling the web service in the connection.</p> <p>The Service Connections tab and the Consumer Connections tab in the table contains similar data as described above with these additional columns:</p> <p>Origin. Origin of the service connection.</p> <p>Destination. Destination of the service connection.</p> <p>Outbound Web Service Connection Details tooltip for the Origin and Destination columns displays the outbound web service connection and call type.</p>
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the operations displayed in this view.</p> <p>See "How to Specify a Name for the Application Server Instance" on page 667.</p> <p>From the Services tab you can drill down to View Operations or View Probes.</p>
<p>Details Pane</p>	<p>Shows metrics collected on the services.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Services by Consumer ID View Description

The Services by Consumer ID view shows a breakdown by consumer for a service. This allows you to see who is making a Web service request or where the traffic is coming from. You can identify who is impacted by performance problems on a particular service and see differences in performance by consumer.

The following is an example of the SOA Services - Services by Consumer ID view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
<p>Important information</p>	<p>You can configure what you want in the Consumer ID field and the number of Consumer IDs to be monitored using property files (see "About Consumer IDs" on page 663).</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five services with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Web Service Details tooltip for the trend line on the graph (or the Service column in the table) displays the following detail data:</p> <p>Web Service. The name of the service. Server requests that are Web service operations are aggregated by the service name.</p> <p>Source. Consumer making the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Type. The server request type of Web Services.</p> <p>The Web Service Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created.</p> <p>You can drill down from the instance tree icon to see a call profile. See "Instance Tree Call Profiles for SOA Services" on page 656 for more information.</p> <p>And see "Call Profile View" on page 405 for how to use the Call Profile view.</p>

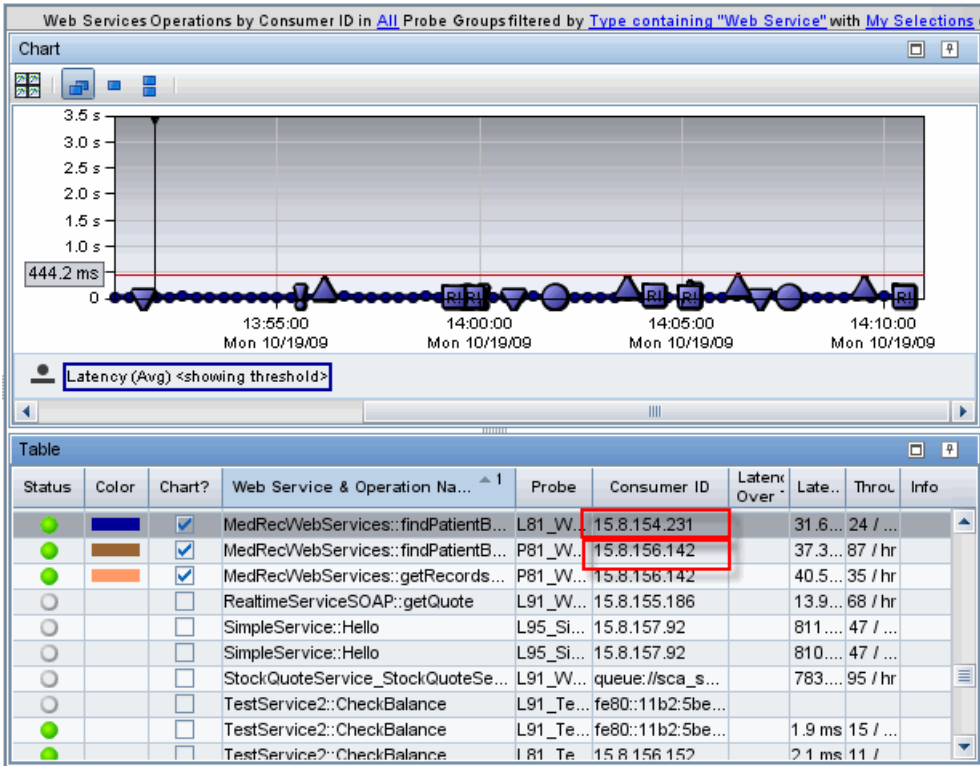
UI Elements	Description
Table	<p>Lists the services that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the service. From a service you can also see the status of its operations because any "red" status from an operation will be rolled-up to its services. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Service. The name of the service.</p> <p>Probe. Name of the probe monitoring the service.</p> <p>Consumer ID. Identifies the Web service consumer.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>CPU (Avg). Average CPU utilization.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Inbound Call Details tooltip for the Probe column in the table displays the source, call type and destination probe.</p>

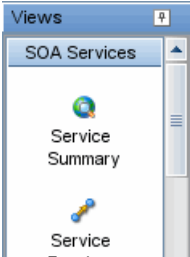
UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the web services displayed in this view.</p> <p>See "How to Specify a Name for the Application Server Instance" on page 667.</p> <p>The drill down navigation based on a pre-defined workflow is View Operations by Consumer ID.</p>
Details Pane	<p>Shows metrics collected on the service.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Operations by Consumer ID View Description

The Operations by Consumer ID view displays inbound Web service operation calls according to the consumer ID from which they originated.

The following is an example of the SOA Services - Operations by Consumer ID view.



<p>To access</p>  <p>The screenshot shows a 'Views' window with a search icon. The 'SOA Services' view group is selected and expanded, showing two options: 'Service Summary' with a globe icon and 'Service' with a plug icon.</p>	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
<p>Important information</p>	<p>You can configure what you want in the Consumer ID field and the number of Consumer IDs to be monitored using property files.</p> <p>If the same Web service operation was called by multiple consumers, the Web service operation is displayed in separate rows of the graph entity table for each unique consumer ID. The actual consumer ID is displayed in the Consumer ID column.</p> <p>Diagnostics creates samples for each consumer ID that calls a Web service. These samples are sent to Business Service Management when the two products are integrated. Increasing the number of consumer IDs captured by Diagnostics will therefore increase the amount of data sent to, and processed by, Business Service Management. You can stop Diagnostics from creating these samples.</p> <p>See "About Consumer IDs" on page 663.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five services with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Web Service Operation Details tooltip for the trend line on the graph (or the Web Service and Operation Name column in the table) displays the following detail data:</p> <p>Web Service & Operation Name. The name of the operation. They appear in the following format: <Web-service-name>::<operation-name>. For example, TestService2::Withdrawal.</p> <p>Source. Consumer making the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Port Name. The port name of the Web service.</p> <p>Root Method. The method that originated from the server request. This method may be a portion of a URL.</p> <p>Type. The server request type of Web Services.</p> <p>Namespace. The namespace of the Web service.</p> <p>Package. The name of the package that contains the class from which the method was called.</p> <p>Deployed Into. Tells where the web service was deployed into (usually an EAR file).</p> <p>URI. The HTTP URI (Uniform Resource Identifier).</p> <p>Method Signature. The signature of the root method.</p> <p>The Web Service Operations Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
	<p>Special icons are displayed to mark the points at which instance trees (for example, minimum, maximum and average) have been created.</p> <p>You can drill down from the instance tree icon to see a call profile. See "Instance Tree Call Profiles for SOA Services" on page 656 for more information.</p> <p>And see "Call Profile View" on page 405 for how to use the Call Profile view.</p>

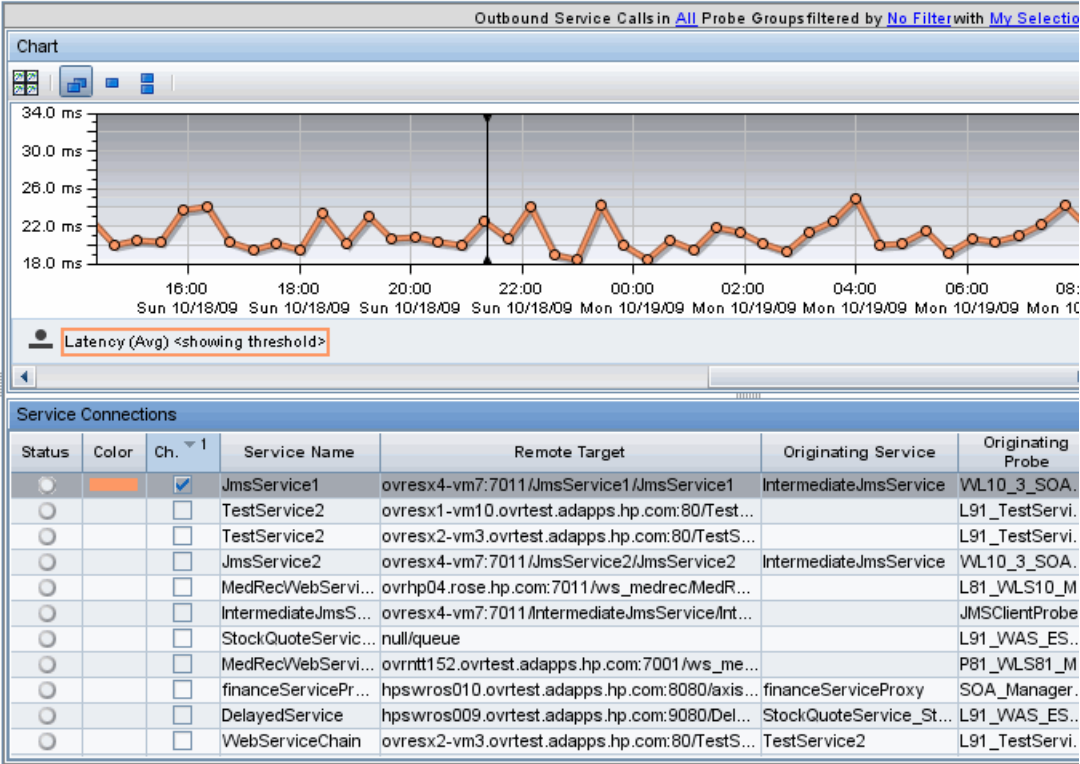
UI Elements	Description
Table	<p>Lists the operations that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the operation. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Web Service & Operation Name. The name of the operation. They appear in the following format: <Web-service-name>::<operation-name>. For example, TestService2::Withdrawal.</p> <p>Probe. Name of the probe monitoring the operation.</p> <p>Consumer ID. Identifies the Web service consumer.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Inbound Call Details tooltip for the Probe column in the table displays the source, call type and destination probe.</p>


UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the web services displayed in this view. See "How to Specify a Name for the Application Server Instance" on page 667. The drill down navigation based on a pre-defined workflow is View Layers.
Details Pane	Shows metrics collected on the operation. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

📌 Outbound Service Calls View Description

Information about the outbound service calls made from your monitored environment is displayed in the Outbound Service Calls view.

The following is an example of the SOA Services - Outbound Service Calls view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SOA Services to open the SOA Services view group.</p> <p>If the SOA Services view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SOA Services and click OK. The SOA Services view group is displayed in the View bar.</p> <p>Select the appropriate view from the SOA Services view group.</p>
---	--

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five services with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Outbound Web Service Connection Details tooltip for the trend line on the graph (or the Service Name column in the table) displays the Outbound Web Service Connection, Remote Target, Probe Group and Call Type.</p> <p>The Outbound Web Service Connection Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

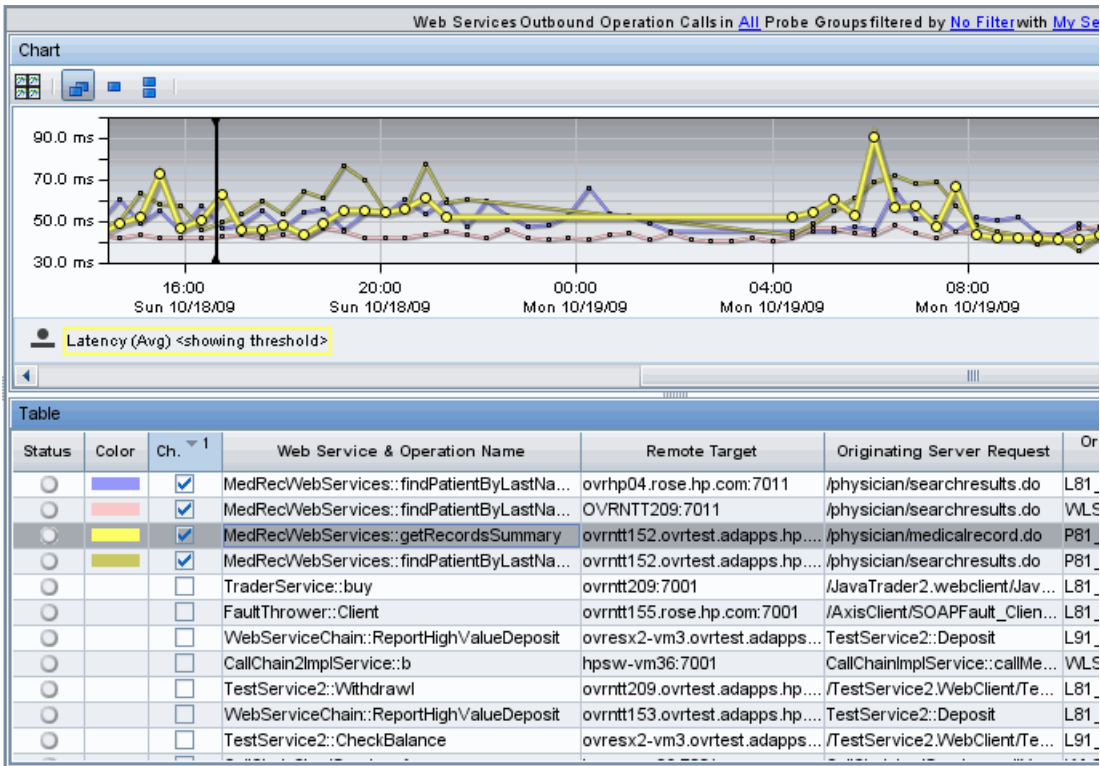
UI Elements	Description
Table	<p>Lists the services that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the called service. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart? From the table you can select the entities that are to be charted on the graph.</p> <p>Service Name. The name of the service being called.</p> <p>Remote Target. For SOAP over HTTP/S web services, the hostname and port number of the Web service that is being called (displayed as <host:port/URI>).</p> <p>For SOAP over JMS web services, the queue name (or topic name) of the web service being called (queue://<queue name/web service name>). Queue name comes from the outbound JMS information and web service name comes from the outbound WS information.</p> <p>For JMS outbound calls it is possible to see a blank remote target. Some Application Server vendor's internal processes use some of the same instrumented methods that Diagnostics use for the remote target of a web services. Since these calls are internal to the Application Server vendor they are not actually web service calls so the remote target will be blank.</p>

UI Elements	Description
	<p>Originating Service. The service making the call. Originating service will be blank when the call was made from an instrumented application that was not itself a Web service.</p> <p>Originating Probe. The probe from which this Web service was called.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	<p>Common tasks and actions relevant to the web services displayed in this view.</p> <p>See "How to Specify a Name for the Application Server Instance" on page 667.</p>
Details Pane	<p>Shows metrics collected on the service.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

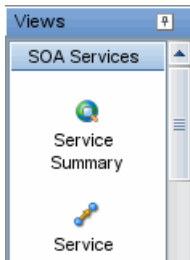
Outbound Operation Calls View Description

Information about the outbound operations calls made from within your monitored environment is displayed in the Outbound Operations Calls view.

The following is an example of the SOA Services - Outbound Operation Calls view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SOA Services** to open the SOA Services view group.

If the SOA Services view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SOA Services** and click **OK**. The SOA Services view group is displayed in the View bar.

Select the appropriate view from the SOA Services view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five operations with the highest average latency during the selected time range are charted.</p> <p>The x-axis of the graph shows the actual chronological time. The y-axis of the graph shows the average latency in seconds and milliseconds.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Outbound Call Details tooltip for the trend line on the graph (or the Web Service & Operation Name column in the table) displays the following:</p> <p>Name. The name of the operation being called.</p> <p>Remote Target. Remote target of the outbound call.</p> <p>Probe Group. The name of the probe group.</p> <p>Originating Request. Name of the originating server request.</p> <p>Call Type. The server request call type of Web service.</p> <p>The Outbound Call Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the operations that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the operation. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph. It is displayed in the following format: <code><Web-service-name>::<operation-name></code>. For example, <code>CurrencyConversionService::ConvertToNum</code>.</p> <p>Remote Target. For SOAP over HTTP/S web services, the hostname and port number of the operation being called (displayed as <code><host:port></code>). For SOAP over JMS web services, the queue name or topic name of the operation being called (displayed as <code>queue://<queue name></code>).</p> <p>Originating Server Request. The server request from which this operation was called.</p> <p>Originating Probe. The probe from which this operation was called.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the web services displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests.</p> <p>See "How to Specify a Name for the Application Server Instance" on page 667.</p>
Details Pane	<p>Shows metrics collected on the operation.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

29

SAP Views

This chapter describes the specialized views for SAP.

This chapter includes:

Concepts

- ▶ SAP Overview on page 744

Reference

- ▶ SAP View Group User Interface on page 745

Concepts

SAP Overview

The **SAP** view group displays two different types of SAP performance data:

- ▶ **SAP ABAP.** Represents the ABAP stack of the SAP system.

When you install and configure the Diagnostics Collector to gather data from a SAP ABAP system, you define instances of SAP ABAP to be monitored. Each one of these instances is represented as an SAP ABAP Probe, belonging to a probe group, in the HP Diagnostics UI.

HP Diagnostics displays SAP ABAP performance data and metrics in the SAP view group.

- ▶ **NetWeaver.** Represents the SAP Web Application Server (WAS) Java stack.

The NetWeaver data is collected by the Diagnostics Probe for Java. HP Diagnostics displays SAP NetWeaver performance data and metrics in the SAP view group.

The Diagnostics Collector is used to gather SAP performance metrics. See the *HP Diagnostics Installation and Configuration Guide* chapter on "Installing the Diagnostics Collector" for details on configuring the SAP collector.

The SAP view group includes the following views:

- ▶ NetWeaver Summary view
- ▶ ABAP Summary view
- ▶ NetWeaver Requests view
- ▶ NetWeaver Threads view
- ▶ ABAP Probes view
- ▶ ABAP Server Requests view

For details on the user interface, see "SAP View Group User Interface" on page 745.

Reference

SAP View Group User Interface

This section includes:

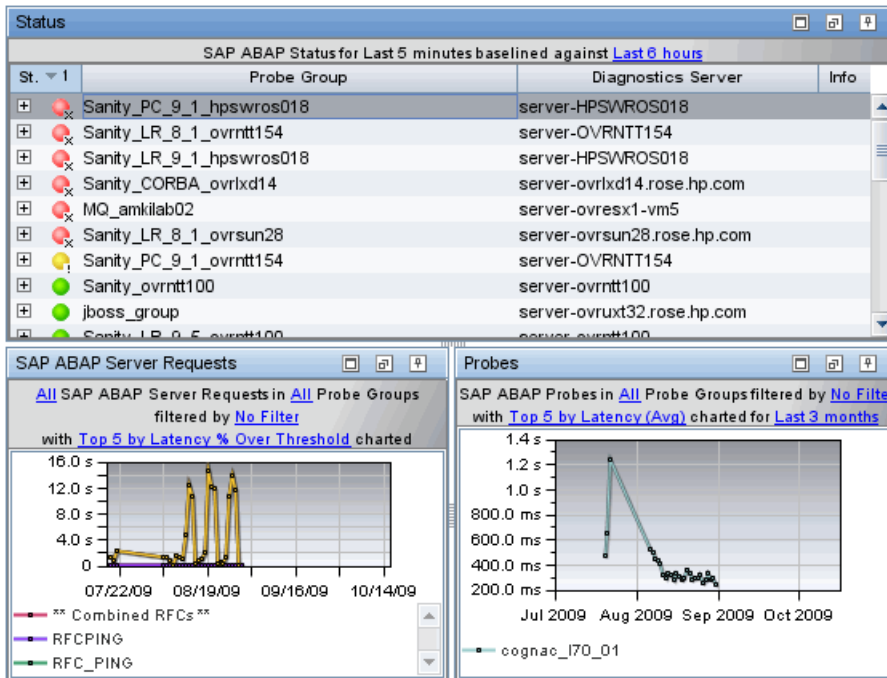
- ▶ ABAP Summary View Description on page 745
- ▶ NetWeaver Summary View Description on page 749
- ▶ ABAP Server Requests View Description on page 756
- ▶ NetWeaver Requests View Description on page 761
- ▶ NetWeaver Threads View Description on page 766
- ▶ ABAP SAP Probes View Description on page 770
- ▶ SAP Probe Summary View Description on page 774

ABAP Summary View Description

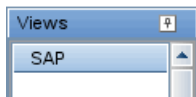
The ABAP Summary dashboard view contains a summary of the SAP ABAP data collected by Diagnostics. The ABAP Summary dashboard view contains the following views:

- ▶ Status table
- ▶ SAP ABAP Server Requests graph
- ▶ SAP ABAP Probes graph

The following is an example of the ABAP Summary view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SAP** to open the SAP view group.

If the SAP view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SAP** and click **OK**. The SAP view group is displayed in the View bar.

Select the appropriate view from the SAP view group.

Status Table

The following user interface elements are included:

UI Elements	Description
Status Table	<p>A monitoring version of the standard Diagnostics Status view. All Diagnostics probe groups, probes, and hosts are displayed in this view, not just those that are SAP ABAP related.</p> <p>The only metrics in the table that are relevant for the SAP ABAP Probes are the Latency and Count metrics.</p> <p>For the SAP ABAP hosts, the table displays CPU metrics. You can view unique performance metrics for the SAP ABAP hosts by drilling down to the Diagnostics Hosts view.</p> <p>For more information about the Status view, see "Status View Description" on page 520.</p>

ABAP Summary Graphs

The ABAP Summary view contains monitoring versions (charts/graphs) of the ABAP Server Requests view and the Probes view.:

UI Elements	Description
<p>ABAP Server Requests graph</p>	<p>By default, the graph displays the top five SAP ABAP server requests with the highest latency percent over threshold during the previous five minutes.</p> <p>By default, the graph is based on all SAP ABAP server requests for all probe groups. You can filter the data by SAP ABAP server requests or individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>ABAP Server Request Details tooltip for the trend line on the graph displays the following detail data:</p> <p>ABAP Server Request. The name of the selected ABAP server request.</p> <p>Probe. The name of the probe that captured the ABAP server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Type. SAP ABAP server request type (for example Batch).</p> <p>The ABAP Server Request Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the ABAP Server Requests view.</p>

UI Elements	Description
Probes graph	<p>By default, the graph displays the top five SAP ABAP probes with the highest latency during the previous five minutes.</p> <p>By default, the graph is based on all SAP ABAP probes for all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SAP application.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. SAP ABAP probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the ABAP Probes view.</p>

NetWeaver Summary View Description

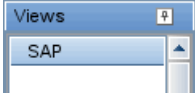
The NetWeaver Summary dashboard view contains a summary of the SAP NetWeaver data collected by Diagnostics. The NetWeaver Summary dashboard view contains the following views:

- Status table
- Server Requests graph
- Portal Component graph
- Probes graph
- NetWeaver Threads graph

► NetWeaver Requests graph

The following is an example of the NetWeaver Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SAP to open the SAP view group.</p> <p>If the SAP view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SAP and click OK. The SAP view group is displayed in the View bar.</p> <p>Select the appropriate view from the SAP view group.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Server Requests View" on page 379 ➤ "Portals Views" on page 613 ➤ "Probes View" on page 433 ➤ "NetWeaver Threads View Description" on page 766 ➤ "NetWeaver Requests View Description" on page 761

Status Table

The following user interface elements are included:

UI Elements	Description
<p>Status Table</p>	<p>A monitoring version of the standard Diagnostics Status view. All Diagnostics probe groups, probes, and hosts are displayed in this view, not just those that are SAP NetWeaver related.</p> <p>For more information about the Status view, see "Status View Description" on page 520</p>

NetWeaver Summary Graphs

The NetWeaver Summary view contains monitoring versions (charts/graphs) of the Server Requests view, Portal Components view, Probes view, NetWeaver Threads view and NetWeaver Requests view.

UI Elements	Description
<p>Server Requests graph</p>	<p>A monitoring version of the standard Diagnostics Server Requests view.</p> <p>By default, the graph displays the top five server requests with the highest latency percent over threshold during the previous five minutes.</p> <p>By default, the graph is based on all server requests for all probe groups. You can filter the data by server request types or individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>Server Request Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Server Request. The name of the selected server request.</p> <p>Probe. The name of the probe that captured the ABAP server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Type. The server request type.</p> <p>The Server Request Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Server Requests view.</p> <p>For more information about the Server Requests view, see "Server Requests View" on page 379.</p>

UI Elements	Description
<p>Portal Components graph</p>	<p>A monitoring version of the standard Diagnostics Portal Components view.</p> <p>By default, the graph displays the top five portal component with the highest latency percent over threshold during the previous five minutes.</p> <p>By default, the graph is based on data for probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency.</p> <p>Portal Component Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Portal Component. The name of the portal component.</p> <p>Probe. The name of the probe that captured the server request.</p> <p>Probe Group. The name of the probe group.</p> <p>The Portal Component Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Portal Components view.</p> <p>For more information about the Portal Components view, see "Portals Views" on page 613.</p>

UI Elements	Description
<p>Probes graph</p>	<p>A monitoring version of the standard Diagnostics Probes view.</p> <p>By default, the graph displays the top five probes with the highest average VM Heap Used during the previous five minutes.</p> <p>By default, the graph is based on data for all probes and probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the Average VM Heap Used.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SAP application.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Type of probe.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Probes view.</p> <p>For more information on the Probes view, see "Probes View" on page 433.</p>

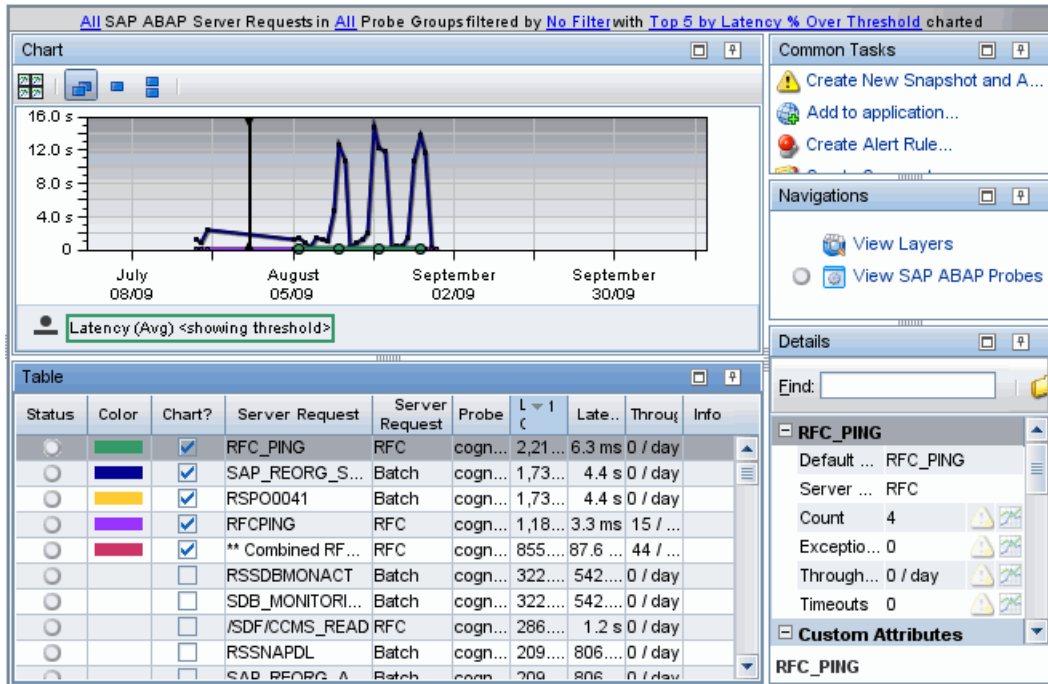
UI Elements	Description
NetWeaver Threads graph	<p>A monitoring version of the NetWeaver Threads view.</p> <p>By default, the graph displays the top five NetWeaver threads with the highest average application threads/active threads during the previous five minutes.</p> <p>By default, the graph is based on data for all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average application threads/active threads.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SAP application.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Type of probe.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the NetWeaver Threads view.</p> <p>For more information on the NetWeaver Threads view, see "NetWeaver Threads View Description" on page 766.</p>

UI Elements	Description
<p>NetWeaver Requests graph</p>	<p>A monitoring version of the NetWeaver Requests view.</p> <p>By default, the graph displays the top five NetWeaver requests with the highest average number HTTP and P4 requests per second during the previous five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average HTTP and P4 requests per second.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SAP application.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Type of probe.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the NetWeaver Requests view.</p> <p>For more information on the NetWeaver Requests view, see "NetWeaver Requests View Description" on page 761.</p>

ABAP Server Requests View Description

The ABAP Server Requests view displays information about the server requests running on the ABAP stack of the SAP system.

The following is an example of the ABAP Server Requests view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SAP** to open the SAP view group.

If the SAP view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SAP** and click **OK**. The SAP view group is displayed in the View bar.

Select the appropriate view from the SAP view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five ABAP server requests that have the highest average latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular ABAP server request or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>ABAP Server Request Details tooltip for the trend line on the graph (or the Server Requests column in the table) displays the following detail data:</p> <p>ABAP Server Request. The name of the selected ABAP server request.</p> <p>Probe. The name of the probe that captured the ABAP server request.</p> <p>Probe Group. The name of the probe group.</p> <p>Type. SAP ABAP server request type (for example Batch, Dialog, RFC).</p> <p>The ABAP Server Request Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

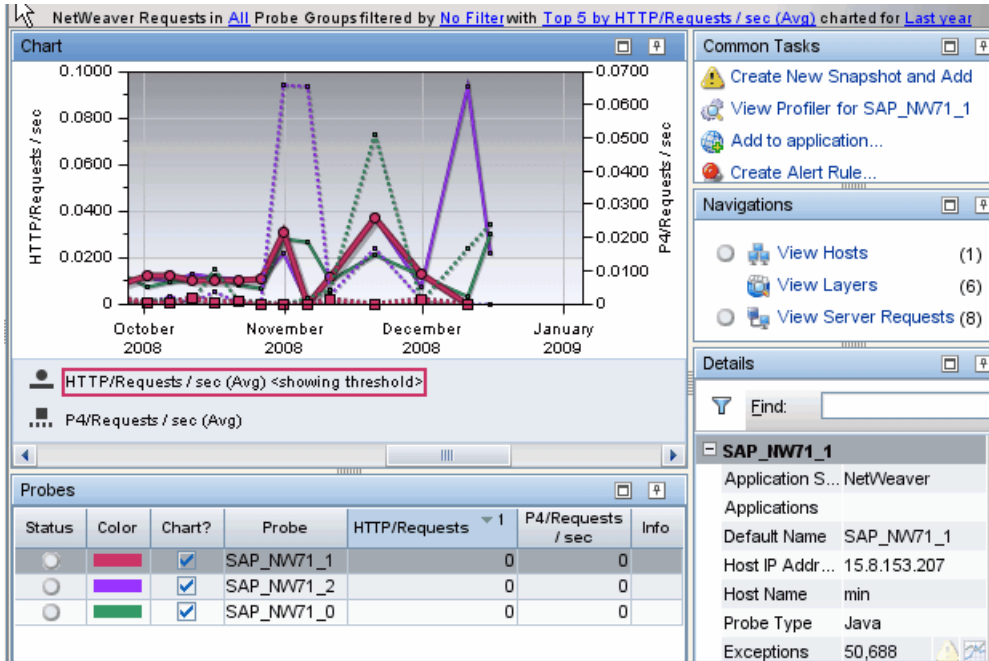
UI Elements	Description
Table	<p>Lists the ABAP server requests that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the server request. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Server Request. Name of the ABAP server request.</p> <p>Server Request Type. SAP ABAP server request type (for example Batch, Dialog, RFC).</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Layers. The Layers view displays a breakdown of the latency contribution for each layer in the server request. Another common drill down is View SAP ABAP Probes.</p>
Details Pane	<p>Shows metrics collected on SAP server requests including:</p> <p>R3 Confidence. The percentage of time latency data is explicitly attributable to this particular RFC.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

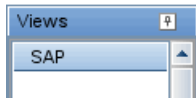
NetWeaver Requests View Description

The NetWeaver Requests view displays the average HTTP and P4 requests per second running on the SAP Web Application Server Java stack.

The following is an example of the NetWeaver Requests view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SAP** to open the SAP view group.

If the SAP view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SAP** and click **OK**. The SAP view group is displayed in the View bar.

Select the appropriate view from the SAP view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

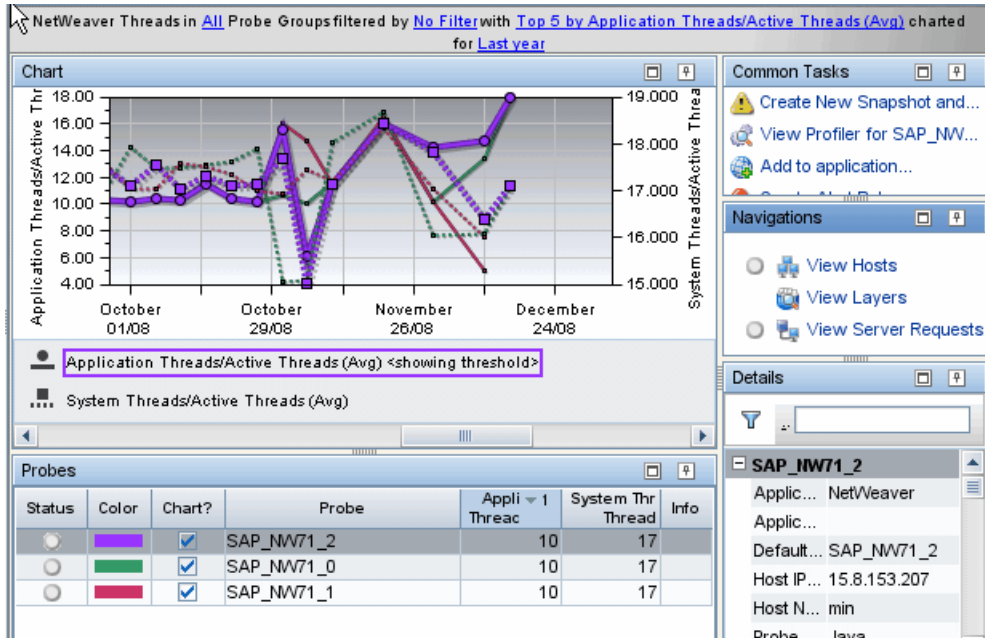
UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the top five NetWeaver requests with the highest average number HTTP and P4 requests per second during the previous five minutes are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average HTTP and P4 requests per second. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics represented.</p> <p>Probe Group. The name of the probe group to which the probe was assigned when it was installed.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Type of probe.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the NetWeaver metrics.</p> <p>HTTPRequests/sec. The number of HTTP requests per second running on the SAP application server Java Stack.</p> <p>P4 Requests/sec. The number of P4 requests per second running on the SAP application server Java stack.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs include View Layers and View Hosts.</p>
Details Pane	<p>Shows metrics collected by the probe for NetWeaver server requests.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

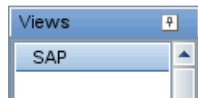
NetWeaver Threads View Description

The NetWeaver Threads view displays information about the active threads running on the SAP Web Application Server Java stack.

The following is an example of the NetWeaver Threads view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SAP** to open the SAP view group.

If the SAP view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SAP** and click **OK**. The SAP view group is displayed in the View bar.

Select the appropriate view from the SAP view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

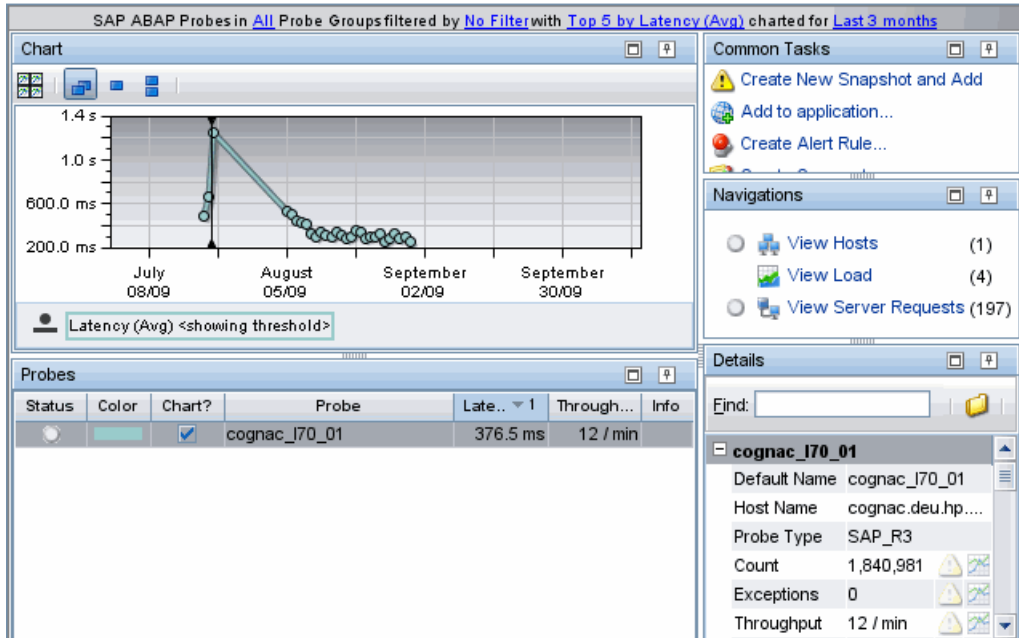
UI Elements	Description
Chart	<p>By default, the graph displays the top five NetWeaver threads with the highest average application threads/active threads during the previous five minutes.</p> <p>By default, the graph is based on data for all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average application threads/active threads.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics represented.</p> <p>Probe Group. The name of the probe group to which the probe was assigned when it was installed.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Type of probe</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the NetWeaver metrics.</p> <p>Application Threads/Active Threads. The number of active threads for this application.</p> <p>System Threads/Active Threads. The number of active system threads.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs include View Layers and View Hosts.</p>
Details Pane	<p>Shows metrics collected by the probe for the NetWeaver requests.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

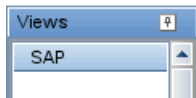
ABAP SAP Probes View Description

The ABAP SAP Probes view displays unique metrics for the ABAP instances.

The following is an example of the SAP ABAP Probes view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SAP** to open the SAP view group.

If the SAP view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SAP** and click **OK**. The SAP view group is displayed in the View bar.

Select the appropriate view from the SAP view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average latency, as a trend line. By default, the five ABAP probes that have the highest average latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average latency. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SAP ABAP application server.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. SAP ABAP probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the ABAP probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Latency % Over Threshold. The percentage over threshold for average latency.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Load and View Hosts.</p>
Details Pane	<p>Shows R3 metrics collected by SAP collector including the following:</p> <p>Bytes Requested. The average number of bytes requested by the application.</p> <p>Database Buffer Requests. The average number of database buffer requests.</p> <p>Database Calls. The average number of database calls made.</p> <p>Database Rows. The average number of database rows returned.</p> <p>Extended Memory. The average of the maximum extended memory used in dialog steps.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

SAP Probe Summary View Description

The Probe Summary view for SAP is a dashboard view that contains a summary of the SAP probe data.

The Summary dashboard view contains the following panes:

- Server Requests graph
- Hosts graph
- Load graph

► SAP ABAP Probes graph

To access	From the SAP view group, drill down from an SAP probe (right-click an SAP probe and select View Probe Summary).
------------------	---

SAP Probe Summary Graphs

The SAP Probe Summary view contains monitoring versions (charts/graphs) of the view displayed in the SAP view group as well as monitoring versions of the Hosts view and Load view.

UI Elements	Description
ABAP Server Requests graph	Graph of ABAP server requests.
Host graph	Hosts for the ABAP server requests.
Load graph	Load breakdown for ABAP server requests.
SAP ABAP Probes graph	Graph of ABAP probes.

30

Oracle Database Views

This chapter describes the specialized views for Oracle Database.

This chapter includes:

Concepts

- ▶ Oracle Database Overview on page 778

Reference

- ▶ Oracle Database View Group User Interface on page 780

Concepts

Oracle Database Overview

The **Oracle Database** view group displays Oracle performance data. When you install and configure the Diagnostics Collector to gather data from an Oracle Database, you define instances of Oracle to be monitored. Each one of these instances is represented as an Oracle probe, belonging to a probe group.

The Diagnostics Oracle collector runs a number of queries to retrieve the information displayed in the Oracle views. Basic database information such as the name, id and status is retrieved from the V\$DATABASE table along with the instance name, host name, version, startup time and status from the V\$INSTANCE table. Diagnostics uses SYSDATE from the DUAL table to help in time-synchronizing the collection of data between the collector and the database.

Instance level metrics are retrieved from the V\$SYSMETRIC table where INTSIZE_CSEC is greater than 3700. All the operating system statistics are collected from the V\$OSSTAT table. The time model stats are collected from the V\$SYS_TIME_MODEL table. And Diagnostics collects all wait events that are in the V\$SYSTEM_EVENT table and matches them up to their wait class from the DBA_HIST_EVENT_NAME table. Additional details on each table can be found on Oracle's website or by querying the different table names and viewing the results.

See the *HP Diagnostics Installation and Configuration Guide* chapter on "Installing the Diagnostics Collector" for details on configuring the Oracle collector.

The Oracle Database view group contains the following views:

- Summary view
- Oracle Probes view
- Wait Time view

For details on the user interface, see "Oracle Database View Group User Interface" on page 780.

Reference

Oracle Database View Group User Interface

This section includes:

- ▶ Oracle Database Summary View Description on page 780
- ▶ Oracle Probes View Description on page 785
- ▶ Wait Time View Description on page 790
- ▶ Oracle Probe Summary View Description on page 795

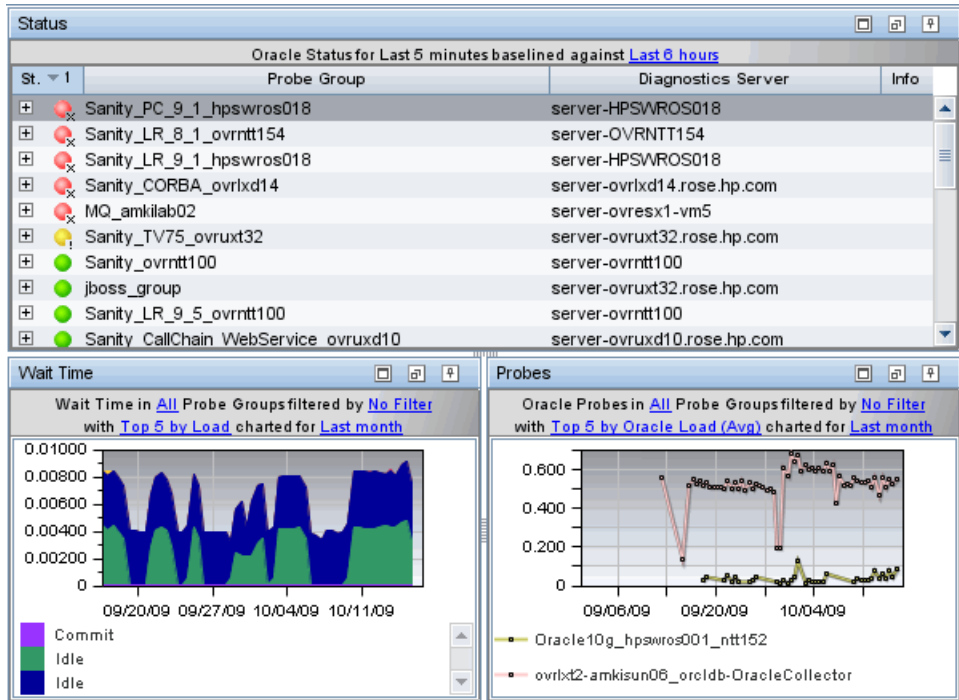
Oracle Database Summary View Description

The Oracle Database Summary view is a dashboard view that contains a summary of the Oracle 10g data displayed by Diagnostics.

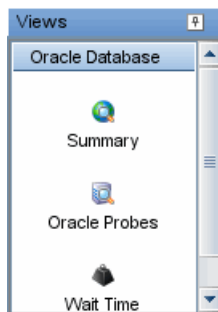
The Summary dashboard view contains the following panes:

- ▶ Status table
- ▶ Wait Time graph
- ▶ Probes graph

The following is an example of the Oracle Database Summary view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **Oracle Database** to open the Oracle Database view group.

If the Oracle Database view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **Oracle Database** and click **OK**. The Oracle Database view group is displayed in the View bar.

Select the appropriate view from the Oracle Database view group.

Important information	In the Oracle Database Summary view all Diagnostics probe groups are displayed, but within each probe group only Oracle probes are displayed.
See also	<ul style="list-style-type: none"> ➤ "Wait Time View Description" on page 790 ➤ "Oracle Probes View Description" on page 785

Status Table

The following user interface elements are included:

UI Elements	Description
Status Table	<p>A monitoring version of the standard Diagnostics Status view.</p> <p>All Diagnostics probe groups are displayed, but within each probe group only Oracle probes are displayed.</p> <p>For Oracle probes, the status is the only metric relevant for Oracle.</p> <p>For the Oracle hosts, the table displays CPU metrics. You can view unique performance metrics for the Oracle hosts by drilling down to the Diagnostics Hosts view.</p> <p>For information about the Status view, see "Status View Description" on page 520</p>

Oracle Database Summary Graphs

The Oracle Database Summary view contains monitoring versions (charts/ graphs) of the Wait Time view and the Probes view.

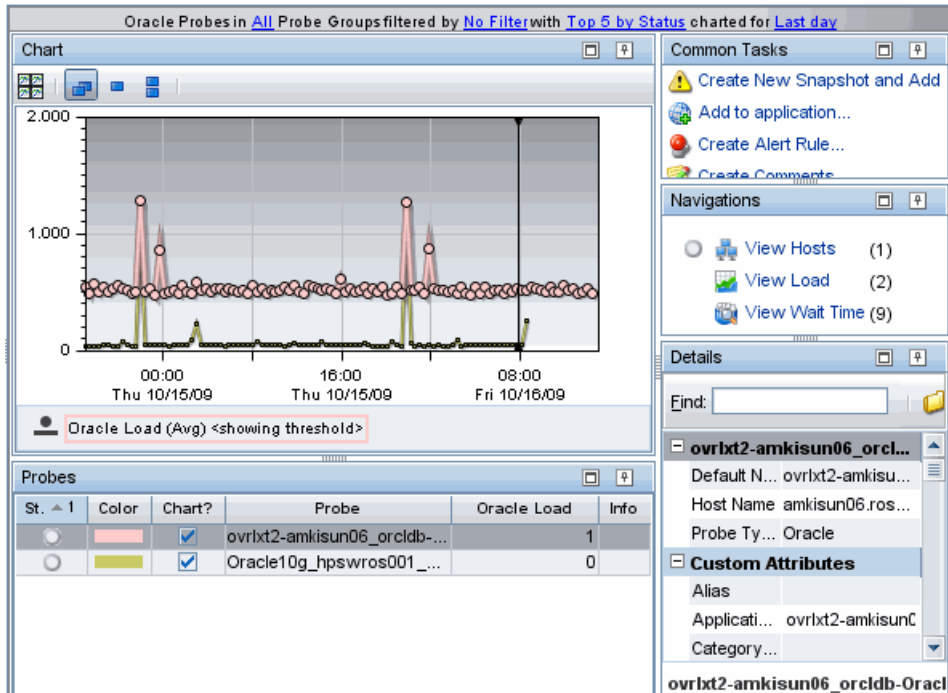
UI Elements	Description
Wait Time graph	<p>A monitoring version of the Oracle Wait Time view.</p> <p>By default, the graph displays the top five wait event classes with the highest Oracle load during the previous five minutes.</p> <p>Wait event classes represent specific types of events that Oracle has to wait for to continue processing.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the wait time.</p> <p>Layer Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Layer. The name of the wait event class (layer).</p> <p>Probe Group. The name of the probe group.</p> <p>The Layer Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Oracle Database Wait Time view.</p> <p>For information about the Wait Time view, see "Wait Time View Description" on page 790.</p>

UI Elements	Description
Probes graph	<p>By default, the graph displays the top five Oracle probes with the highest average Oracle load during the previous five minutes.</p> <p>By default, the graph is based on all Oracle probes for all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average load.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Probe. The name of the probe monitoring the Oracle application.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Oracle probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the Oracle Probes view.</p> <p>For information about the Oracle Probes view, see "Oracle Probes View Description" on page 785.</p>

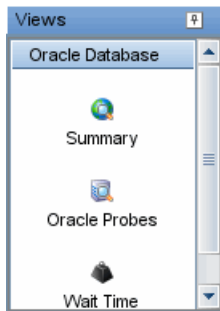
Oracle Probes View Description

The Oracle Probes view displays unique metrics for the Oracle 10g Database instances.

The following is an example of the Oracle Probes view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **Oracle Database** to open the Oracle Database view group.

If the Oracle Database view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **Oracle Database** and click **OK**. The Oracle Database view group is displayed in the View bar.

Select the appropriate view from the Oracle Database view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average Oracle load, as a trend line. By default, the five Oracle probes that have the highest average Oracle load during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Oracle load. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe monitoring the Oracle application server.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. Oracle probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

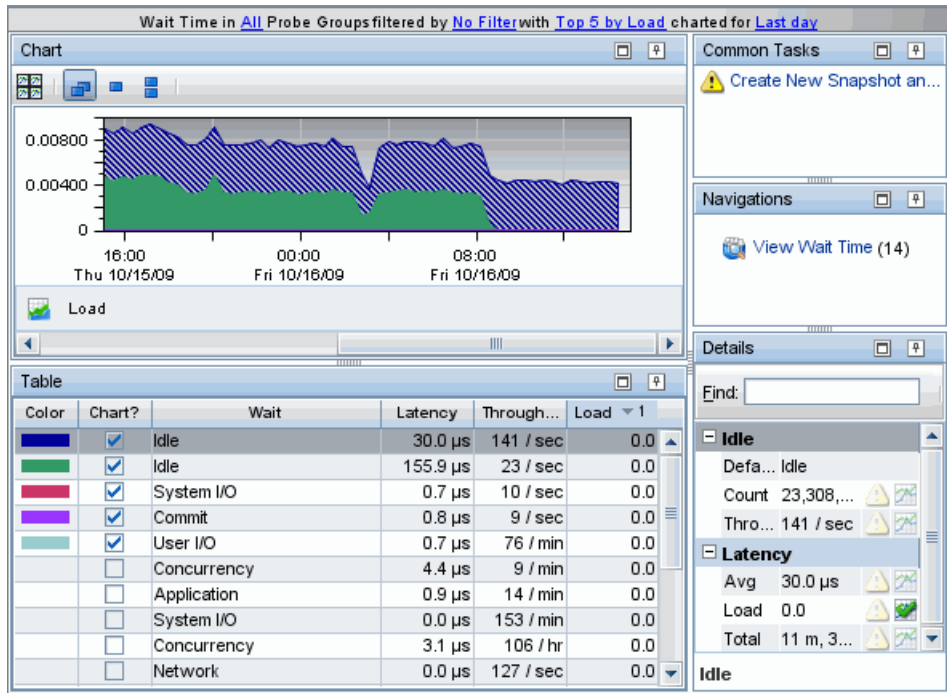
UI Elements	Description
Table	<p>Lists the Oracle probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Oracle Load. The average amount of work performed. Note that the sum of child layers can be greater than their parent due to Oracle time measurements not being mutually exclusive.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Wait Time. Other common drill downs are View Load, View Hosts, View Probe Summary.</p>
<p>Details Pane</p>	<p>Shows metrics collected for Oracle probes. An extensive number of metrics are collected, see "Oracle Database Overview" on page 778 for information on retrieval of metrics from Oracle.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

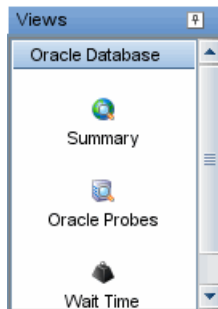
Wait Time View Description

The Wait Time view displays wait event classes for the Oracle 10g instances.

The following is an example of the Wait Time view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **Oracle Database** to open the Oracle Database view group.

If the Oracle Database view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **Oracle Database** and click **OK**. The Oracle Database view group is displayed in the View bar.

Select the appropriate view from the Oracle Database view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of load for each wait event class where processing has taken place in your application. The breakdown of the load across wait event classes is shown in a stacked area graph.</p> <p>By default, the five Oracle wait event classes that have the highest average Oracle load during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Oracle load.</p> <p>Layer Details tooltip for the stacked areas on the graph (or the Wait column in the table) displays the following detail data:</p> <p>Layer. The name of the layer.</p> <p>Probe Group. The name of the probe group.</p> <p>The Layer Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the Oracle wait event classes (layers) that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Wait. Type of Oracle wait time. Wait event classes represent specific types of events that Oracle has to wait for to continue processing.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Load. Load value for the wait event class (layer).</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
Common Tasks and Navigations	<p>Common tasks and actions relevant to the server requests displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Wait Time to see further wait event statistics for a particular layer.</p>
Details Pane	<p>Shows metrics collected on wait event classes.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

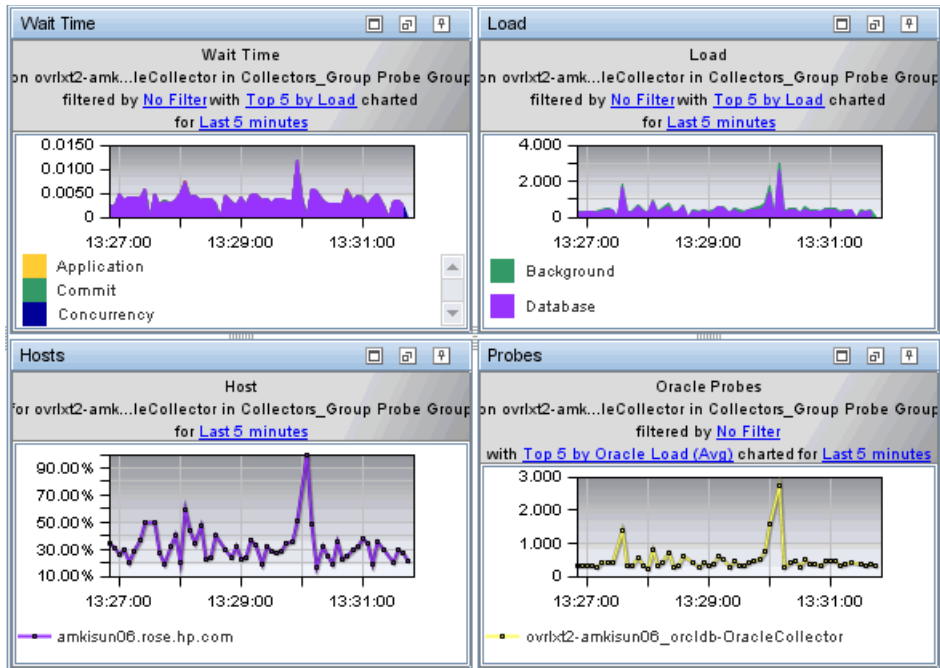
Oracle Probe Summary View Description

The Probe Summary view for Oracle is a dashboard view that contains a summary of the Oracle probe data.

The Summary dashboard view contains the following panes:

- Wait Time graph
- Load graph
- Hosts graph
- Probes graph

The following is an example of the Probe Summary view for Oracle probes.



To access	From the Oracle Data base view group, drill down from an Oracle probe (right-click an Oracle probe and select View Probe Summary).
See also	<ul style="list-style-type: none"> ➤ "Wait Time View Description" on page 790 ➤ "Load View Description" on page 511 ➤ "Hosts View" on page 369 ➤ "Oracle Probes View Description" on page 785

Oracle Probe Summary Graphs

The Oracle Probe Summary view contains monitoring versions (charts/graphs) of the view displayed in the Oracle Database view group as well as monitoring versions of the Load view and Host view.

UI Elements	Description
Wait Time graph	<p>A monitoring version of the Wait Time view.</p> <p>By default, the graph displays the load for the top five wait event classes for the selected Oracle probe. The breakdown across wait event classes is shown in a stacked area graph.</p> <p>The x-axis shows chronological time. The y-axis shows the average Oracle load for a wait event.</p> <p>If you double-click the graph the pre-defined workflow displays the Wait Time view.</p> <p>For information about the Wait Time view, see "Wait Time View Description" on page 790.</p>
Load graph	<p>A monitoring version of the Load view.</p> <p>By default, the graph displays the load for the top five layers for the selected Oracle probe. The breakdown across layers is shown in a stacked area graph.</p> <p>The x-axis shows chronological time. The y-axis shows the average load for a layer.</p> <p>If you double-click the graph the pre-defined workflow displays the Load view.</p> <p>For information about the Load view, see "Load View Description" on page 511.</p>

UI Elements	Description
<p>Host graph</p>	<p>A monitoring version of the Host view.</p> <p>By default, the graph displays the Average CPU used on the system hosting the Oracle collector.</p> <p>The x-axis shows chronological time. The y-axis shows CPU (Avg).</p> <p>If you double-click the graph the pre-defined workflow displays the Hosts view.</p> <p>For information about the Hosts view, see "Hosts View" on page 369.</p>
<p>Oracle Probes graph</p>	<p>A monitoring version of the Oracle Probes view.</p> <p>By default, the graph displays the average load for the selected Oracle probe.</p> <p>The x-axis shows chronological time. The y-axis shows Oracle load (Avg).</p> <p>If you double-click the graph the pre-defined workflow displays the Oracle Probes view.</p> <p>For information on the Oracle Probes view, see "Oracle Probes View Description" on page 785.</p>

31

SQL Server Database Views

This chapter describes the specialized views for SQL Server Database.

This chapter includes:

Concepts

- ▶ SQL Server Database Overview on page 800

Reference

- ▶ SQL Server Database View Group User Interface on page 802

Concepts

SQL Server Database Overview

The **SQL Server Database** view group displays SQL Server performance data. When you install and configure the SQL Server Database Collector to gather data from an SQL Server database, you define instances of SQL Server to be monitored. Each one of these instances is represented as an SQL Server probe, belonging to a probe group.

Diagnostics collector automatically collects data for all SQL Server databases in the instance. If you want to exclude some of these databases from collection (for example system databases), you can specify a comma-separated list in the **exclude.db.list** property in the `<collector_install_dir>\etc\sqlserver.properties` file. See the *HP Diagnostics Installation and Configuration Guide* chapter on "Installing the Diagnostics Collector" for details on configuring the SQL Server collector.

When you have n databases in your instance you actually will have $n+1$ probes; an extra probe for the totals of the instance which includes metrics such as wait events. The extra probe uses probeName, the probes for each database use probeName_databaseName.

The Diagnostics SQL Sever collector runs a number of queries to retrieve the information displayed in the SQL Server Database views. Basic database information such as the name, id, status and the list of databases is retrieved from the master.dbo.sysdatabases table. Diagnostics uses the GETDATE() function that SQL Server provides to help in time-synchronizing the collection of data between the collector and the database.

Instance level metrics are retrieved from the sys.dm_os_performance_counters table. The operating system statistics are collected from the sys.dm_os_sys_info table. The wait events are collected from the sys.dm_os_wait_stats table. Version information is collected from the sys.servers table.

Additional details on each table can be found on Microsoft's SQL Server website or by querying the different table names and viewing the results.

See the *HP Diagnostics Installation and Configuration Guide* chapter on "Installing the Diagnostics Collector" for details on configuring the SQL Server collector.

The SQL Server Database view group contains the following views:

- Summary view
- SQL Server Probes view
- Wait Time view

For details on the user interface, see "SQL Server Database View Group User Interface" on page 802.

Reference

SQL Server Database View Group User Interface

This section includes:

- ▶ SQL Server Database Summary View Description on page 802
- ▶ SQL Server Probes View Description on page 808
- ▶ Wait Time View Description on page 813
- ▶ SQL Server Probe Summary View Description on page 817

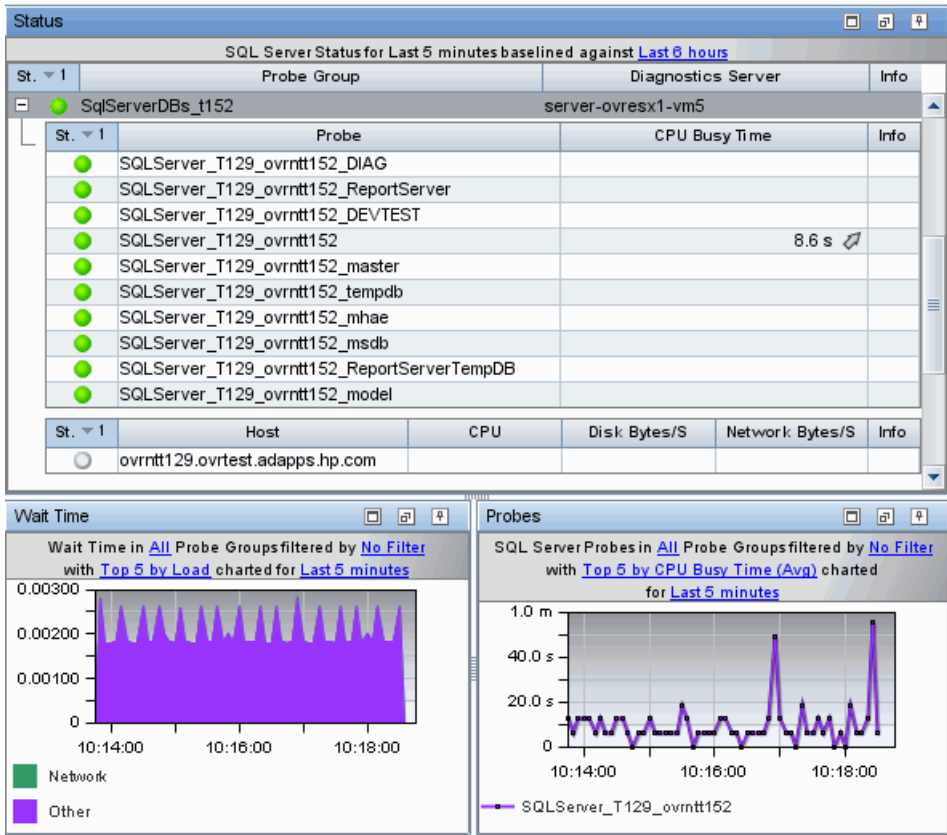
SQL Server Database Summary View Description

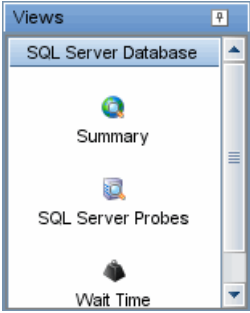
The Summary view is a dashboard view that contains a summary of the SQL Server data displayed by Diagnostics.

The Summary dashboard view contains the following panes:

- ▶ Status table
- ▶ Wait Time graph
- ▶ Probes graph

The following is an example of the SQL Database Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click SQL Server Database to open the SQL Server Database view group.</p> <p>If the SQL Server Database view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select SQL Server Database and click OK. The SQL Server Database view group is displayed in the View bar.</p> <p>Select the appropriate view from the SQL Server Database view group.</p>
<p>Important information</p>	<p>In the SQL Server Database Summary view all Diagnostics probe groups are displayed, but within each probe group only SQL Server probes are displayed.</p>
<p>Relevant tasks</p>	<p>Diagnostics collector automatically collects data for all SQL Server databases in the instance. If you want to exclude some of these databases from collection (for example system databases), you can specify a comma-separated list in the exclude.db.list property in the <code><collector_install_dir>\etc\sqlserver.properties</code> file. See the <i>HP Diagnostics Installation and Configuration Guide</i> chapter on "Installing the Diagnostics Collector" for details on configuring the SQL Server collector.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "Status View Description" on page 520 ➤ "Wait Time View Description" on page 813 ➤ "SQL Server Probes View Description" on page 808

Status Table

The following user interface elements are included:

UI Elements	Description
<p>Status Table</p>	<p>A monitoring version of the standard Diagnostics Status view.</p> <p>All Diagnostics probe groups are displayed, but within each probe group only SQL Server probes are displayed.</p> <p>For SQL Server probes, the status and CPU Busy Time metrics are relevant for SQL Server.</p> <p>CPU Busy Time shows a metric trend indicator. The arrows indicate the result of the comparison of the baseline performance to the current performance. The title of the status table contains the baseline control. You use this control to set the time that Diagnostics uses as the baseline performance level with which to compare the current performance level.</p> <p>Host metrics are collected and displayed in the hosts status table only if a Java probe is also configured on the host system.</p> <p>For the SQL Server hosts, the table displays the following:</p> <p>CPU. Average CPU utilization.</p> <p>Disk Bytes/S. The average number of bytes read and written per second.</p> <p>Network Bytes/S. The average number of bytes sent and received per second.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>You can view unique performance metrics for the SQL Server hosts by drilling down to the Diagnostics Hosts view.</p> <p>For information about the Status view, see "Status View Description" on page 520.</p>

SQL Server Database Summary Graphs

The SQL Server Database Summary view contains monitoring versions (charts/graphs) of the Wait Time view and the Probes view.

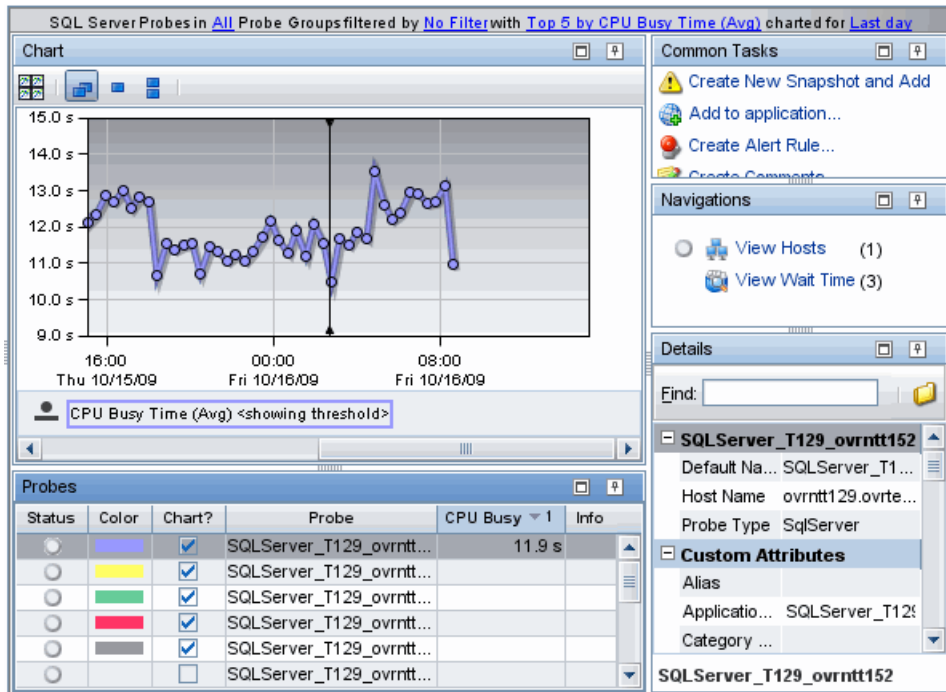
UI Elements	Description
Wait Time graph	<p>A monitoring version of the SQL Server Wait Time view.</p> <p>By default, the graph displays the top five wait event classes with the highest SQL Server load during the previous five minutes.</p> <p>Wait event classes represent specific types of events that SQL Server has to wait for to continue processing.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the wait time.</p> <p>Layer Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Layer. The name of the wait event class (layer).</p> <p>Probe Group. The name of the probe group.</p> <p>The Layer Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the SQL Server Database Wait Time view.</p> <p>For information about the Wait Time view, see "Wait Time View Description" on page 813.</p>

UI Elements	Description
Probes graph	<p>A monitoring version of the SQL Server Probes view.</p> <p>By default, the graph displays the top five SQL Server probes with the highest average CPU Busy Time during the previous five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the CPU Busy Time.</p> <p>Probe Details tooltip for the trend line on the graph displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SQL Server application.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. SQLServer probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p> <p>If you double-click the graph, the pre-defined workflow displays the SQL Server Probes view.</p> <p>For information about the SQL Server Probes view, see "SQL Server Probes View Description" on page 808.</p>

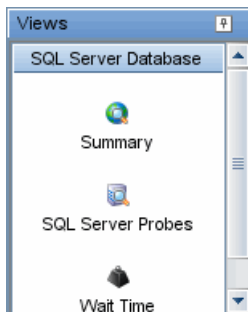
SQL Server Probes View Description

The SQL Server Probes view displays unique metrics for the SQL Server Database instances.

The following is an example of the SQL Server Probes view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SQL Server Database** to open the SQL Server Database view group.

If the SQL Server Database view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SQL Server Database** and click **OK**. The SQL Server Database view group is displayed in the View bar.

Select the appropriate view from the SQL Server Database view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

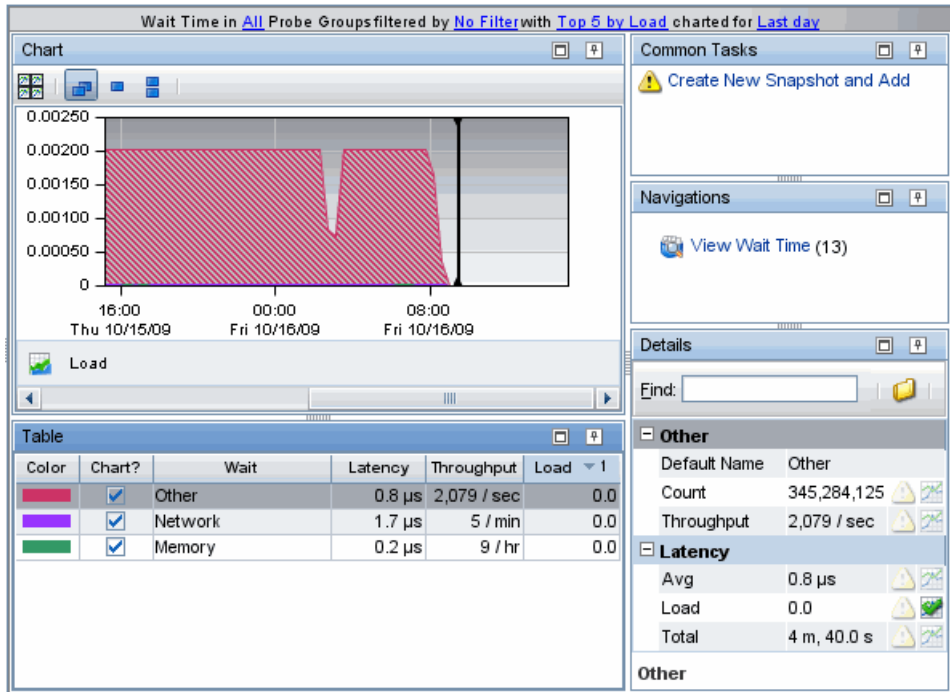
The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average CPU Busy Time, as a trend line. By default, the five SQL Server probes that have the highest average CPU Busy Time during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average CPU Busy Time. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe monitoring the SQL Server application server.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. SQLServer probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

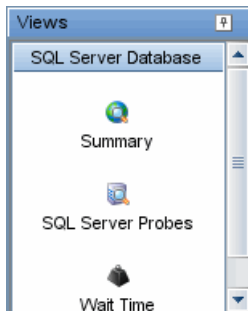
UI Elements	Description
<p>Table</p>	<p>Lists the SQL Server probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>CPU Busy Time. The average time the CPU is busy based on the SQL Server time model.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Wait Time. Another common drill down is View Hosts.</p>
<p>Details Pane</p>	<p>Shows metrics collected for SQL Server probes. An extensive number of metrics is collected, see "SQL Server Database Overview" on page 800 for information on retrieval of metrics from SQL Server.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Wait Time View Description

The Wait Time view displays wait event classes for the SQL Server instances. The following is an example of the Wait Time view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **SQL Server Database** to open the SQL Server Database view group.

If the SQL Server Database view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **SQL Server Database** and click **OK**. The SQL Server Database view group is displayed in the View bar.

Select the appropriate view from the SQL Server Database view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of load for each wait event class where processing has taken place in your application. The breakdown of the load across wait event classes is shown in a stacked area graph.</p> <p>By default, the five SQL Server wait event classes that have the highest average load during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average SQL Server load.</p> <p>Layer Details tooltip for the stacked areas on the graph (or the Wait column in the table) displays the following detail data:</p> <p>Layer. The name of the layer.</p> <p>Probe Group. The name of the probe group.</p> <p>The Layer Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
<p>Table</p>	<p>Lists the SQL Server wait event classes (layers) that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Wait. Type of SQL Server wait time. Wait event classes represent specific types of events that SQL Server has to wait for to continue processing.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the selected time period.</p> <p>Load. Load value for the wait event class (layer).</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the wait event classes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Wait Time to see further wait event statistics for a particular layer.</p>
<p>Details Pane</p>	<p>Shows metrics collected on wait event classes.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

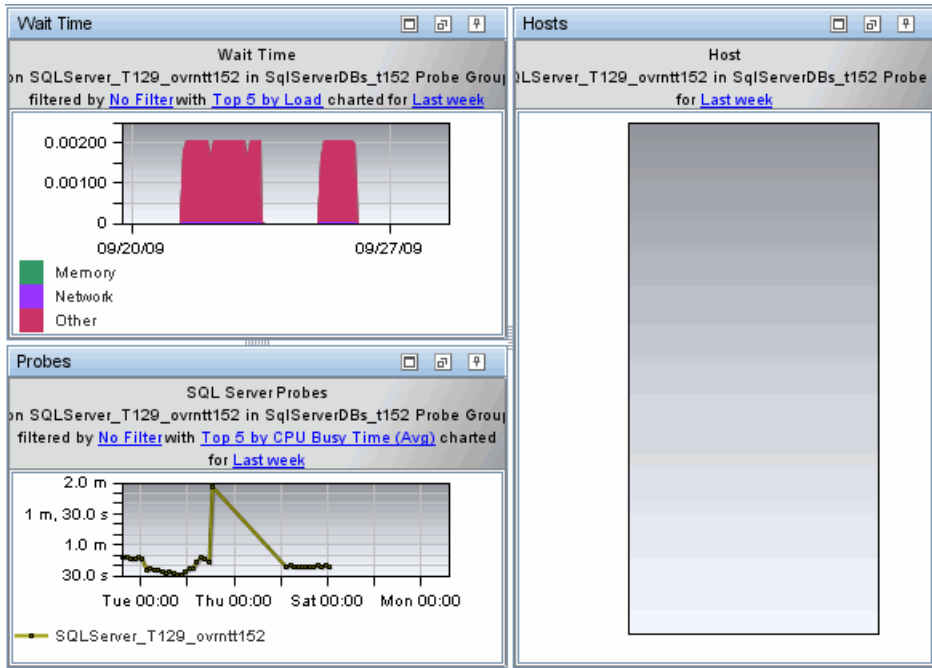
SQL Server Probe Summary View Description

The Probe Summary view for SQL Server is a dashboard view that contains a summary of the SQL Server probe data.

The Summary dashboard view contains the following panes:

- Wait Time graph
- Hosts graph
- SQL Server Probes graph

The following is an example of the Probe Summary view for SQL Server probes.



To access	From the SQL Server Database view group, drill down from an SQL Server probe (right-click an SQL Server probe and select View Probe Summary).
See also	<ul style="list-style-type: none"> ➤ "Wait Time View Description" on page 813 ➤ "Hosts View" on page 369 ➤ "SQL Server Probes View Description" on page 808

SQL Server Probe Summary Graphs

The SQL Server Probe Summary view contains monitoring versions (charts/graphs) of the view displayed in the SQL Server Database view group as well as a monitoring version of the Host view.

UI Elements	Description
Wait Time graph	<p>A monitoring version of the Wait Time view.</p> <p>By default, the graph displays load for the top five wait event classes for the probe. The breakdown of the load across wait event classes is shown in a stacked area graph. The x-axis shows chronological time. The y-axis shows the average SQL Server load for a wait event.</p> <p>If you double-click the graph the pre-defined workflow displays the Wait Time view.</p> <p>For information about the Wait Time view, see "Wait Time View Description" on page 813.</p>
Host graph	<p>A monitoring version of the Host view.</p> <p>By default, the graph displays the Average CPU used on the system hosting the SQL Server collector.</p> <p>The x-axis shows chronological time. The y-axis shows CPU (Avg).</p> <p>If you double-click the graph the pre-defined workflow displays the Hosts view.</p> <p>For information about the Hosts view, see "Hosts View" on page 369.</p>
SQL Server Probes graph	<p>A monitoring version of the SQL Server Probes view.</p> <p>By default, the graph displays the average load for the selected SQL Server probe.</p> <p>The x-axis shows chronological time. The y-axis shows SQL Server load (Avg).</p> <p>If you double-click the graph the pre-defined workflow displays the SQL Server Probes view.</p> <p>For information on the SQL Server Probes view, see "SQL Server Probes View Description" on page 808.</p>

32

MQ Views

This chapter describes the specialized views for WebSphere MQ.

This chapter includes:

Concepts

- ▶ MQ Overview on page 822

Reference

- ▶ MQ View Group User Interface on page 823

Concepts

MQ Overview

The MQ view group displays WebSphere MQ performance data. IBM WebSphere MQ is a system for messaging across multiple platforms.

The queue manager is the primary component of an IBM WebSphere MQ installation. It provides a logical container for the message queue, which is an object that stores messages in an application. The queue manager is also responsible for transferring data to other queue managers by using message channels. Communication between queue managers relies on a separate program, called a channel. The channel runs on the same host as the queue manager. It handles sending and receiving data over the network.

The Diagnostics Collector is used to gather MQ performance metrics. See the *HP Diagnostics Installation and Configuration Guide* chapter on "Installing the Diagnostics Collector" for details on configuring the MQ collector.

The Diagnostics MQ view group contains the following views:

- ▶ MQ Channels
- ▶ MQ Queues
- ▶ Queue Managers

For details on the user interface, see "MQ View Group User Interface" on page 823.

Reference

MQ View Group User Interface

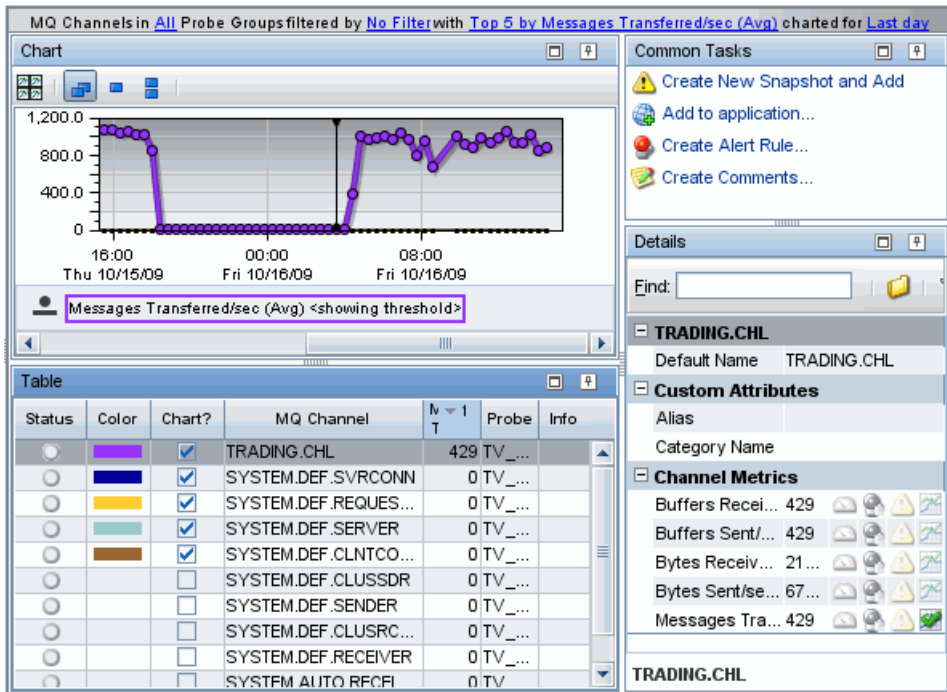
This section includes:

- ▶ [MQ Channels View Description on page 823](#)
- ▶ [MQ Queues View Description on page 828](#)
- ▶ [MQ Queue Managers View Description on page 833](#)

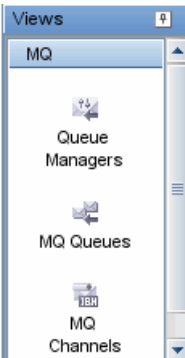
MQ Channels View Description

The MQ Channels view displays performance metrics for the channels in your MQ environment.

The following is an example of the MQ Channels view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **MQ** to open the MQ view group.

If the MQ view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **MQ** and click **OK**. The MQ view group is displayed in the View bar.

Select the appropriate view from the MQ view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average messages transferred/sec, as a trend line. By default, the five MQ channels that have the highest average number of messages transferred during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis of the graph shows the average number of messages transferred per second.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>MQ Channel Details tooltip for the trend line on the graph (or the MQ Channel column in the table) displays the following detail data:</p> <p>MQ Channel. The name of the MQ channel.</p> <p>Probe. The name of the probe monitoring the MQ channels.</p> <p>Probe Group. The name of the probe group containing the probe.</p> <p>The MQ Channel Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

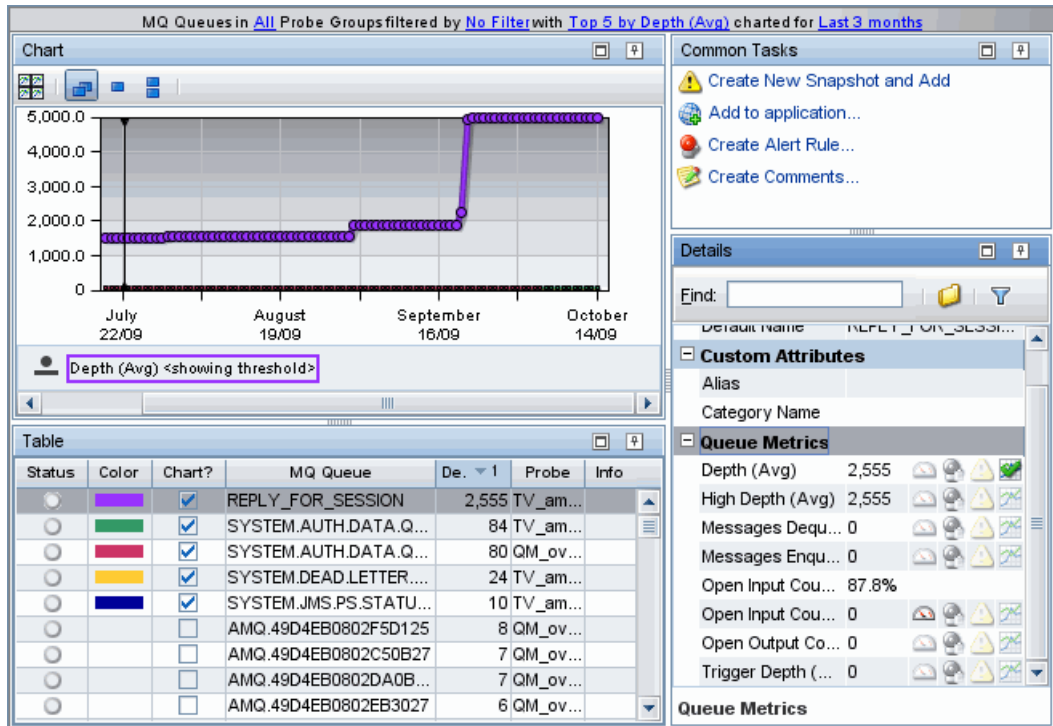
UI Elements	Description
Table	<p>Lists the MQ channels that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the MQ channel. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>MQ Channel. The name of the MQ channel.</p> <p>Messages Transferred/Sec. The rate of messages transferred in this channel.</p> <p>Probe. The name of the probe.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

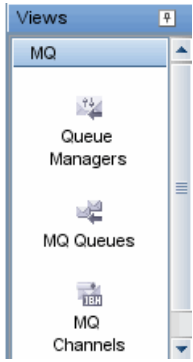
UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the data displayed in this view.
Details Pane	<p>Shows metrics collected on MQ channels, including the following:</p> <p>Buffers Received/sec. The rate of buffers received by this channel.</p> <p>Buffers Sent/sec. The rate of buffers sent from this channel.</p> <p>Bytes Received/sec. The rate of bytes received by this channel.</p> <p>Bytes Sent/sec. The rate of bytes sent from this channel.</p> <p>Messages Transferred/sec (Avg). The rate of messages transferred in this channel.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

MQ Queues View Description

The MQ Queues view displays performance metrics for the queues in your MQ environment.

The following is an example of the MQ Queues view.



<p>To access</p>  <p>The screenshot shows a 'Views' pane with a search bar at the top. Below the search bar, 'MQ' is selected and highlighted in blue. Underneath, there are three expandable categories: 'Queue Managers' (with a folder icon), 'MQ Queues' (with a document icon), and 'MQ Channels' (with a document icon).</p>	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click MQ to open the MQ view group.</p> <p>If the MQ view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select MQ and click OK. The MQ view group is displayed in the View bar.</p> <p>Select the appropriate view from the MQ view group.</p>
<p>Important information</p>	<p>You can specify which types of queues are included in the metrics. By default, all types of queues are included. For more information about how to specify the queues included in the metrics, refer to the <i>HP Diagnostics Installation and Configuration Guide</i>.</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average queue depth, as a trend line. By default, the five MQ queues that have the highest average depth during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis of the graph shows the average queue depth.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>MQ Queue Details tooltip for the trend line on the graph (or the MQ Queue column in the table) displays the following detail data:</p> <p>MQ Queue The name of the MQ queue.</p> <p>Probe. The name of the probe monitoring the MQ queues.</p> <p>Probe Group. The name of the probe group containing the probe.</p> <p>The MQ Queue Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

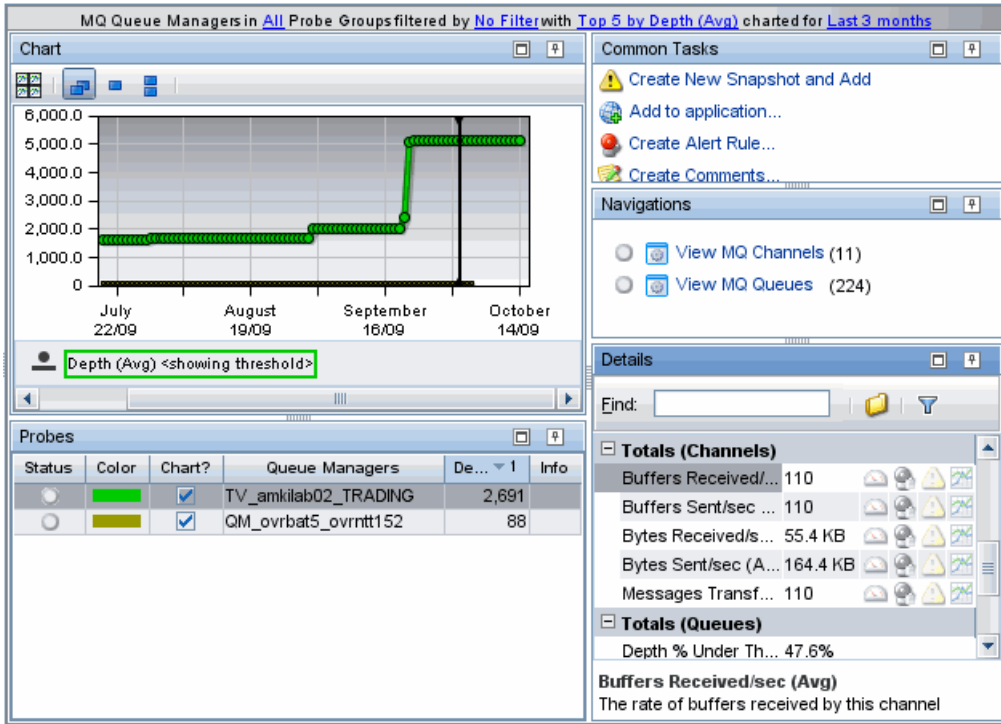
UI Elements	Description
Table	<p>Lists the MQ queues that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the MQ queue. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>MQ queue. The name of the MQ queue.</p> <p>Depth. The depth of the queue.</p> <p>Probe. The name of the probe.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	Common tasks and actions relevant to the data displayed in this view.
Details Pane	<p>Shows metrics collected on MQ queues, including the following:</p> <p>Depth. The current depth of the queue.</p> <p>High Depth. The maximum number of messages on the queue during the last interval. (The interval is the time between queries. It can be configured in mq.properties. The default is 20 seconds.)</p> <p>Messages Dequeued/sec. The number of messages dequeued per second.</p> <p>Messages Enqueued/sec. The number of messages enqueued per second.</p> <p>Open Input Count % Over Threshold.</p> <p>Open Input Count. The total number of handles that are currently valid for removing messages from the queue.</p> <p>Open Output Count. The total number of handles that are currently valid for adding messages to the queue.</p> <p>Trigger Depth. The number of messages that have to be on the queue before a trigger message is written when trigger type is set to MQC.MQTT_DEPTH.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

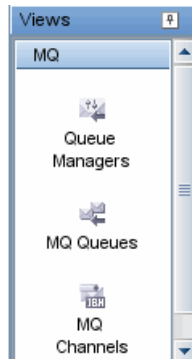
MQ Queue Managers View Description

The MQ Queue Managers view displays performance metrics for the queue managers in your MQ environment.

The following is an example of the MQ Queue Managers view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **MQ** to open the MQ view group.

If the MQ view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **MQ** and click **OK**. The MQ view group is displayed in the View bar.

Select the appropriate view from the MQ view group.

Important information

The metrics on this screen are an aggregation of the queue and channel metrics shown in the MQ Channels and MQ Queues view.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average queue depth, as a trend line. By default, the five MQ queue managers that have the highest average depth during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis of the graph shows the average queue depth.</p> <p>Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Queue Managers column in the table) displays the following detail data:</p> <p>Probe. The name of the probe monitoring the MQ channels and queues.</p> <p>Probe Group. The name of the probe group containing the probe.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. MQ probe type.</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the Queue Managers that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the queue managers. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Queue Manager. The name of the Queue Manager.</p> <p>Depth. The depth of the queue.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the data displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View MQ Queues. Another common drill down is View MQ Channels.</p>
<p>Details Pane</p>	<p>Shows metrics collected on Queue Managers.</p> <p>The metrics are an aggregation of the queue and channel metrics shown in the MQ Channels and MQ Queues view. They are grouped in two categories: Totals (Channels) and Totals (Queues).</p> <p>For example, under the Total (Channels) category, you can see a metric called Messages Transferred/sec. This is the total rate of messages transferred for the entire MQ environment (that is, for all the channels that are shown under the MQ Channels view).</p> <p>For example, under the Total (Queues) category, you can see a metric called Depth. This is the total depth for all the queues. It is obtained by adding up the individual queue depths of the queues shown under the MQ Queues view.</p> <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

33

BEA WebLogic Views

This chapter describes the specialized views for BEA WebLogic.

This chapter includes:

Concepts

- ▶ BEA WebLogic Overview on page 840

Reference

- ▶ BEA WebLogic View Group User Interface on page 841

Concepts

BEA WebLogic Overview

The **BEA WebLogic** view group displays performance data for BEA WebLogic application servers.

The Diagnostics BEA WebLogic view group contains the following views:

- ▶ BEA WebLogic Summary view
- ▶ EJB Pooled Resource Contention view
- ▶ JDBC Connection Status view
- ▶ JDBC Resource Contention view
- ▶ Server Threads view

For details on the user interface, see "BEA WebLogic View Group User Interface" on page 841.

Reference

BEA WebLogic View Group User Interface

This section includes:

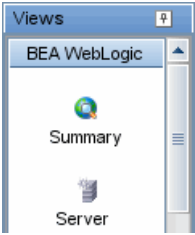
- [BEA WebLogic Summary View Description on page 842](#)
- [EJB Pooled Resource Contention View Description on page 848](#)
- [JDBC Connection Status View Description on page 853](#)
- [JDBC Resource Contention View Description on page 858](#)
- [Server Threads View Description on page 863](#)

BEA WebLogic Summary View Description

The Summary view is a dashboard view containing the standard Diagnostics Status view along with monitoring versions of the standard Diagnostics Server Requests view and some of the views displayed in the BEA WebLogic view group.

The following is an example of the BEA WebLogic Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click BEA WebLogic to open the BEA WebLogic view group.</p> <p>If the BEA WebLogic view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select BEA WebLogic and click OK. The BEA WebLogic view group is displayed in the View bar.</p> <p>Select the appropriate view from the BEA WebLogic view group.</p>
<p>Important information</p>	<p>The view title in each graph or table describes the view using the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in “Working With View Filters”.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "JDBC Resource Contention View Description" on page 858 ➤ "JDBC Connection Status View Description" on page 853 ➤ "Server Threads View Description" on page 863 ➤ "EJB Pooled Resource Contention View Description" on page 848 ➤ "Server Requests View Description" on page 389 ➤ "Status View Description" on page 520

BEA WebLogic Graphs

The BEA WebLogic Summary view contains monitoring versions (charts/graphs) of some of the views displayed in the BEA WebLogic view group as well as a monitoring version of the Server Requests view.

UI Elements	Description
<p>JDBC Connection Status graph</p>	<p>A monitoring version of the JDBC Connections Status view.</p> <p>By default, the graph displays the top five probes by JDBC average Active Connections. during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average connections.</p> <p>If you double-click the graph the pre-defined workflow displays the JDBC Connection Status view.</p> <p>For information about the JDBC Connection Status view, see "JDBC Resource Contention View Description" on page 858.</p>
<p>JDBC Resource Contention graph</p>	<p>A monitoring version of the JDBC Resource Contention view.</p> <p>By default, the graph displays the top five probes by average JDBC Wait Seconds High during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Wait Seconds High.</p> <p>If you double-click the graph the pre-defined workflow displays the JDBC Resource Contention view.</p> <p>For information about the JDBC Resource Contention view, see "JDBC Connection Status View Description" on page 853.</p>

UI Elements	Description
Server Threads graph	<p>A monitoring version of the Server Threads view.</p> <p>By default, the graph displays the top five probes by average Execute Queues Pending Requests during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Execute Queues Pending Requests.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Threads view.</p> <p>For information about the Server Threads view, see "Server Threads View Description" on page 863.</p>

UI Elements	Description
EJB Pooled Resource Contention graph	<p>A monitoring version of the EJB Pooled Resource Contention view.</p> <p>By default, the graph displays the top five probes by average EJB-Pool Current Waiters during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average EJB-Pool Current Waiters.</p> <p>If you double-click the graph the pre-defined workflow displays the EJB Pooled Resource Contention view.</p> <p>For information about the EJB Pooled Resource Contention view, see "EJB Pooled Resource Contention View Description" on page 848.</p>
Server Requests graph	<p>A monitoring version of the Server Requests view.</p> <p>By default, the graph displays the top five server requests by Latency % Over Threshold during the last five minutes.</p> <p>By default, the graph is based on all server requests and all probe groups. You can filter the data by individual server request or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the Latency % Over Threshold.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Requests view.</p> <p>For information about the Server Requests view, see "Server Requests View Description" on page 389.</p>

Status Table

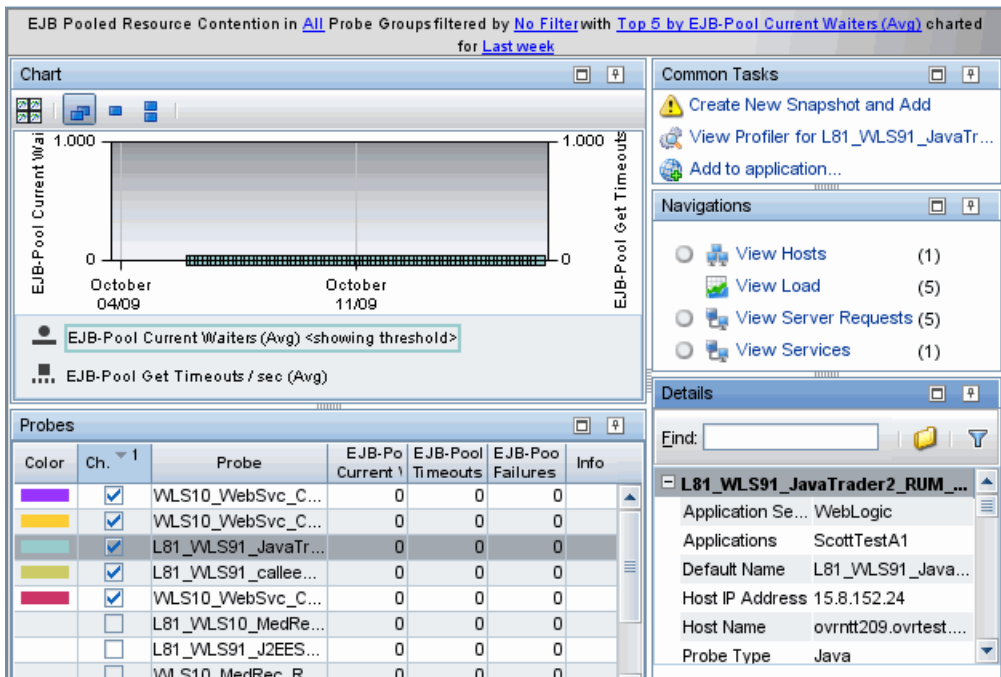
The following user interface elements are included:

UI Elements	Description
Status table	<p>A monitoring version of the Diagnostics Status view.</p> <p>All Diagnostics probe groups are displayed in the status table, but within each probe group, only BEA WebLogic probes are displayed.</p> <p>By default, the first column in the Status table presents the status for each probe group, probe, and host. The status reflects a comparison with thresholds.</p> <p>For BEA WebLogic probes, you see the following metrics:</p> <p>Probe. The name of the probe.</p> <p>VM Heap Used. The average amount of heap used by the applications being monitored by the probe.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Host metrics include the following:</p> <p>Host. The host system on which the probe is running.</p> <p>CPU. Average CPU utilization.</p> <p>Disk Bytes/S. The average number of bytes read and written per second.</p> <p>Network Bytes/S. The average number of bytes sent and received per second.</p> <p>PageIn/Second. The average number of virtual memory pages swapped in per second.</p> <p>Memory. The average amount of memory in use.</p> <p>Info. Indicates if an alert rule or comment is defined.</p> <p>For information the Status view, see "Status View Description" on page 520.</p>

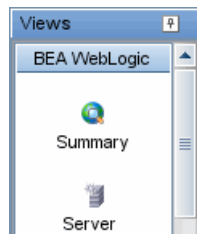
EJB Pooled Resource Contention View Description

The EJB Pooled Resource Contention view displays performance metrics for contention of EJB pools.

The following is an example of the EJB Pooled Resource Contention view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **BEA WebLogic** to open the BEA WebLogic view group.

If the BEA WebLogic view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **BEA WebLogic** and click **OK**. The BEA WebLogic view group is displayed in the View bar.

Select the appropriate view from the BEA WebLogic view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average EJB-Pool Current Waiters, as a trend line. By default, the five probes that have the highest average EJB-Pool Current Waiters during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average EJB-Pool Current Waiters. Average EJB-Pool Get Timeouts per second is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

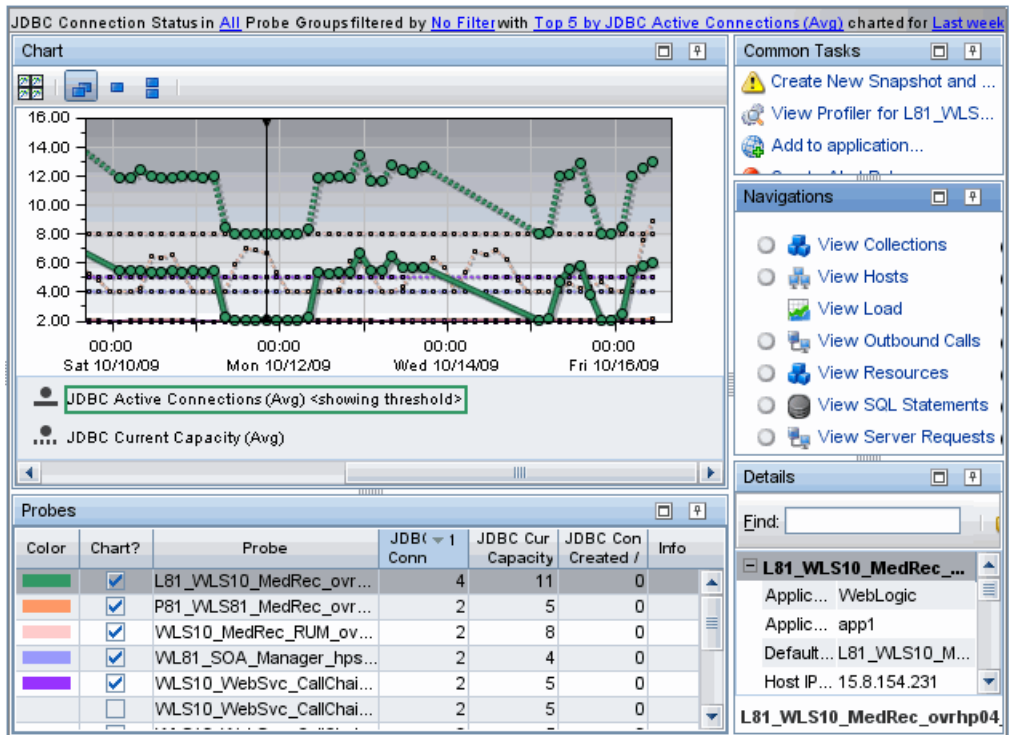
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>EJB-Pool Current Waiters. The number of EJB pool requests waiting on the pool.</p> <p>EJB-Pool Get Timeouts/sec. The number of EJB pool timeouts per second.</p> <p>EJB-Pool Get Failures/sec. The number of EJB pool access failures per second.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
Details Pane	<p>Shows metrics collected for WebLogic. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ➤ Classes ➤ EJB ➤ Execute Queues ➤ GC ➤ JDBC ➤ JDBC<instances> ➤ JMS ➤ JRockit ➤ JTA ➤ Servlets ➤ Thread Pool ➤ Threads ➤ Web Services ➤ Web Sessions <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

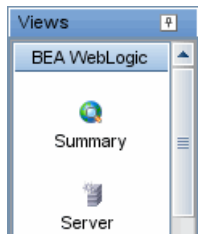
JDBC Connection Status View Description

The JDBC Connection Status view displays performance metrics for JDBC usage.

The following is an example of the JDBC Connection Status view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **BEA WebLogic** to open the BEA WebLogic view group.

If the BEA WebLogic view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **BEA WebLogic** and click **OK**. The BEA WebLogic view group is displayed in the View bar.

Select the appropriate view from the BEA WebLogic view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average JDBC Active Connections, as a trend line. By default, the five probes that have the highest average JDBC Active Connections during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Active Connections. Average JDBC Current Capacity is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

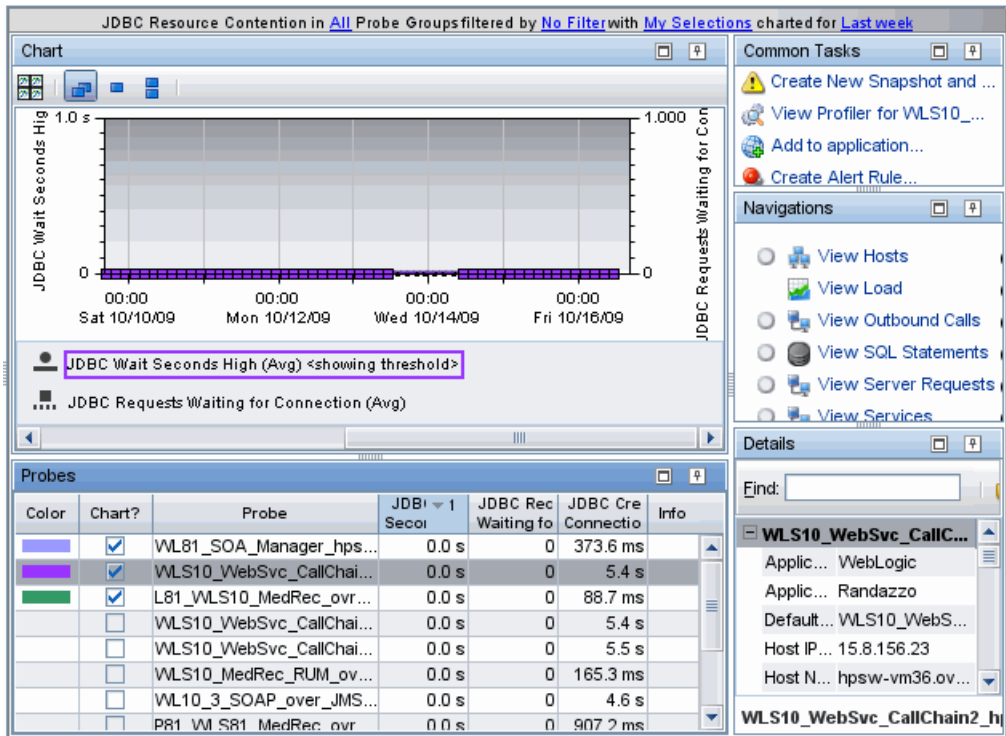
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>JDBC Active Connections. The number of active JDBC database connections.</p> <p>JDBC Current Capacity. The current JDBC database connection capacity.</p> <p>JDBC Connections Created/sec. The number of JDBC connections created per second.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
Details Pane	<ul style="list-style-type: none"> ➤ Shows metrics collected for WebLogic. They are grouped into categories that may include the following: ➤ Classes ➤ EJB ➤ Execute Queues ➤ GC ➤ JDBC ➤ JDBC<instances> ➤ JMS ➤ JRockit ➤ JTA ➤ Servlets ➤ Thread Pool ➤ Threads ➤ Web Services ➤ Web Sessions <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

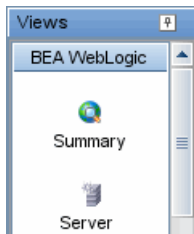
JDBC Resource Contention View Description

The JDBC Resource Contention view displays performance metrics for contention on JDBC resources.

The following is an example of the JDBC Resource Contention view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **BEA WebLogic** to open the BEA WebLogic view group.

If the BEA WebLogic view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **BEA WebLogic** and click **OK**. The BEA WebLogic view group is displayed in the View bar.

Select the appropriate view from the BEA WebLogic view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average JDBC Wait Seconds High, as a trend line. By default, the five probes that have the highest average JDBC Wait Seconds High during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Wait Seconds High (number of seconds the longest request had to wait for a connection). Average JDBC Requests Waiting for Connection is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

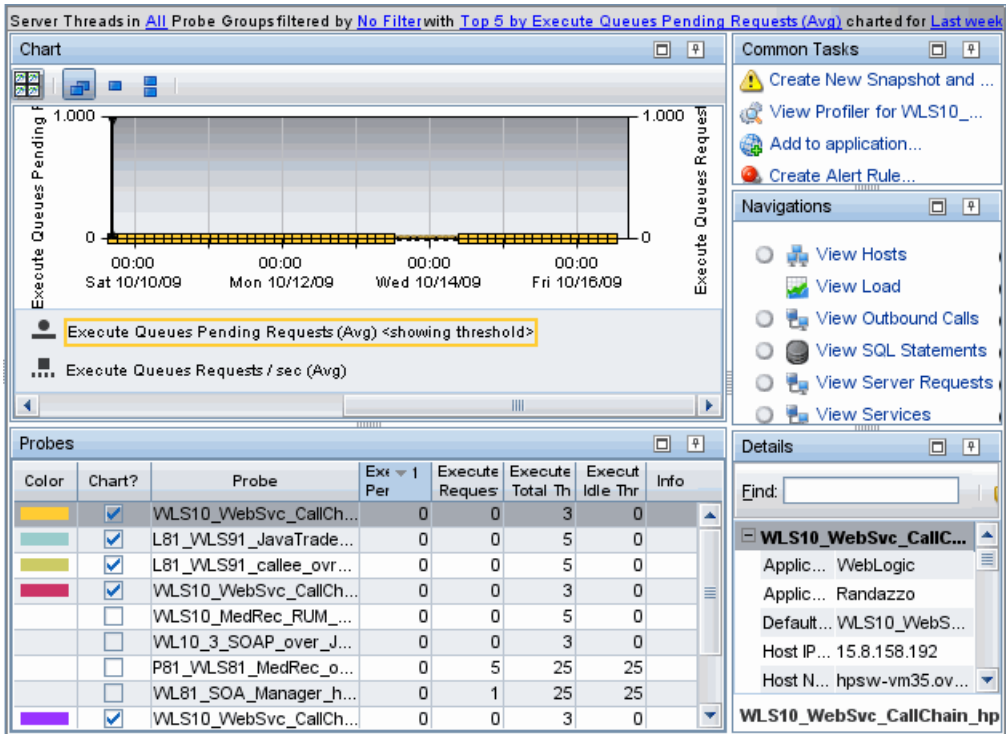
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>JDBC Wait Seconds High. The maximum time a requests has waited for a JDBC database connection.</p> <p>JDBC Requests Waiting for Connection. The number of JDBC database requests waiting for a database connection.</p> <p>JDBC Create Connection Delay. The time delay incurred when creating a new JDBC database connection.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
Details Pane	<p>Shows metrics collected for WebLogic. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ➤ Classes ➤ EJB ➤ Execute Queues ➤ GC ➤ JDBC ➤ JDBC<instances> ➤ JMS ➤ JRockit ➤ JTA ➤ Servlets ➤ Thread Pool ➤ Threads ➤ Web Services ➤ Web Sessions <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

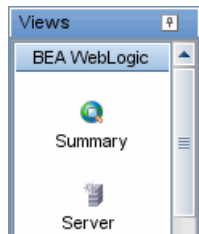
Server Threads View Description

The Server Threads view displays performance metrics for the execute queues.

The following is an example of the Server Threads view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **BEA WebLogic** to open the BEA WebLogic view group.

If the BEA WebLogic view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **BEA WebLogic** and click **OK**. The BEA WebLogic view group is displayed in the View bar.

Select the appropriate view from the BEA WebLogic view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average Execute Queues Pending Requests, as a trend line. By default, the five probes that have the highest average Execute Queues Pending Requests during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Execute Queues Pending Requests. Average Execute Queues Requests per second is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Execute Queues Pending Requests. The number of pending requests in the execution queues.</p> <p>Execute Queues Requests/sec. The number of requests per second in the execution queues.</p> <p>Execute Queues Total Threads. The number of total threads in the execution queues.</p> <p>Execute Queues Idle Threads. The number of idle threads in the execution queue.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
<p>Details Pane</p>	<p>Shows metrics collected for WebLogic. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ➤ Classes ➤ EJB ➤ Execute Queues ➤ GC ➤ JDBC ➤ JDBC<instances> ➤ JMS ➤ JRockit ➤ JTA ➤ Servlets ➤ Thread Pool ➤ Threads ➤ Web Services ➤ Web Sessions <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

34

IBM WebSphere Views

This chapter describes the specialized views for IBM WebSphere.

This chapter includes:

Concepts

- ▶ IBM WebSphere Overview on page 870

Reference

- ▶ IBM WebSphere View Group User Interface on page 871

Concepts

IBM WebSphere Overview

The **IBM WebSphere** view group displays performance data for IBM WebSphere application servers.

The Diagnostics IBM WebSphere view group contains the following views:

- ▶ IBM WebSphere Summary view
- ▶ JDBC Connection Status view
- ▶ JDBC Resource Contention view
- ▶ MQ Connection Stats view
- ▶ MQ Message Driven Bean Stats view
- ▶ MQ Queue Stats view
- ▶ Server Threads view
- ▶ Servlet Session Management view

For details on the user interface, see "IBM WebSphere View Group User Interface" on page 871.

Reference

IBM WebSphere View Group User Interface

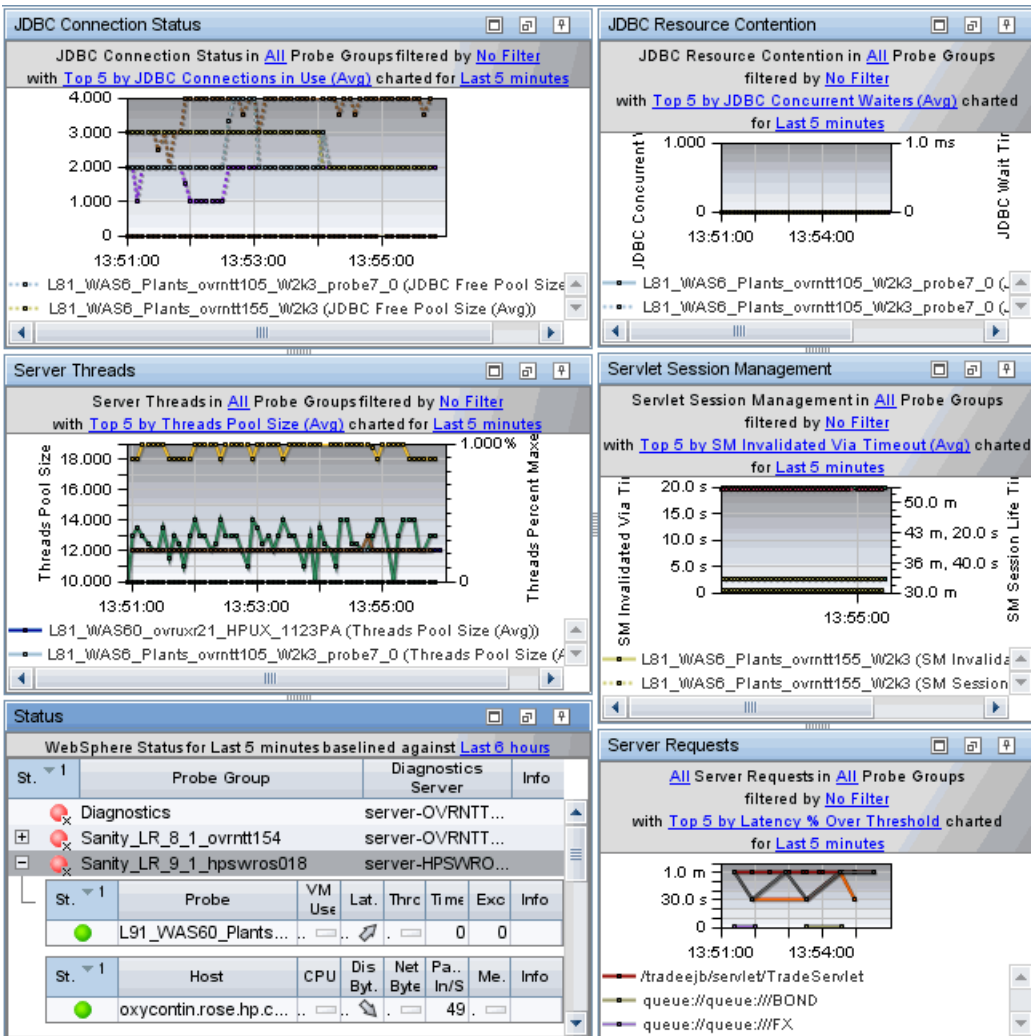
This section includes:


- IBM WebSphere Summary View Description on page 871
- JDBC Connection Status View Description on page 878
- JDBC Resource Contention View Description on page 883
- MQ Connection Stats View Description on page 888
- MQ Message Driven Bean Stats View Description on page 893
- MQ Queue Stats View Description on page 898
- Server Threads View Description on page 903
- Servlet Session Management View Description on page 908

IBM WebSphere Summary View Description

The Summary view is a dashboard view containing the standard Diagnostics Status view along with monitoring versions of the standard Diagnostics Server Requests view and selected views displayed in the IBM WebSphere view group.

The following is an example of the IBM WebSphere Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click IBM WebSphere to open the IBM WebSphere view group.</p> <p>If the IBM WebSphere view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select IBM WebSphere and click OK. The IBM WebSphere view group is displayed in the View bar.</p> <p>Select the appropriate view from the IBM WebSphere view group.</p>
<p>Important information</p>	<p>The view title in each graph or table describes the view using the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.</p>
<p>See also</p>	<ul style="list-style-type: none"> ▶ "JDBC Connection Status View Description" on page 878 ▶ "JDBC Resource Contention View Description" on page 883 ▶ "Server Threads View Description" on page 903 ▶ "Servlet Session Management View Description" on page 908 ▶ "Server Requests View Description" on page 389

IBM WebSphere Graphs

The IBM WebSphere Summary view contains monitoring versions (charts/graphs) of all the views displayed in the IBM WebSphere view group as well as a monitoring version of the Server Requests view and selected views.

UI Elements	Description
JDBC Connections Status graph	<p>A monitoring version of the JDBC Connections Status view.</p> <p>By default, the graph displays the top five probes by JDBC Connections in Use during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Connections in Use.</p> <p>If you double-click the graph the pre-defined workflow displays the JDBC Connection Status view.</p> <p>For information about the JDBC Connection Status view, see "JDBC Connection Status View Description" on page 878.</p>
JDBC Resource Contention graph	<p>A monitoring version of the JDBC Resource Contention view.</p> <p>By default, the graph displays the top five probes by average JDBC Concurrent Waiters during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Concurrent Waiters.</p> <p>If you double-click the graph the pre-defined workflow displays the JDBC Resource Contention view.</p> <p>For information about the JDBC Resource Contention view, see "JDBC Resource Contention View Description" on page 883.</p>

UI Elements	Description
Server Threads graph	<p>A monitoring version of the Server Threads view.</p> <p>By default, the graph displays the top five probes by average Threads Pool Size during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Threads Pool Size.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Threads view.</p> <p>For information about the Server Threads view, see "Server Threads View Description" on page 903.</p>

UI Elements	Description
<p>Servlet Session Management graph</p>	<p>A monitoring version of the Servlet Session Management view.</p> <p>By default, the graph displays the top five probes by average SM Invalidated Via Timeout.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average SM Invalidated Via Timeout.</p> <p>If you double-click the graph the pre-defined workflow displays the Servlet Session Management view.</p> <p>For information about the Servlet Session Management view see "Servlet Session Management View Description" on page 908.</p>
<p>Server Requests graph</p>	<p>A monitoring version of the Server Requests view.</p> <p>By default, the graph displays the top five server requests by Latency % Over Threshold during the last five minutes.</p> <p>By default, the graph is based on all server requests and all probe groups. You can filter the data by individual server request or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the Latency % Over Threshold.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Requests view.</p> <p>For information about the Server Requests view, see "Server Requests View Description" on page 389.</p>

Status Table

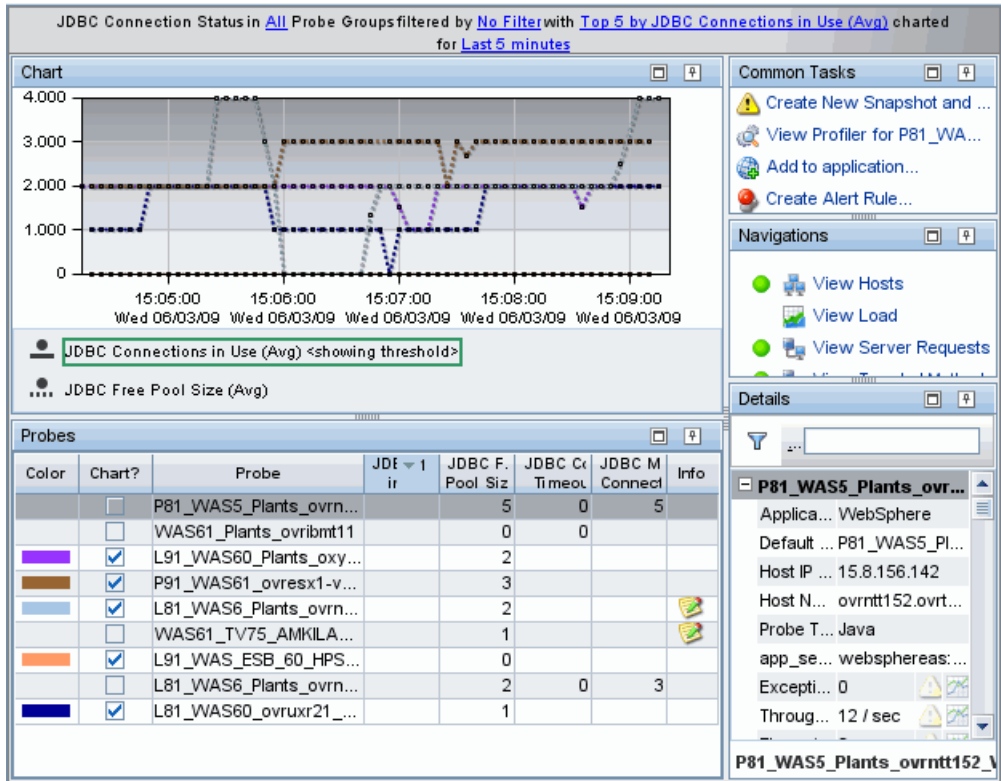
The following user interface elements are included:

UI Elements	Description
Status table	<p>A monitoring version of the Diagnostics Status view.</p> <p>All Diagnostics probe groups are displayed in the status table, but within each probe group, only IBM WebSphere probes are displayed.</p> <p>By default, the first column in the Status table presents the status for each probe group, probe, and host. The status reflects a comparison with thresholds.</p> <p>For IBM WebSphere probes, you see the following:</p> <p>VM Heap Used. The average amount of heap used by the applications being monitored by the probe.</p> <p>Latency. The average response time across all instances.</p> <p>Throughput. The rate of requests for the time period.</p> <p>Timeouts. The number of instances which failed to complete in a reasonable amount of time, possibly due to long periods of time spent in uninstrumented code.</p> <p>Exceptions. The number of instances which failed because the code threw an exception.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p> <p>Host metrics include the following:</p> <p>Host. The host system on which the probe is running.</p> <p>CPU. Average CPU utilization.</p> <p>Disk Bytes/S. The average number of bytes read and written per second.</p> <p>Network Bytes/S. The average number of bytes sent and received per second.</p> <p>PageIn/Second. The average number of virtual memory pages swapped in per second.</p> <p>Memory. The average amount of memory in use.</p> <p>Info. Indicates if an alert rule or comment is defined.</p> <p>For information the Status view, see "Status View Description" on page 520.</p>

JDBC Connection Status View Description

The JDBC Connection Status view displays performance metrics for JDBC usage.

The following is an example of the JDBC Connection Status view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average JDBC Connections in Use, as a trend line. By default, the five probes that have the highest average JDBC Connections in Use during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Connections in Use. JDBC Free Pool Size is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

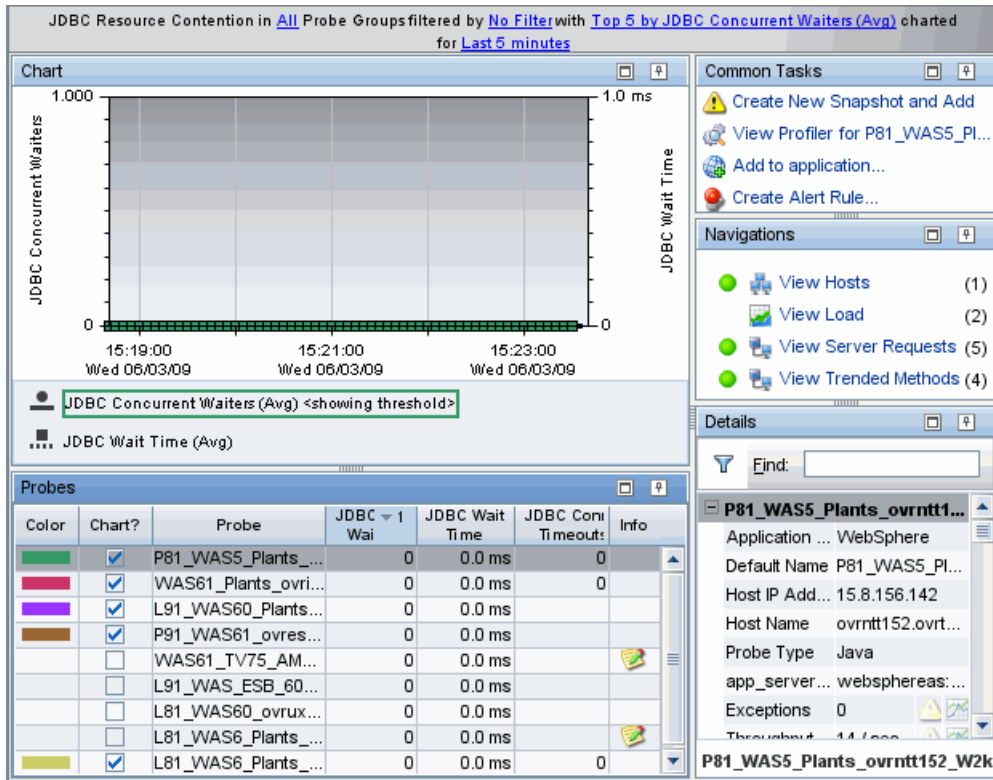
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>JDBC Connections in Use. The number of connection objects in use for a particular connection pool.</p> <p>JDBC Free Pool Size. The number of free connections in the pool.</p> <p>JDBC Connection Timeouts. The number of JDBC connection timeouts.</p> <p>JDBC Managed Connections in Use.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
Details Pane	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

JDBC Resource Contention View Description

The JDBC Resource Contention view displays performance metrics for contention on JDBC resources.

The following is an example of the JDBC Resource Contention view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average JDBC Concurrent Waiters, as a trend line. By default, the five probes that have the highest average JDBC Concurrent Waiters during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average JDBC Concurrent Waiters (number of concurrent requests waiting for a connection). Average JDBC Wait Time is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

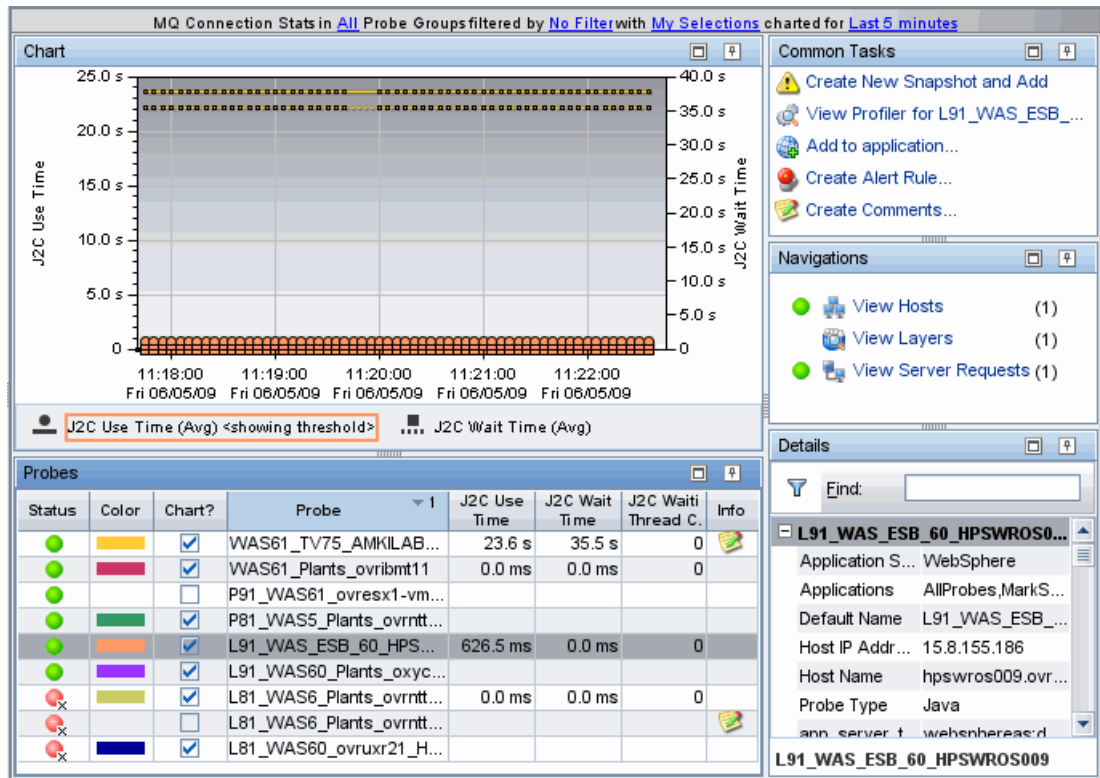
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>JDBC Concurrent Waiters. The number of concurrent requests waiting for a JDBC connection.</p> <p>JDBC Wait Time. The average amount of time a JDBC connection waits before being serviced.</p> <p>JDBC Connection Timeouts. The number of JDBC connection timeouts.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
<p>Details Pane</p>	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

MQ Connection Stats View Description

The MQ Connection Stats view displays performance metrics for the execute queues.

The following is an example of the MQ Connection Stats view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average J2C Use Time, as a trend line. By default, the five probes that have the highest average J2C Use Time during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Use Time. Average J2C Wait Time is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

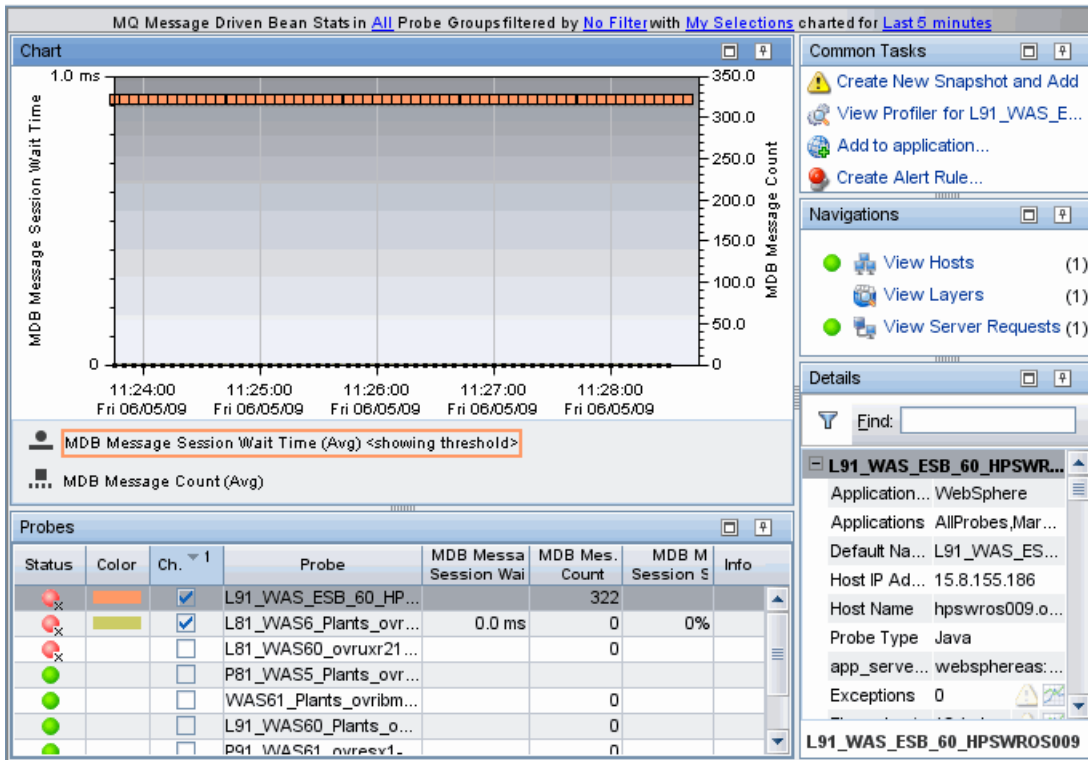
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>J2C Use Time. The average time a J2C connection is in use.</p> <p>J2C Wait Time. The average time a J2C connection waits before service.</p> <p>J2C Waiting Thread Count. The number of threads waiting for J2C service.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
Details Pane	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

MQ Message Driven Bean Stats View Description

The MQ Message Driven Bean Stats view displays performance metrics for the execute queues.

The following is an example of the MQ Message Driven Bean Stats view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of averageMDB Message Session Wait Time, as a trend line. By default, the five probes that have the highest average MDB Message Session Wait Time during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average MDB Message Session Wait Time. Average MDB Message Count is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

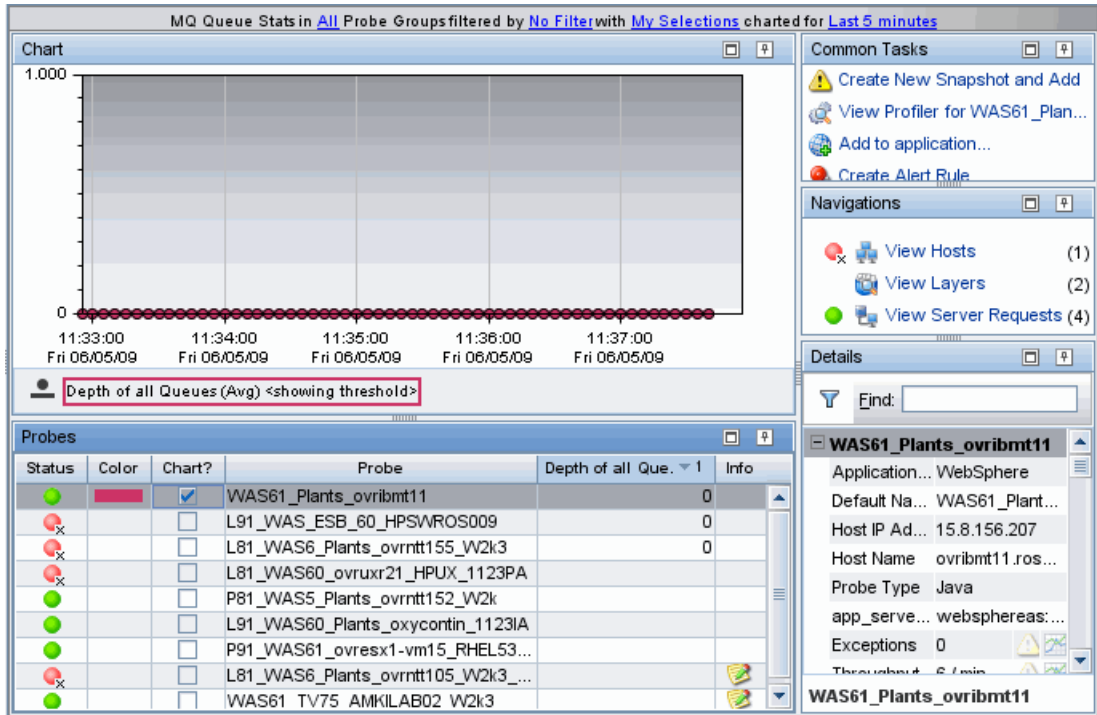
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>MDB Message Session Wait Time.</p> <p>MDB Message Count.</p> <p>MDB Message Session Server Percent in Use.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
<p>Details Pane</p>	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ➤ Classes ➤ EJB ➤ GC, ➤ J2C Connections ➤ JDBC Connection Pools ➤ MDB ➤ Servlet Session Manager ➤ Thread Pools ➤ Threads ➤ Transaction Manager ➤ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

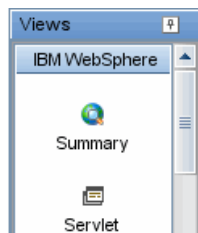
MQ Queue Stats View Description

The MQ Queue Stats view displays performance metrics for the execute queues.

The following is an example of the MQ Queue Stats view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of averageDepth of All Queues, as a trend line. By default, the five probes that have the highest average Depth of All Queues during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Depth of All Queues. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

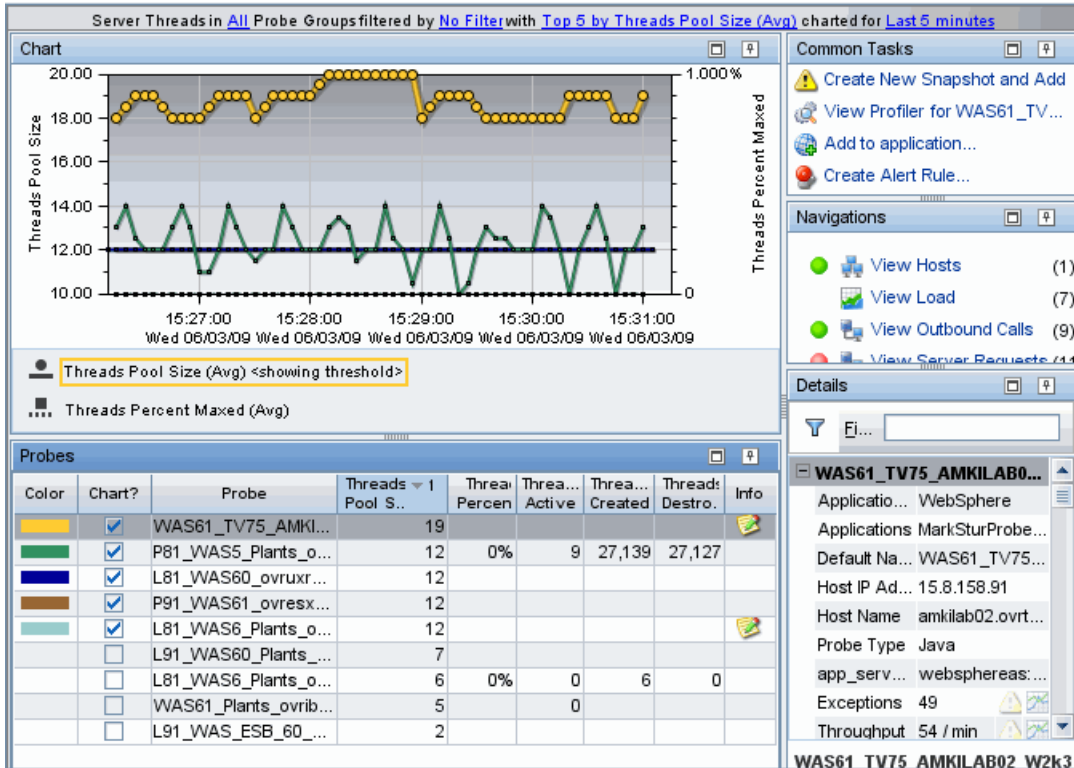
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Depth of All Queues. The average depth of all JMS queues.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
Details Pane	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Server Threads View Description

The Server Threads view displays performance metrics for the execute queues.

The following is an example of the Server Threads view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average Threads Pool Size, as a trend line. By default, the five probes that have the highest average Threads Pool Size during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Threads Pool Size. Average Threads Percent Maxed is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

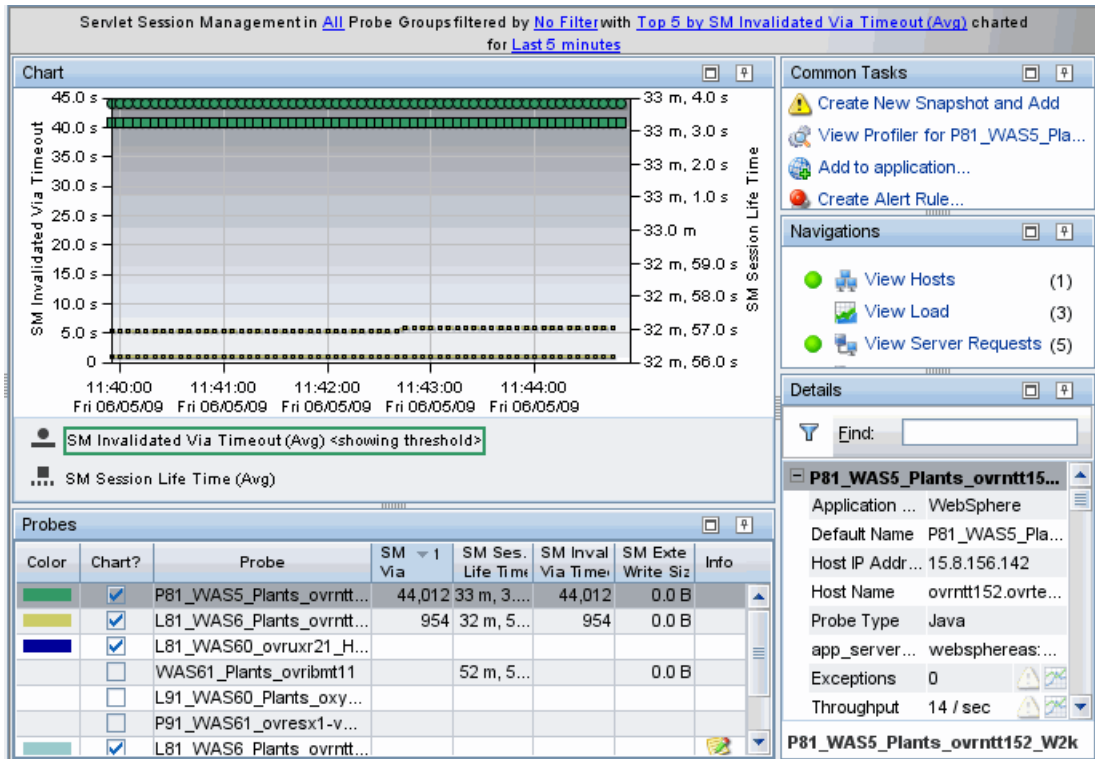
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>Threads Pool Size. Average number of threads in the pool.</p> <p>Threads Percent Maxed. Average percent of the time that all threads are in use.</p> <p>Threads Active. The number of active Java threads.</p> <p>Threads Created. The average number of Java threads created per second</p> <p>Threads Destroyed. The average number of Java threads destroyed per second.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Host.</p>
<p>Details Pane</p>	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ➤ Classes ➤ EJB ➤ GC, ➤ J2C Connections ➤ JDBC Connection Pools ➤ MDB ➤ Servlet Session Manager ➤ Thread Pools ➤ Threads ➤ Transaction Manager ➤ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

Servlet Session Management View Description

The Servlet Session Management view displays performance metrics for the execute queues.

The following is an example of the Servlet Session Management view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **IBM WebSphere** to open the IBM WebSphere view group.

If the IBM WebSphere view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **IBM WebSphere** and click **OK**. The IBM WebSphere view group is displayed in the View bar.

Select the appropriate view from the IBM WebSphere view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average SM Invalidated Via Timeout, as a trend line. By default, the five probes that have the highest average SM Invalidated Via Timeout during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average SM Invalidated Via Timeout. Average SM Session Life Time is also charted. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>SM Invalidated Via Timeout. Number of session that were invalidated by timeout.</p> <p>SM Session Life Time. Average session lifetime in milliseconds.</p> <p>SM External Write Size.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Load.</p>
Details Pane	<p>Shows metrics collected for WebSphere. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

35

CICS Views

This chapter describes the specialized views for CICS.

This chapter includes:

Concepts

- ▶ CICS Overview on page 916

Reference

- ▶ CICS View Group User Interface on page 917

Concepts

CICS Overview

The **CICS** view group displays performance data for CICS (Customer Information Control System) transaction servers. CICS runs primarily on IBM mainframe systems under z/OS.

CICS Transaction Gateway is a Java connector that handles communication between a WebSphere application server and a CICS system.

HP Diagnostics monitors communication that takes place between the WebSphere application Server and the CICS Transaction Gateway, and displays these metrics in the CICS view group.

The Diagnostics CICS view group contains the following views:

- ▶ CICS Summary view
- ▶ J2C Connection Pool view
- ▶ J2C Connections and Faults view
- ▶ J2C Load view
- ▶ J2C Usage Time view
- ▶ J2C Wait Time view

For details on the user interface, see "CICS View Group User Interface" on page 917.

Reference

CICS View Group User Interface

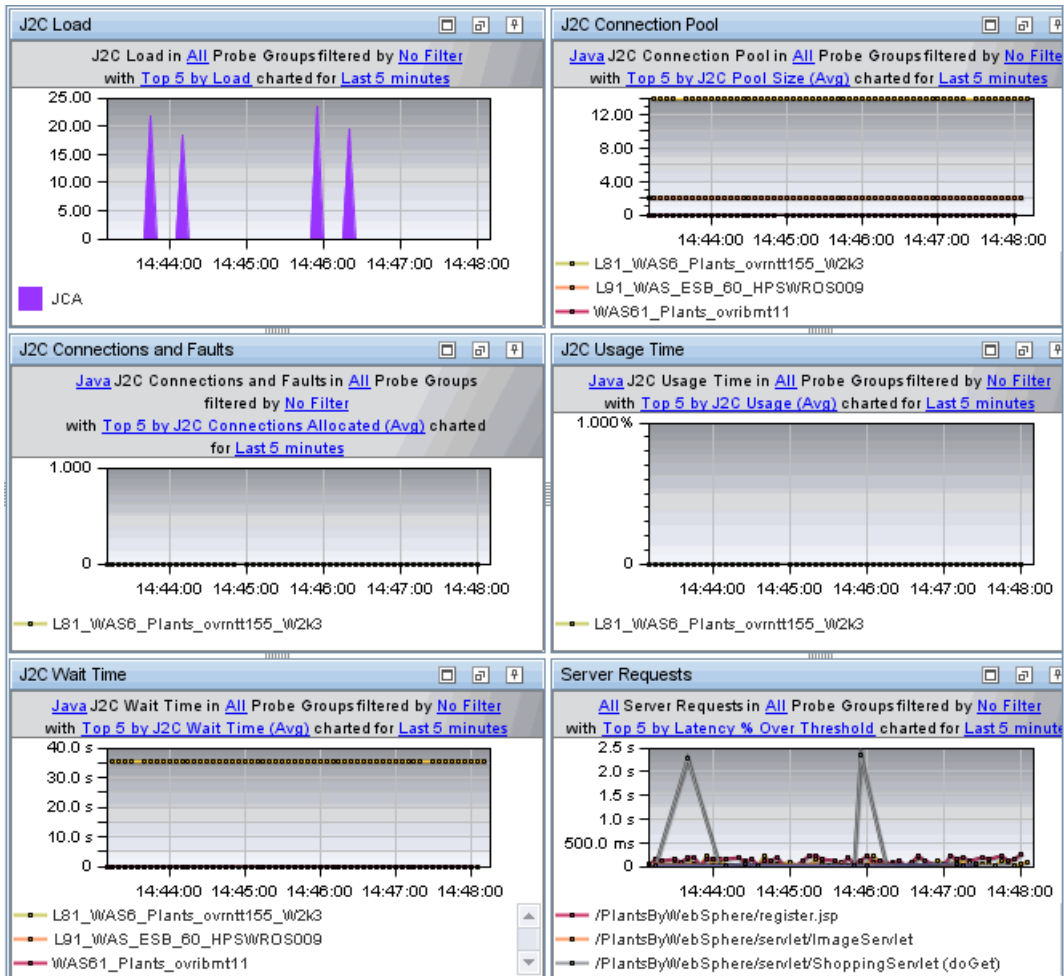
This section includes:

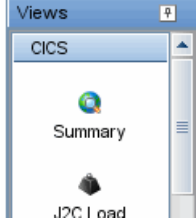
- ▶ CICS Summary View Description on page 917
- ▶ J2C Connection Pool View Description on page 922
- ▶ J2C Connections and Faults View Description on page 927
- ▶ J2C Load View Description on page 932
- ▶ J2C Usage Time View Description on page 936
- ▶ J2C Wait Time View Description on page 941

CICS Summary View Description

The Summary view is a dashboard view containing monitoring versions of the standard Diagnostics Server Requests view and the other views displayed in the CICS view group. CICS views show Java 2 Connector (2EC) metrics. The data is used to monitor J2C connection pool performance.

The following is an example of the CICS Summary view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click CICS to open the CICS view group.</p> <p>If the CICS view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select CICS and click OK. The CICS view group is displayed in the View bar.</p> <p>Select the appropriate view from the CICS view group.</p>
<p>See also</p>	<ul style="list-style-type: none"> ➤ "J2C Load View Description" on page 932 ➤ "J2C Connection Pool View Description" on page 922 ➤ "J2C Connections and Faults View Description" on page 927 ➤ "J2C Usage Time View Description" on page 936 ➤ "J2C Wait Time View Description" on page 941 ➤ "Server Requests View Description" on page 389

The CICS Summary view contains monitoring versions (charts/graphs) of the views displayed in the CICS view group as well as a monitoring version of the Server Requests view. CICS views show Java 2 Connector (2EC) metrics. The data is used to monitor J2C connection pool performance.

UI Elements	Description
<p>J2C Load graph</p>	<p>A monitoring version of the J2C Load view.</p> <p>By default, the graph displays the top five probes by Load during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average load.</p> <p>If you double-click the graph the pre-defined workflow displays the J2C Load view.</p> <p>For information about the J2C Load view, see "J2C Load View Description" on page 932.</p>
<p>J2C Connection Pool graph</p>	<p>A monitoring version of the J2C Connection Pool view.</p> <p>By default, the graph displays the top five probes by average J2C Pool Size during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Pool Size.</p> <p>If you double-click the graph the pre-defined workflow displays the J2C Connection Pool view.</p> <p>For information about the J2C Connection Pool view, see "J2C Connection Pool View Description" on page 922.</p>

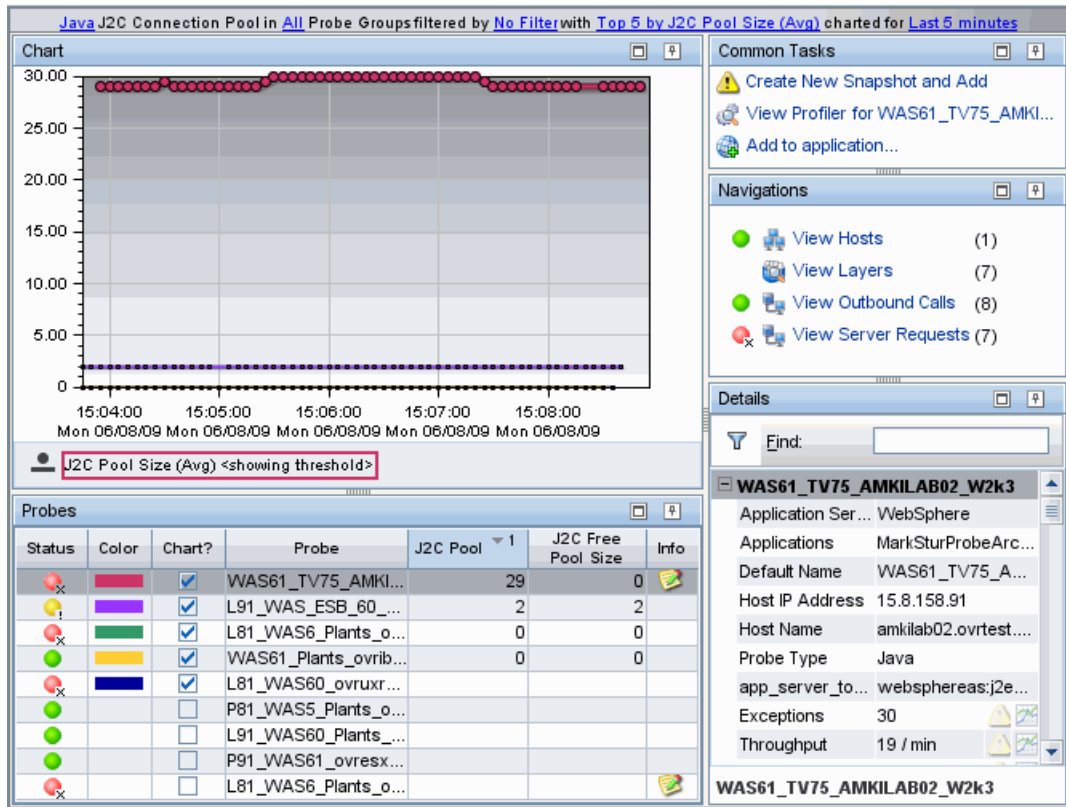
UI Elements	Description
J2C Connections and Faults graph	<p>A monitoring version of the SJ2C Connections and Faults view.</p> <p>By default, the graph displays the top five probes by average J2C Connections Allocated during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Connections Allocated.</p> <p>If you double-click the graph the pre-defined workflow displays the J2C Connections and Faults view.</p> <p>For information about the J2C Connections and Faults view, see "J2C Connections and Faults View Description" on page 927.</p>
J2C Usage Time graph	<p>A monitoring version of the J2C Usage Time view.</p> <p>By default, the graph displays the top five probes by average J2C Usage during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Usage.</p> <p>If you double-click the graph the pre-defined workflow displays the J2C Usage Time view.</p> <p>For information about the J2C Usage Time view, see "J2C Usage Time View Description" on page 936.</p>

UI Elements	Description
J2C Wait Time graph	<p>A monitoring version of the J2C Wait Time view.</p> <p>By default, the graph displays the top five probes by average J2C Wait Time during the last five minutes.</p> <p>By default, the graph is based on all probe groups. You can filter the data by individual probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Wait Time.</p> <p>If you double-click the graph the pre-defined workflow displays the J2C Wait Time view.</p> <p>For information about the J2C Wait Time view, see "J2C Wait Time View Description" on page 941.</p>
Server Requests graph	<p>A monitoring version of the Server Requests view.</p> <p>By default, the graph displays the top five server requests by Latency % Over Threshold during the last five minutes.</p> <p>By default, the graph is based on all server requests and all probe groups. You can filter the data by individual server request or probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the Latency % Over Threshold.</p> <p>If you double-click the graph the pre-defined workflow displays the Server Requests view.</p> <p>For information about the Server Requests view, see "Server Requests View Description" on page 389.</p>

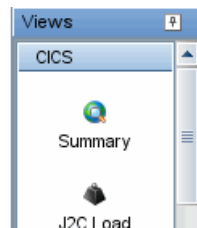
J2C Connection Pool View Description

The J2C Connection Pool view displays performance metrics for J2C connection pools.

The following is an example of the J2C Connection pool view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **CICS** to open the CICS view group.

If the CICS view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **CICS** and click **OK**. The CICS view group is displayed in the View bar.

Select the appropriate view from the CICS view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average J2C Pool Size, as a trend line. By default, the five probes that have the highest average J2C Pool Size during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Pool Size. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

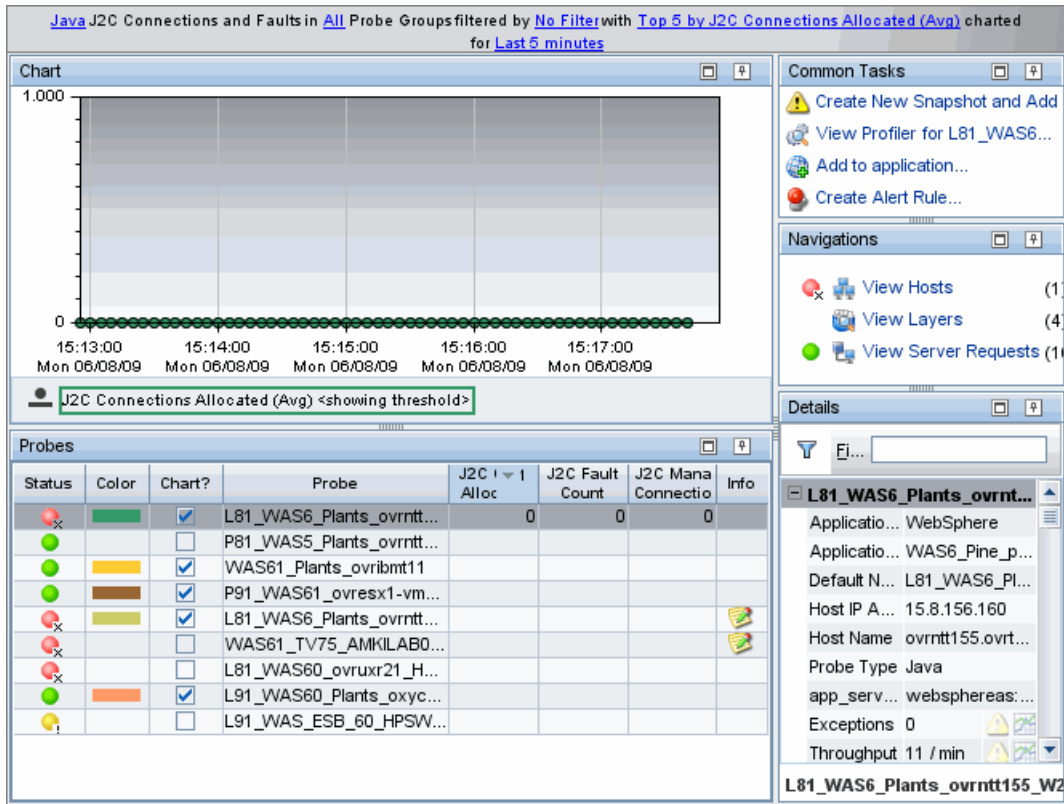
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>J2C Pool Size. The size of the J2C connection pool.</p> <p>J2C Free Pool Size. The size of the free J2C connection pool.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
Details Pane	<p>Shows metrics collected for CICS. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

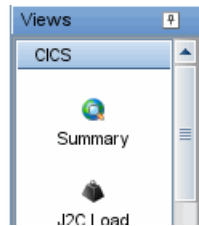
J2C Connections and Faults View Description

The J2C Connections and Faults view displays performance metrics for J2C connections.

The following is an example of the J2C Connections and Faults view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **CICS** to open the CICS view group.

If the CICS view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **CICS** and click **OK**. The CICS view group is displayed in the View bar.

Select the appropriate view from the CICS view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average J2C Connections Allocated, as a trend line. By default, the five probes that have the highest average J2C Connections Allocated during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Connections Allocated. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

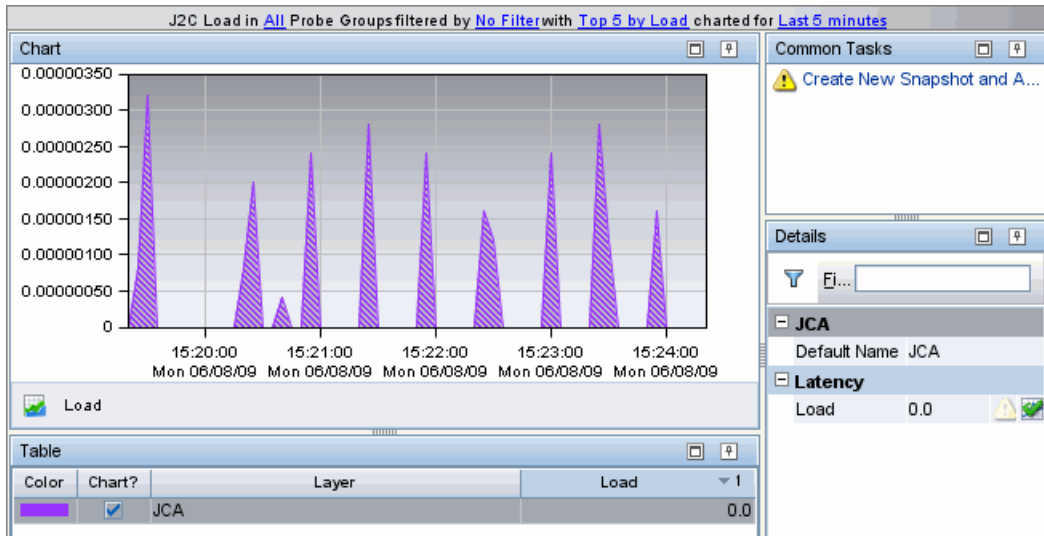
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>J2C Connections Allocated.</p> <p>J2C Fault Count.</p> <p>J2C Managed Connections.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
Details Pane	<p>Shows metrics collected for CICS. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

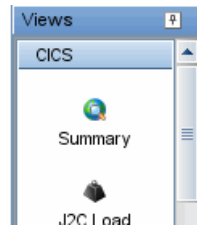
J2C Load View Description

The J2C Connections and Faults view displays a breakdown of the load across the different J2C classes.

The following is an example of the J2C Load view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **CICS** to open the CICS view group.

If the CICS view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **CICS** and click **OK**. The CICS view group is displayed in the View bar.

Select the appropriate view from the CICS view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

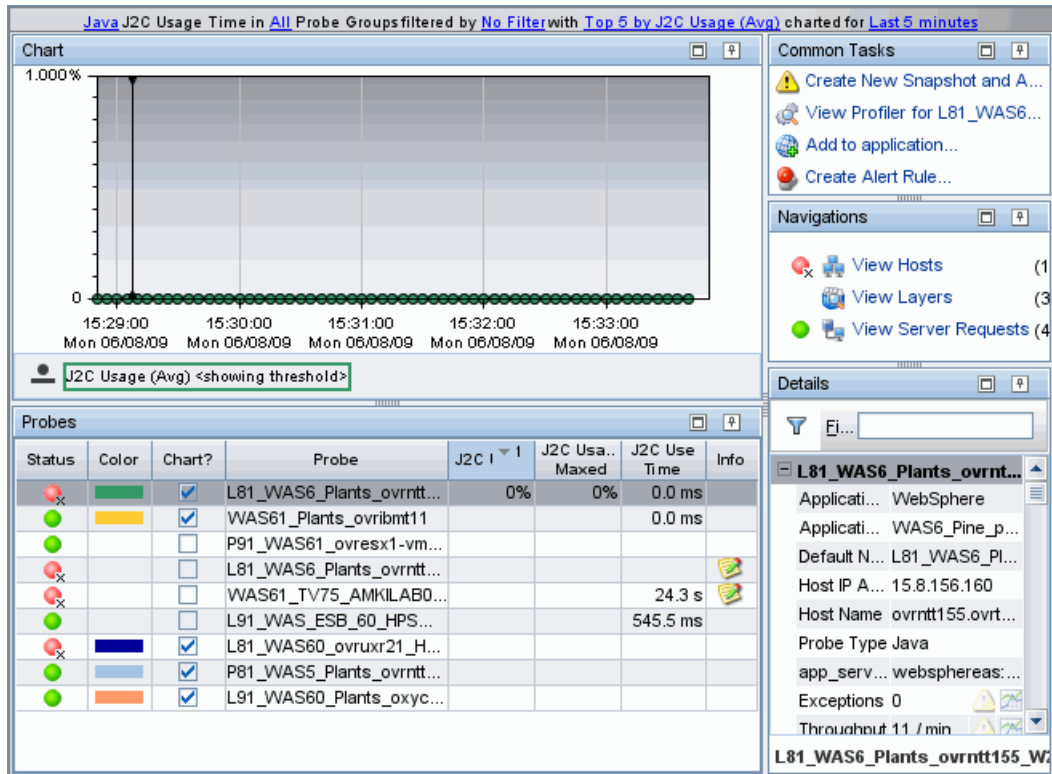
UI Elements	Description
Chart	<p>Displays a graph of average Load, as stacked graph. By default, the five layers that have the highest average Load during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average Load.</p> <p>Layer Details tooltip for the trend line on the graph (or the Layer column in the table) displays the following detail data:</p> <p>Layer. The name of the layer (J2C class).</p> <p>The Layers Details tooltip for selected points in the stacked graph displays additional information about the data used to plot the selected data point such as sample start and end times.</p>
Table	<p>Lists the layers that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Layer. Name of the layer (J2C classes).</p> <p>Load. An indication of how much work this layer performs, averaged over the selected time range.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is to view sub-layers, if any.</p>
Details Pane	<p>Shows metrics collected for CICS. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

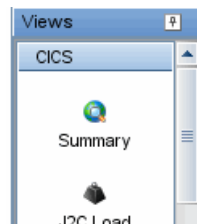
J2C Usage Time View Description

The J2C Usage Time view displays performance metrics for J2C usage.

The following is an example of the J2C Usage Time view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **CICS** to open the CICS view group.

If the CICS view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **CICS** and click **OK**. The CICS view group is displayed in the View bar.

Select the appropriate view from the CICS view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average J2C Usage, as a trend line. By default, the five probes that have the highest average J2C Usage during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Usage. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

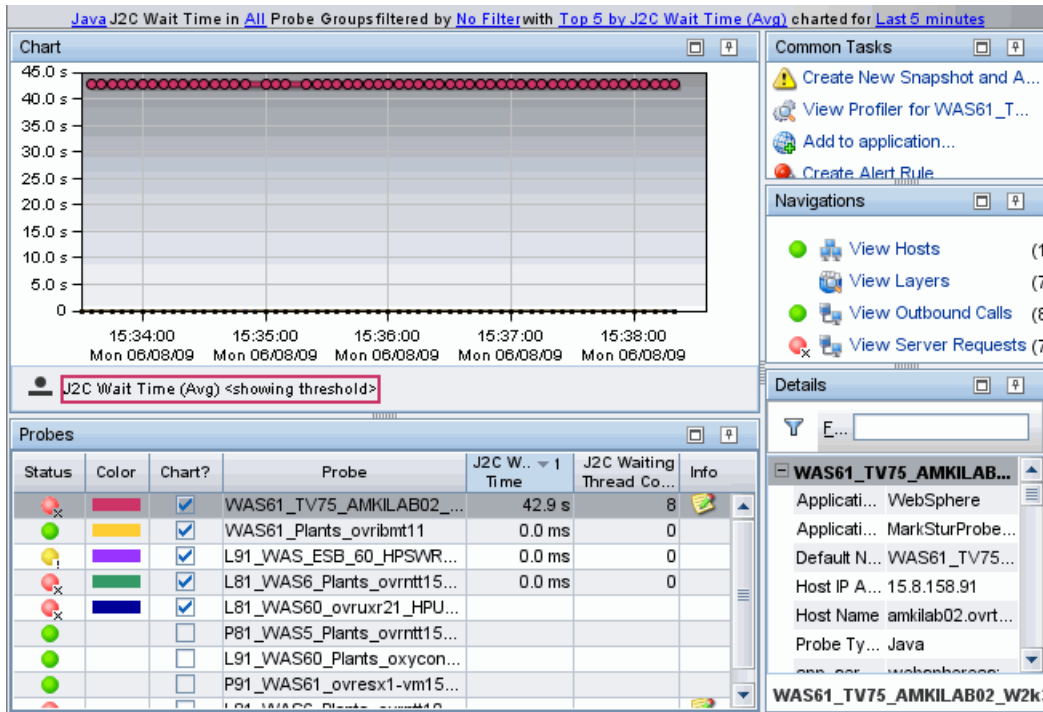
UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>J2C Usage. Average percent of the pool that is in use.</p> <p>J2C Usage Maxed. Average percent of the time that all connections are in use.</p> <p>J2C Use Time. The average time in milliseconds that connections are in use.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
Details Pane	<p>Shows metrics collected for CICS. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ▶ Classes ▶ EJB ▶ GC, ▶ J2C Connections ▶ JDBC Connection Pools ▶ MDB ▶ Servlet Session Manager ▶ Thread Pools ▶ Threads ▶ Transaction Manager ▶ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

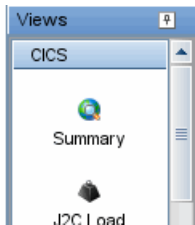
J2C Wait Time View Description

The J2C Usage Time view displays performance metrics for J2C wait time.

The following is an example of the J2C Wait Time view.



To access



In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, **Entire Enterprise**). The Diagnostics views open.

In the View bar, click **CICS** to open the CICS view group.

If the CICS view group is hidden, right-click inside the View bar and select **Open a View Group**. In the Open a View Group dialog box, select **CICS** and click **OK**. The CICS view group is displayed in the View bar.

Select the appropriate view from the CICS view group.

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average J2C Wait Time, as a trend line. By default, the five probes that have the highest average J2C Wait Time during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis shows chronological time. The y-axis shows the average J2C Wait Time. Thresholds (if set on an entity's metric value) are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Probe Details tooltip for the trend line on the graph (or the Probe column in the table) displays the following detail data:</p> <p>Probe. The name of the probe collecting the metrics.</p> <p>Probe Group. The name of the probe group.</p> <p>Host. The host on which the probe is running.</p> <p>Probe Type. The type of probe (for example, Java).</p> <p>The Probe Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p>Lists the probes that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p>The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p>By default the following columns are displayed:</p> <p>Status. By default the first column shows the status for the probe. The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p>Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p>Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p>Probe. Name of the probe collecting the metrics.</p> <p>J2C Wait Time. The average time a J2C connection waits before service.</p> <p>J2C Waiting Thread Count. The number of threads waiting for J2C service.</p> <p>Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
<p>Common Tasks and Navigations</p>	<p>Common tasks and actions relevant to the probes displayed in this view.</p> <p>The drill down navigation based on a pre-defined workflow is View Server Requests. Other common drill downs are View Hosts and View Layers.</p>
<p>Details Pane</p>	<p>Shows metrics collected for CICS. They are grouped into categories that may include the following:</p> <ul style="list-style-type: none"> ➤ Classes ➤ EJB ➤ GC, ➤ J2C Connections ➤ JDBC Connection Pools ➤ MDB ➤ Servlet Session Manager ➤ Thread Pools ➤ Threads ➤ Transaction Manager ➤ Web Applications <p>See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."</p>

36

External Monitors Views

This chapter describes the specialized views for data coming from external sources such as SiteScope monitors.

This chapter includes:

Concepts

- ▶ External Monitors Overview on page 948

Tasks

- ▶ How to Display a Graph and See Status in the Monitors View on page 952

Reference

- ▶ External Monitors View Group User Interface on page 953

Concepts

External Monitors Overview

The External Monitors view group contains a single view labeled **Monitors**. The Monitors view currently presents data gathered by SiteScope monitors.

HP SiteScope is an agentless monitoring solution designed to ensure the availability and performance of distributed IT infrastructures—for example, servers, operating systems, network devices, network services, applications, and application components. SiteScope can monitor utilization, response time, usage, and resource availability of a variety of host types and application platforms.

Within SiteScope, you create your monitoring structure, configure the Diagnostics integration, and select objects for the integration. SiteScope then forwards data to HP Diagnostics for display in the Diagnostics UI.

SiteScope can forward data to HP Diagnostics that allows the user to see a more complete view of the application servers that are monitored by Diagnostics and/or provide insight into the infrastructure components that these application servers are deployed into. For example, integrating data from the SNMP monitor can help determine problems with the infrastructure that the application server runs on.

SiteScope sends data to Diagnostics from the monitor's counters, along with the name of the monitor, the name of the system that the monitor is targeting, the name of the SiteScope server and the hierarchy of groups that the monitor is defined in.

The Diagnostics integration with SiteScope is configured in SiteScope (see the "Integration Preferences" section of the SiteScope documentation for details).

When configuring SiteScope for integration with Diagnostics it will be important to ensure the metrics sent to Diagnostics conform to the allowable units in Diagnostics. The allowable units within Diagnostics are as follows:

Time Units	Size Units	Numerical Counter	Contribution	Load
microseconds	bytes	count	percent	load
milliseconds	kilobytes			
seconds	megabytes		fraction_percent -	
minutes	gigabytes		(A value in the	
hours			range 0.0-1.0	
days			which is	
			multiplied by 100	
			by the UI)	

If an invalid unit is sent to Diagnostics from SiteScope, it will be defaulted to the count unit.

After configuring the SiteScope integration with Diagnostics (see the SiteScope documentation), and as monitors are associated with the integration, SiteScope generates a file `/SiteScope/conf/integration/data_integration_uom.xml` that controls the mappings of SiteScope monitors to Diagnostics metrics. SiteScope units are captured from the monitored source and will sometimes need to be mapped to the appropriate Diagnostics metric unit. Modify the xml file as needed.

For example:

```
<monitor type="CPU">
  <counter units="%" name="utilization"/>
  <counter units="%" name="utilization cpu # 1"/>
  <counter units="%" name="utilization cpu # 2"/>
  <counter units="%" name="utilization cpu # 3"/>
  <counter units="%" name="utilization cpu # 4"/>
</monitor>
```

Should be modified as follows:

```
<monitor type="CPU">
  <counter units="percent " name="utilization"/>
```

```
<counter units="percent " name="utilization cpu # 1"/>  
<counter units="percent " name="utilization cpu # 2"/>  
<counter units="percent " name="utilization cpu # 3"/>  
<counter units="percent " name="utilization cpu # 4"/>  
</monitor>
```

Sometimes SiteScope is unable to determine an appropriate unit of measure, and sets the units to "unknown". For any counters with a unit of "unknown", SiteScope will not forward the data to Diagnostics. You must modify the **data_integration_uom.xml** file and assign an appropriate unit to the counter before data will flow to Diagnostics for that counter.

Once configured, SiteScope data flows to the Diagnostics servers configured to receive the data after the monitors finish running.

In the Monitors view the SiteScope data is presented in the form of graphs. This SiteScope data can help in understanding of the root cause of the problems identified by the probes (for example, slow method execution) by correlating between slow performance of the application (based on data from the probes) and various metrics provided by SiteScope monitors.

Caution: When you first view SiteScope data in the Monitors view, by default the monitor's status is gray and no data is graphed. To see a status (red, yellow, green), you must first set a threshold on a metric (in the details pane). To see data in the graph, you must first select a metric to be charted (in the details pane). See "How to Display a Graph and See Status in the Monitors View" on page 952.

You can set thresholds, define metric alert rules for the SiteScope metrics and add SiteScope metrics to snapshots as you would other Diagnostics metrics.

You can see monitors listed under probe groups in views such as the Server Summary view. Monitors are associated with Diagnostics probe groups whose names are derived from the names of the groups and sub-groups in which the monitor is defined in SiteScope. The group hierarchy is represented as a path. For example, a monitor in the "Windows" sub-group of the "Servers" group will be in the "Servers/Windows" probe group in Diagnostics.

Note: You may see some of the same metrics in both the Monitors view (displaying SiteScope data) and others Diagnostics views. The difference between the SiteScope and Diagnostics view of the same metrics has to do with the frequency of the measurements and aggregation.

For details on the user interface, see "Monitors View Description" on page 954.

Tasks

How to Display a Graph and See Status in the Monitors View

This task describes how to display a graph in the Monitors view and see the status of a monitor.

To display graph data and see monitor status in this view:

- 1** Check the Chart? column for a monitor in the graph entity table.
- 2** Select a metric in the details pane and check the icon to **Start Charting this Metric**.
- 3** To see monitor status, select a metric in the details pane and set a threshold. You'll note when setting a threshold in this view that there aren't any default threshold values for the SiteScope metrics. To set a threshold, use the Set Threshold dialog box, select **Use custom threshold** and define the threshold you want set for the SiteScope metric.

Reference

External Monitors View Group User Interface

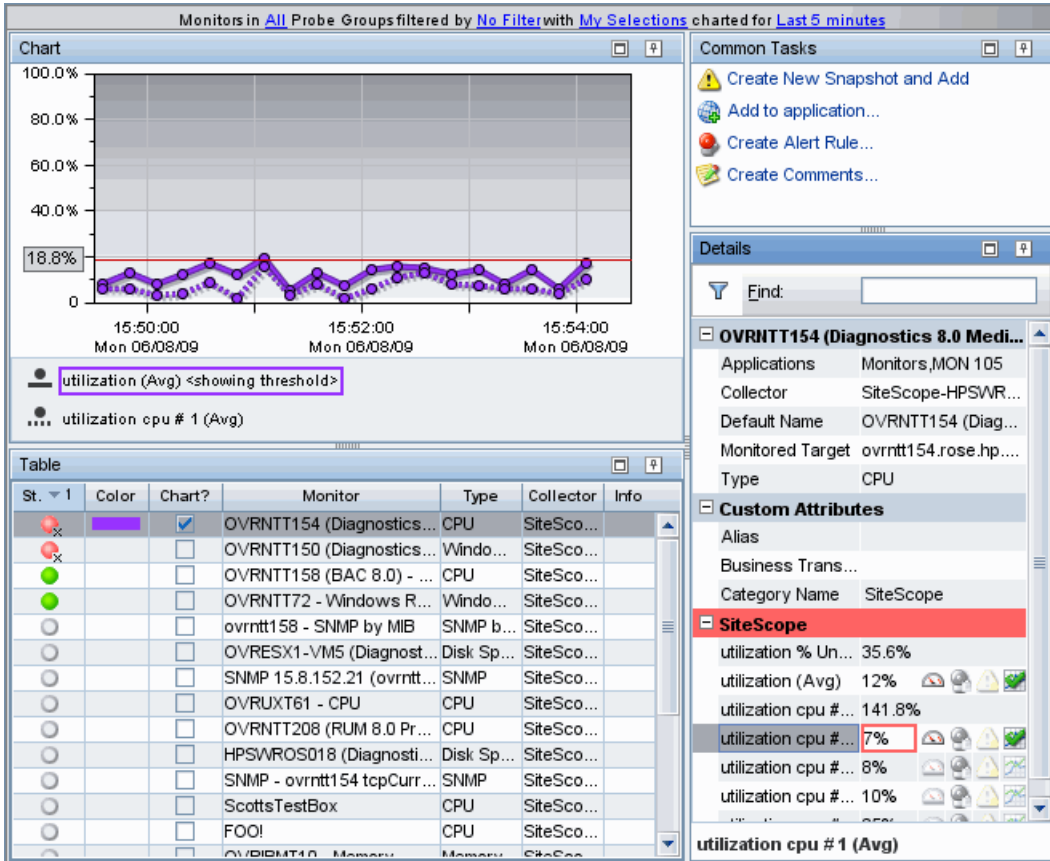
This section includes:


- ▶ [Monitors View Description on page 954](#)

Monitors View Description

The Monitors view currently presents data gathered by SiteScope monitors.

The following is an example of the Monitors view.



<p>To access</p> 	<p>In the navigation pane of the Diagnostics Applications window, double-click an application name (for example, Entire Enterprise). The Diagnostics views open.</p> <p>In the View bar, click External Monitors to open the External Monitors view group.</p> <p>If the External Monitors view group is hidden, right-click inside the View bar and select Open a View Group. In the Open a View Group dialog box, select External Monitors and click OK. The External Monitors view group is displayed in the View bar.</p> <p>Select the appropriate view from the External Monitors view group.</p>
<p>Important information</p>	<p>The Diagnostics integration with SiteScope is configured in SiteScope. When configuring SiteScope for integration with Diagnostics it is important to ensure the metrics sent to Diagnostics conform to the allowable units in Diagnostics (see "External Monitors Overview" on page 948).</p>
<p>Relevant tasks</p>	<p>See "How to Display a Graph and See Status in the Monitors View" on page 952.</p> <p>When you first view SiteScope data in the Monitors view, by default the monitor's status is gray and no data is graphed. To see a status (red, yellow, green), you must first set a threshold on a metric (in the details pane). To see data in the graph, you must first select a metric to be charted (in the details pane).</p>

Common Elements

The following user interface elements are included:

UI Elements	Description
<Common elements>	For details on the common UI controls, see Chapter 3, "Common Controls in the Diagnostics UI."
<View title and view filters>	The view title describes the view and shows the current settings for the view filters. You can update the view filters to change the information that is displayed in the view as described in "How to Work with View Filters" on page 131.

Page Content

The following user interface elements are included:

UI Elements	Description
Chart	<p>Displays a graph of average utilization, as a trend line. By default, the five SiteScope monitors that have the highest average latency during the selected time range are charted. You can use the view filters in the Title bar to filter the data for a particular probe group.</p> <p>The x-axis of the graph shows chronological time. The y-axis of the graph shows the average Utilization. Thresholds are displayed as a red horizontal line with the threshold value shown to the left of the chart.</p> <p>Monitor Details tooltip for the trend line on the graph (or the Trended Methods column in the table) displays the following detail data:</p> <p>Monitor. The name of the SiteScope monitor configured by the SiteScope administrator.</p> <p>Probe Group. The name of the probe group associated with the monitor.</p> <p>Type. The type of SiteScope monitor (for example, Disk Space, CPU, Memory, SNMP, Windows Services, UNIX Resources).</p> <p>Category Name. SiteScope.</p> <p>The Monitor Details tooltip for nodes on the trend line displays additional information about the data used to plot the selected node such as sample start and end times.</p>

UI Elements	Description
Table	<p data-bbox="582 222 1216 348">Lists the SiteScope monitors that pertain to the context selected. The metrics reported in the table are aggregated based on the time period selected in the view filters in the Title bar.</p> <p data-bbox="582 366 1216 552">The table is highly customizable. Right-click any column to select Customize Columns and specify which metrics you want displayed in the columns in the table and the order in which they should appear. You can also specify which columns to use to sort the rows in the table. See "Table Header Controls" on page 114 for details.</p> <p data-bbox="582 569 1115 597">By default the following columns are displayed:</p> <p data-bbox="582 614 1216 829">Status. By default the first column shows the status for the monitor. When you first view SiteScope data in the Monitors view, by default the monitor's status is gray and no data is graphed. To see a status (red, yellow, green), you must first set a threshold on a metric (in the details pane). To see data in the graph, you must first select a metric to be charted (in the details pane).</p> <p data-bbox="582 847 1196 939">The status reflects how the performance of the item compares with the threshold setting. See "Status Indicators" on page 517 for details on status indicators.</p> <p data-bbox="582 956 1216 1013">Color. The color of the block is the color that Diagnostics uses to chart the metrics for the entity.</p> <p data-bbox="582 1031 1216 1088">Chart?. From the table you can select the entities that are to be charted on the graph.</p> <p data-bbox="582 1105 1200 1197">Type. The type of SiteScope monitor (for example, Disk Space, CPU, Memory, SNMP, Windows Services, UNIX Resources).</p> <p data-bbox="582 1215 1065 1272">Collector. The SiteScope server displayed as SiteScope-<SiteScope_server_name>.</p> <p data-bbox="582 1289 1216 1347">Info. This column indicates if the entity has an alert rule or comment defined.</p>

UI Elements	Description
Common Tasks and Navigations	Common actions relevant to the monitor. You can select Add to Snapshot, which is useful if you want to view SiteScope data combined with probe data.
Details Pane	Shows metrics collected on external monitors. See Chapter 4, "Working with Thresholds and Metrics in the Details Pane."

Part V

Using the Diagnostics Profilers for Java and .NET

37

Diagnostics Profiler for Java

This chapter describes how to use the Diagnostics Profiler for Java.

This chapter includes:

Concepts

- ▶ About the Java Diagnostics Profiler on page 965
- ▶ How the Java Agent Provides Data for the Java Profiler on page 966
- ▶ Java Diagnostics Profiler UI Navigation and Display Controls on page 968
- ▶ Analyzing Performance Using the Call Profile Window on page 969
- ▶ Thread Call Stack Trace Sampling on page 976
- ▶ Comparison of Collection Leak Pinpointing and LWMD on page 978
- ▶ Object Lifecycle Monitoring on page 980
- ▶ Heap Walker Memory Analysis Execution Steps on page 984
- ▶ Heap Walker Performance Characteristics on page 989

Tasks

- ▶ How to Access the Java Diagnostics Profiler on page 991
- ▶ How to Enable LWMD for Memory Analysis and Collections Displays on page 992
- ▶ How to Enable Allocation Capture on page 993
- ▶ How to Enable Object Lifecycle Monitoring on page 994
- ▶ How to Analyze Object Allocation on page 995
- ▶ How to Enable Heap Breakdown on page 996
- ▶ How to Enable Memory Analysis on page 997

Reference

- ▶ [Java Diagnostics Profiler User Interface on page 999](#)

Concepts

About the Java Diagnostics Profiler

The Diagnostics Profiler for Java is installed with the Java Agent. The Profiler runs in a separate UI and provides near real-time data, enabling you to pinpoint application performance bottlenecks.

Note: The Diagnostics Profiler for Java operates in an unlicensed mode with load restrictions until the probe is able to connect to a Diagnostics Server that has been properly licensed. In unlicensed mode, the Java Profiler is limited to capturing data from five concurrent threads.

For more information about licensing, refer to the *HP Diagnostics Installation and Configuration Guide*.

If you installed the unlicensed trial software agent from the HP Software Web site and you want to use it with a Diagnostics Server, contact HP Software Support to purchase HP Diagnostics.

You can use the different tabs in the Java Profiler to analyze method latency for the selected application. And you can analyze memory problems for the selected application using the memory diagnostics metrics displayed in the Java Profiler.

Special Features Available in the Profiler

Some of the information presented in the Java Profiler is also available in the Diagnostics enterprise UI. However the following features are only available in the Java Profiler. Many of these features are real time and so are enabled and viewed only in the Java Profiler.

- Dynamic instrumentation of a sampled method from the Java Profiler Call Profile (accessible from the Server Requests tab)
- Threads tab

- ▶ Allocation/Lifecycle Analysis tab
- ▶ Heap Breakdown tab (including the heap walker)
- ▶ Probe Configuration tab

How the Java Agent Provides Data for the Java Profiler

This section describes the way in which the Java Agent runs probes to monitor your application and how this data is displayed in the Java Diagnostics Profiler.

Monitoring Method Latency and Call Stacks

The Diagnostics Agent for Java (Java Agent) runs probes to monitor your application and keep track of the metrics for all of the instrumented methods that your application calls. As probes are monitoring, they capture the call stack for the three slowest instances of each server request. The probe also captures a call stack representing all call instances for a type of service request and calculates the aggregated latency

When a server request instance is encountered that is slower than one of the captured instances for the server request, the slower instance replaces one of the previously captured instances.

The Java Diagnostics Profiler displays metrics for all of the instrumented methods. You can drill down to the method instances that are included in the captured call stacks.

While you are analyzing the information displayed on the various tabs of the Java Diagnostics Profiler, you are working with the methods and call stacks captured from the time that the user interface was started. In the meantime, to minimize performance impacts, the probe continues to monitor your application, capture method metrics, and capture call stacks.

Monitoring Application Memory Use

The Java Diagnostics Profiler allows you to monitor your application's memory usage using one of the following methods:

- ▶ Collection Leak Pinpointing

- Light Weight Memory Diagnostics
- Heap Breakdown

Collection Leak Pinpointing allows you to pinpoint Java collection related memory leak locations in Java applications. The data collection for this feature has very low overhead and so it can be used in a production environment.

Light Weight Memory Diagnostics allows you to monitor the collections that your application has created, and to identify the largest collections and the fastest growing collections.

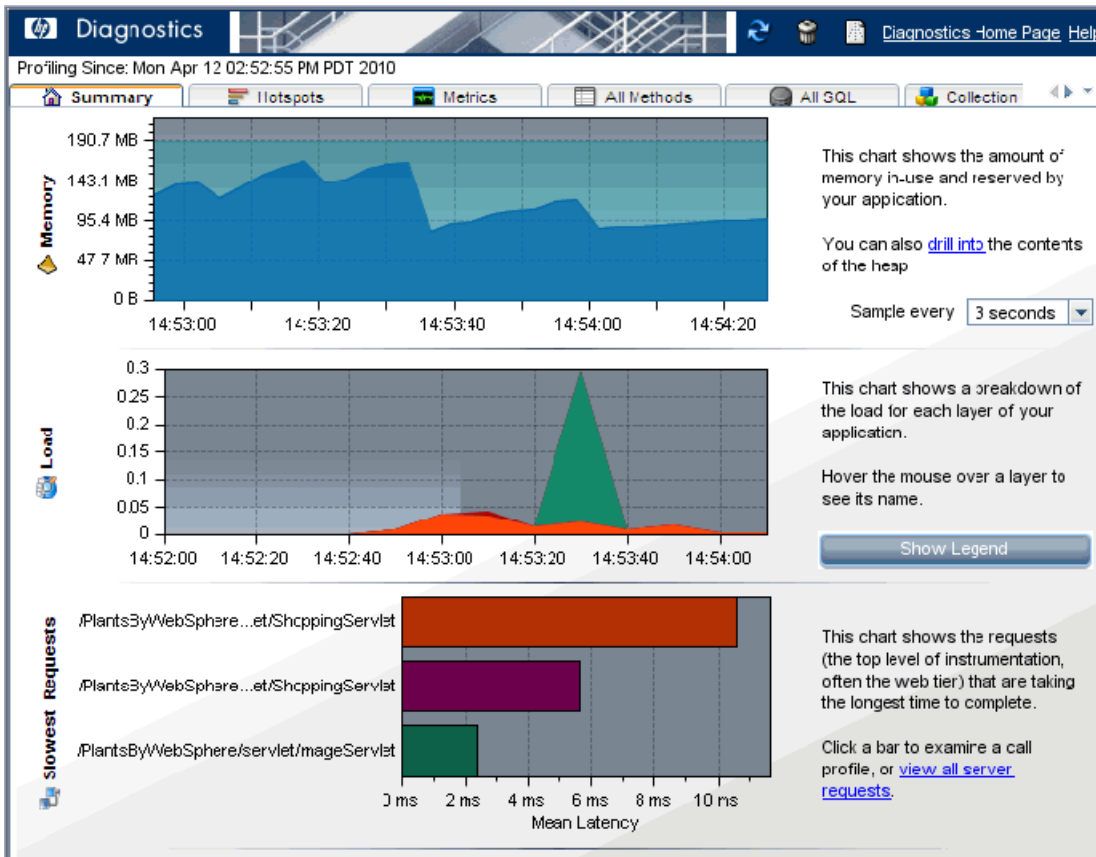
With Heap Breakdown you can monitor the heap generation breakdown and the objects that are stored in heap. This helps you to identify objects that may be leaking. By default, Light Weight Memory Diagnostics and Heap Breakdown are disabled.

For more information see "Comparison of Collection Leak Pinpointing and LWMD" on page 978. Also see "How to Enable LWMD for Memory Analysis and Collections Displays" on page 992.

For more information on Heap Breakdown, see "How to Enable Heap Breakdown" on page 996.

Java Diagnostics Profiler UI Navigation and Display Controls

This section describes the features and controls that are common to the tabs of the Java Diagnostics Profiler:



Refresh Metrics



When you are ready to view more current performance metrics, click **Refresh** on the top right corner of the screen to refresh the information displayed. The Profiler is refreshed with the latest metrics and call stacks. The system does not refresh itself automatically.



Reset Metrics

You can force the Java Diagnostics Profiler to use new baselines for the calculation of instance counts, average latency, and slowest latency, and to force-drop all captured call stacks, by clicking **Reset**.

Note: You may want to reset metrics after your system has warmed up so that the metrics represent processing that takes place when your application is running in a more steady state.

Garbage Collection



When you want to deallocate used memory, you can forcibly perform garbage collection inside the JVM of the probed application by clicking **Garbage Collection** on the top right corner of the screen.

Accessing Help

When you click **Help**, on the top right hand corner of the screen, you access the on-line manual for the Diagnostics Profiler for Java. This same information for the Java Profiler is provided in the online help for the main Diagnostics UI.

Analyzing Performance Using the Call Profile Window

The Call Profile window (accessed from the Server Requests tab) displays a graphical representation of the method call stack for a selected server request. The depicted server request can be an aggregation of all of the calls made to the selected server request or a single instance of the server request depending on the server request on which you drilled down to open the call profile window. The metrics depicted in the graphical representation of the call stack are also depicted in the Call Tree Table on the same tab.

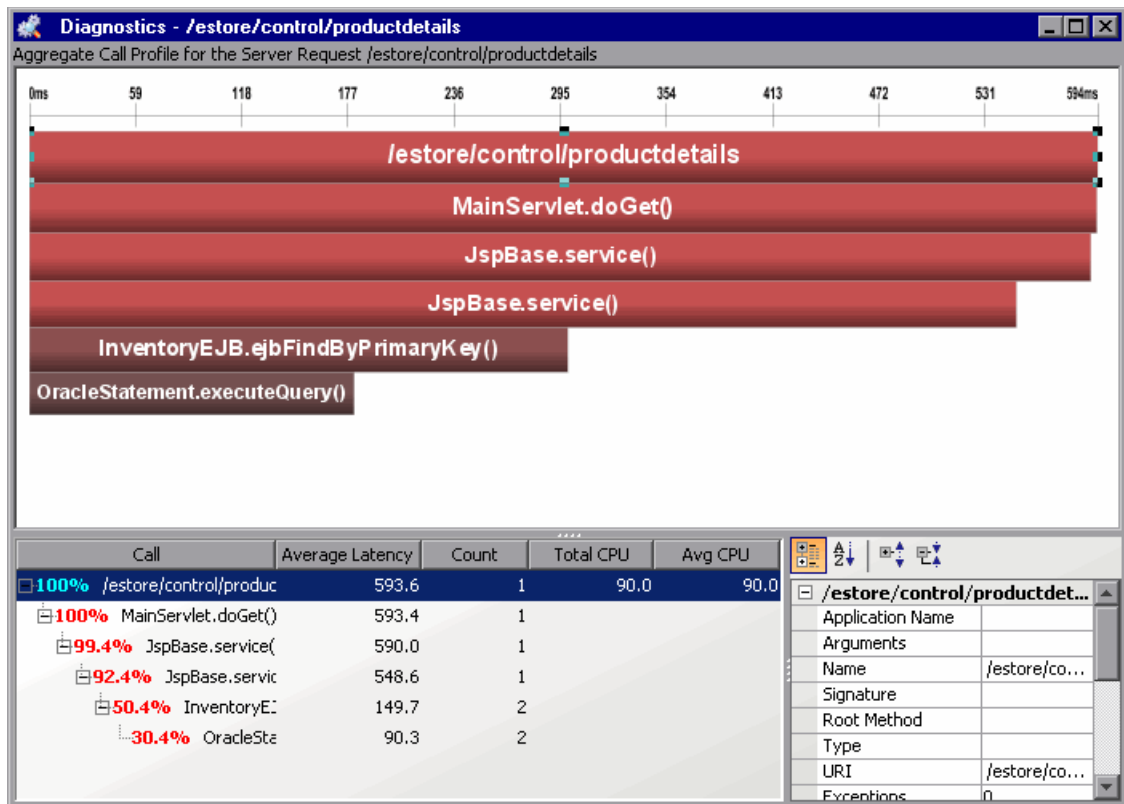
There are two types of call profile windows that are displayed depending on the how you navigated to the tab:

- ▶ **The Instance Call Profile window** displays the method calls that were made during the processing of the server request on which you drilled down.
- ▶ The **Aggregate Profile window** displays an aggregation of all of the method calls that were made during the processing of all of the server requests that were the same as the one on which you drilled down.

The Call Profile Window is made up of three areas:

- ▶ Call Profile Graph
- ▶ Call Tree Table
- ▶ Details Pane

The following image is an example of an Aggregate Call Profile Window showing all three of these areas:



When you click a call box in the Call Profile graph, the corresponding row is selected in the Call Tree table and the metrics for the selected call are displayed in the Details pane. When you click a row in the Call Tree table the corresponding call box in the Call Profile graph is selected and the metrics for the selected call are displayed in the Details pane.

Note: There are differences in the layout and the metrics that are displayed in the Call Profile Window depending on the type of call profile that Diagnostics is displaying. These differences will be noted as each of the areas of the window are described.

Call Profile Graph

The horizontal axis of the Call Profile represents elapsed time, where time progresses from left to right. For aggregated call profiles, the scale across the top of the profile denotes the total time.

For instance call profiles, the calls are distributed across the horizontal axis based upon the actual time when they occurred and so their positions help to show the sequence of each call relative to each other. The scale across the top of the instance call profile denotes the elapsed time since the server request was started.

The vertical axis of the call profile depicts the call stack depth or nesting level. Calls that are made at the higher levels of the call stack are shown at the top of the call profile and those made at deeper levels of the call stack are shown at the lower levels of the profile.

Each call box in the instance call profile represents a method call. The left edge of the box is the start time of the method call and the right edge is the return time from the call. The duration of the call is therefore represented by the length of the box. The position of the call box along the horizontal axis indicates the actual time when the call started and ended. The call boxes that appear directly beneath a call box are the child calls that are invoked by the parent call above them.

The gaps between the call boxes on a layer of the instance profile indicate one of the following processing conditions:

- The processing that took place during the gap occurred in code that is local to the parent at the previous higher level in the call profile and not in child calls in a lower layer.

- The call was waiting to acquire a lock or mutex.
- The processing that took place during the gap occurred in a child call that was not instrumented or included in a capture plan for the run.

The path through the call profile that has the highest total latency is the critical path. Call boxes that are part of the critical path are colored red so that you can identify the methods that make up the critical path. Call boxes that represent calls that are not a part of the critical, high-latency path are colored grey.

If the duration of a call is very short or if the call appears further down in the call stack, the size of the call box can cause the name of the method that the call box represents to become too small to read. You can view the name of the method along with other details for a selected method by holding your pointer over the call box to cause the tooltip to be displayed. You can also see the details for a method selected from the call profile in the Details pane.

The tooltip contains the following details for the selected call box:

Method Detail	Description	Window Type
Method Name	Name of the method represented by the call box.	Aggregate Instance
Layer Name	The name of the Diagnostics layer where the call occurred.	Aggregate Instance
Total Contribution	The percentage contribution to the total latency of the server request that the methods processing contributed.	Aggregate Instance
Call Count	The total number of times that the method was called during the execution of the aggregated server requests instances.	Aggregate

Method Detail	Description	Window Type
Total Latency	The cumulative latency attributed to the processing of the method.	Aggregate Instance
Average Latency	The average latency that can be attributed to each of the method executions for the aggregated server request instances.	Aggregate

Call Tree Table

The **Call Tree** table appears directly below the **Call Profile**. This table shows the same information that is represented in the **Call Profile**.

The first row in the table contains the root of the call stack which is the server request on which you drilled down when the **Call Profile Window** was displayed. The children rows in the tree are the method calls that were made as a result of the server call. The method calls that are parents in the call stack have the expand/collapse control in front of them so that you can control whether the parent's children are displayed or not.

Note: When you click a row call in the Call Tree table, the corresponding box is selected in the call profile graph and the metrics for the selected call are displayed in the Details pane.

The Call Tree Table contains the following columns:

Column Label	Description	Window Type
Call	The name of the Server Request or Method Name. The percentage contribution of the method call to the total latency of the service request precedes the name. The percentage is colored red for those calls which are on the call tree's critical path.	Aggregate Instance
Average Latency	The average latency that can be attributed to each of the method executions for the aggregated server request instances.	Aggregate
Count	The total number of times that the method was called during the execution of the aggregated server requests instances.	Aggregate
Total Latency	The cumulative latency attributed to the processing of the method.	Instance
Total CPU	The total amount of CPU time used by the processing for the selected method or server request.	Aggregate Instance
Average CPU	The average amount of CPU time used by each of the aggregated method calls included in the selected method or server request.	Aggregate

The Total Latency for a parent call includes not only the sum of the latency of each of its children but also the latency for the processing that the method did on its own.

Call Profile Details Pane

The **Details pane** lists the metrics related to the server request or method selected in the Call Profile Graph or in the Call Tree Table.

To view the details of a particular call in the Details pane, select the call from the Call Tree Table or in the Call Profile Graph.

The metrics that are included in a metric category can be hidden or displayed by expanding or collapsing the list of metrics using the plus sign (+) and minus sign (-) next to the category name. Alternatively, you can double-click the category name to expand or collapse the list of metrics.

Thread Call Stack Trace Sampling

When asynchronous thread sampling is enabled you can see additional nodes added into the call profile graph by sampling. These nodes are distinguished by their different (fuzzy) shading to emphasize lack of data about the represented method start and end times. See the *HP Diagnostics Installation and Configuration Guide* chapter on "Advanced Java Agent and Application Server Configuration" for more information on how to use the Profiler UI to configure thread sampling. Also refer to "Asynchronous Thread Stack Trace Sampling" on page 411.

Instrumenting a Sampled Method Dynamically

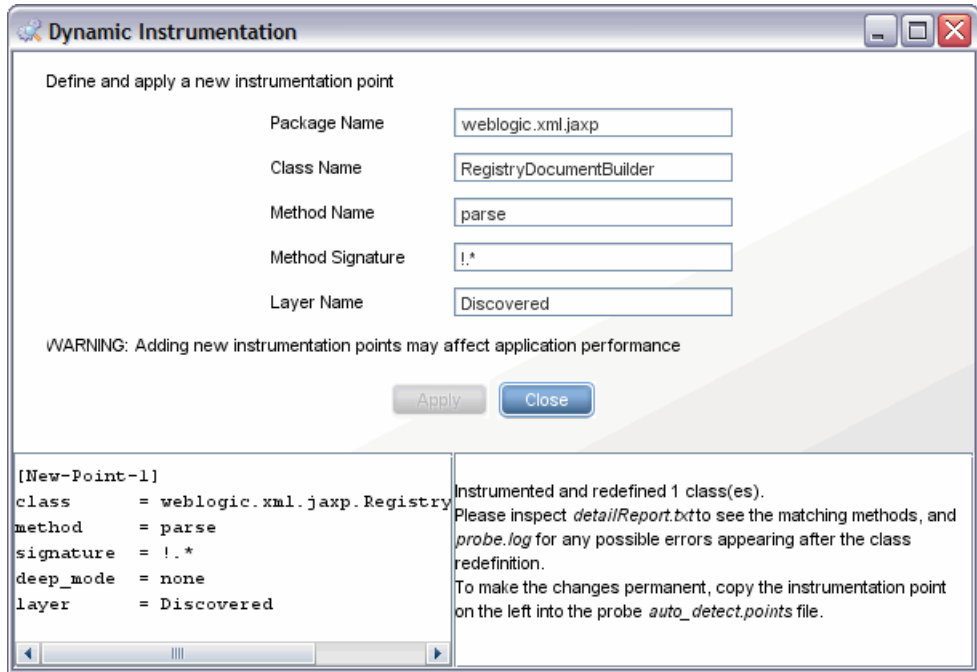
Non-instrumented methods displayed in the Call Profile when Thread Stack Trace Sampling is enabled give you an insight into the call hierarchy and latencies of these methods. But you may want to identify one of these methods to instrument to get additional detail information.

Dynamic instrumentation is Java bytecode instrumentation performed during the application execution *after* the respective class has been first loaded by the Java Virtual Machine. Instrumentation is temporary, for the current Java process. If you want to permanently instrument this method you must add the point you created to the instrumentation points file.

Important: Dynamic instrumentation (the **Instrument** menu item) is ONLY available for sampled data and you need to access the Call Profile from the Diagnostics Profiler for Java. It is NOT available when accessing the Call Profile from an instance tree icon in the main Diagnostics UI.

From the standalone Java Profiler UI, select the Server Requests tab and open the Call Profile window. Select the sampling (fuzzy) node in the Call Profile window and right-click to select **Instrument**.

The Dynamic Instrumentation dialog box is displayed with values corresponding to the selected method.



You can change the package, class, and method name, and provide a method signature, if known, to narrow down the scope of the instrumentation. Since sampling does not reveal method signatures, by default all methods with matching names will be instrumented.

Note: The classes belonging to the Diagnostics Java probe or the Java runtime cannot be instrumented.

Click **Apply** after making the changes you want and the Java probe automatically creates a new point definition and tries to apply the instrumentation dynamically. The bottom part of the dialog window contains the result of this operation: the new instrumentation point definition is placed on the left side, while the result of instrumentation is located on the right.

Once the instrumentation is successful, you should copy and paste the instrumentation point to save it because when you refresh the Call Profile, the Dynamic Instrumentation window with the details on the instrumentation point you created is no longer available.

When you refresh the Call Profile view, the dynamically instrumented method will be displayed as a solid node because it is now instrumented. Instrumentation is temporary for the current Java process.

If you want to permanently instrument this method you must add the point you created to the instrumentation points file.

Comparison of Collection Leak Pinpointing and LWMD

Collection Leak Pinpointing allows you to pinpoint Java collection related memory leak locations in Java applications. Enabling the feature in the probe is optional. Once enabled, the probe will automatically detect and report the leaking Java collection objects and their leak locations (stack traces), without any user interaction. The data collection for this feature has very low overhead and so it can be used in a production environment. See *HP Diagnostics Installation and Configuration Guide* chapter on "Custom Instrumentation for Java Applications" and the chapter on "Advanced Java Agent and Application Server Configuration" for more information on configuring collection leak pinpointing.

Light Weight Memory Diagnostics (LWMD) can also be used to help you locate memory leaks. Enabling LWMD in the probe is optional. User interaction is required to enable LWMD. The data collection overhead for this feature is relatively high and it is not recommended for use in a production environment.

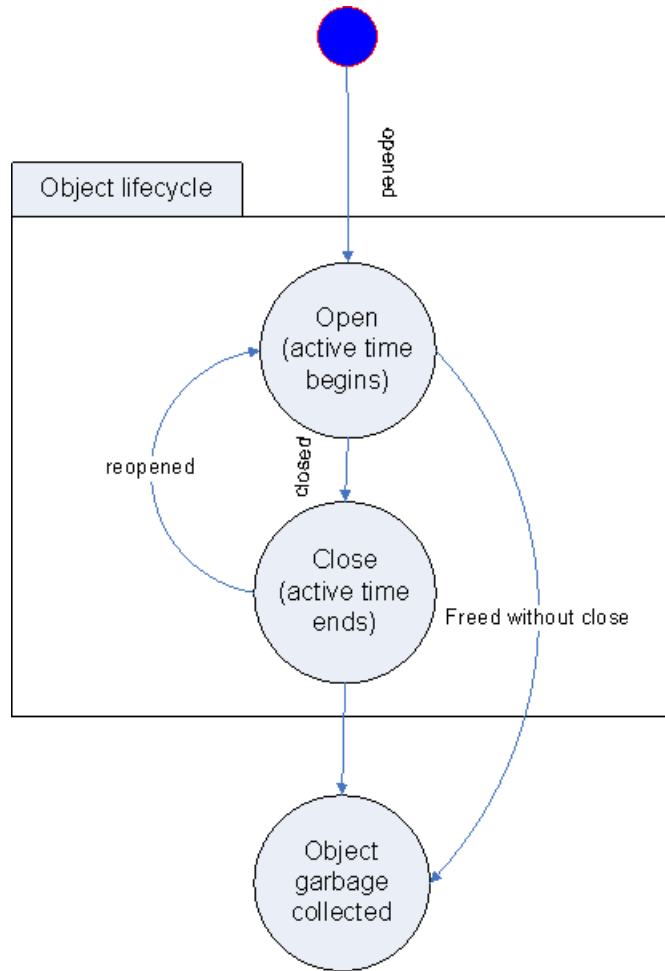
A comparison of Collection Leak Pinpointing and LWMD is shown in the table below. Both are optional features and are used to help detect and locate the Java collection related memory leaks.

	Collection Leak Pinpointing	LWMD
User interaction	Does not need user interaction at all. The probe will automatically detect and report the leaking Java collection objects and their leak locations.	Needs user interaction and manual steps.
Data collection overhead	Very low overhead, can be used in production environment.	Relatively high overhead, depends on the user specified scope. Not recommended to use in production environment.
Out-of-the-box status	Enabled by default	Disabled by default
Instrumentation approach	To use this feature, you need to run the JRE instrumenter to pre-instrument the Java collection classes in the JRE jar file, and add the instrumented JRE classpath to the <code>-Xbootclasspath/p</code> java option to run the probe.	Once the feature is enabled, the application classes within the specified scope will be instrumented at runtime.
Instrumented classes	The Java collection classes (in <code>java.util</code> package and subpackages) in the JRE jar file.	The application classes within the specified scope that have Java collection object allocation.
Common data collected	Both collect full classname of the collection and size of the collection.	
Differences in data collected	Leak location: stack trace when called to add new elements to the leaking collection	The collection object allocation site: class, method, line number

Object Lifecycle Monitoring

Every object has a lifespan. The lifespan begins with object construction and ends with its garbage collection. You can use the Allocation/LifeCycle Analysis tab in the Java Diagnostics Profiler to monitor and analyze object lifespan (see "How to Analyze Object Allocation" on page 995).

However, some objects follow a **lifecycle** during their lifespan. For example, the objects representing database resources (like database connection or cursors) go through such a lifecycle during their lifespan. See the diagram below:



These objects are brought into an *open* state by some resource acquisition operation and then *closed* after their usage. They usually acquire their resources before entering an open state and relinquish their resources after reaching a close state. Some of these objects are designed for re-use (for example, objects based on connection pool). So these objects might be re-opened and closed multiple times during their lifespan.

You can enable object lifecycle monitoring in Diagnostics and view lifecycle information for these objects in the Allocation/Lifecycle Analysis tab.

Two Examples of These Types of Objects

- ▶ **Database connection:** An object of type `java.sql.Connection` represents a database connection. The connection is opened by invoking `javax.sql.DataSource.getConnection()` method and it is closed by invoking `java.sql.Connection.close()` method.
- ▶ **Database cursors:** An object of type `java.sql.ResultSet` represents a database cursor. The cursor is opened by invoking `java.sql.Statement.executeQuery()` method and is closed by invoking `java.sql.ResultSet.close()` method.

Types of Performance Problems with These Objects

Diagnostics allows you to monitor the object's lifecycle between its open and close states to identify the following types of performance problems.

- ▶ **Resources are not released:** This problem arises when the object is not brought into a close state. This causes the resources attached to the object to be wasted for the lifetime of the object.
- ▶ **Resources are not released in a timely manner:** This problem arises when the resources are released after unnecessarily keeping them around for quite long period of time. This can also happen if the object is not closed but the garbage collector automatically closes it during object finalization.

Viewing Object Lifecycle Information

Object lifecycle information is available in the details pane in the Profiler Allocation/Lifecycle Analysis tab for objects enabled for monitoring.

Tips for performance analysis:

The metric **Objects 'Opened' (Total)** shows the number of objects opened during the application's lifetime.

Please note that, if an object is re-opened at multiple location (a common case for pooling), the 'opened' metrics shows the number of times the object was opened. However, the location information refers to the location of the first 'opening' of the object.

Also an object is re-opened without being 'closed' then it is assumed that the object kept itself in the 'open' state. The 'opened' counter is not incremented.

The metric **Objects 'Closed' (Total)** shows the number of objects closed during the application's lifetime. If an object is re-closed without being 'opened' again, then it is assumed that the object kept itself in the 'close' state. The 'closed' counter is not incremented.

The metric **Objects Deallocated without Close** will have a value greater than zero if the resources are not properly released.

The metric **Object Active Lifespan** will have a higher average latency if the resources are not released in timely manner. Note that this metric shows the active lifespan for only those objects that have been closed.

Differences Between Object Lifecycle and Allocation Analysis

Unlike allocation analysis, the object lifecycle feature is not managed. This means that while allocation analysis can be performed by specifically selecting the **Start tracking allocations** and **Stop tracking allocations** links in the Allocation/Lifecycle Analysis tab. Object lifecycle monitoring, if enabled, will show data since the application start-up and the data will not be cleared by the **Clear allocation information** link.

Also, unlike allocation analysis, the object lifecycle feature does not support sampling. This means that all the method calls are captured for the object's lifecycle monitoring.

Heap Walker Memory Analysis Execution Steps

Heap Walker is a memory analysis process accessible from the Memory Analysis tab. You can use it to troubleshoot Java lingering object problems that are difficult to debug or reproduce. Using object tagging and heap snapshots, Heap Walker enables you to inspect individual objects suspected of having “leaked,” and to determine why they are kept alive in the Java heap. This feature targets testing (pre-deployment) environments. You can also use it in production environments.

The steps for using the Heap Walker are described below. The Heap Walker also contains a wizard that guides you through the process of diagnosing a memory leak.

Step 1 - Establishing a Baseline

A typical large Java application allocates many objects during its initialization and warmup. Classes are loaded, thread and database connection pools are populated, and numerous caches in all components are filled. These objects typically stay alive throughout the application execution. To avoid identifying these objects as potential leaks (that is, to avoid false positives), you should let the application run under load for some time to arrive at a stable state.

The application can be placed under memory leak test after initialization has completed, and object allocation has stabilized. Clicking **Start Tracking New Objects** initiates the test operation. After that, any objects allocated by the operation will be tracked as potential leaks.

The assumption here is that the deployed Java application, if allowed to fully initialize, allocate only temporary objects for all of its operations. All temporary objects should eventually be garbage collected. While most server applications comply with this design principle, there are known exceptions to this rule. Database connections, or threads in dynamically sized thread pools, can be created at any time during the application execution without time constraints on when they should be terminated.

The Heap Walker operations may also leave a footprint on the heap. (For example, some probe classes are loaded and initialized only when you start using Heap Walker.) Footprints should not be a problem if you are aware of them. However, if you need a clear picture, it is recommended that you perform the execution twice, treating the first pass as a warm-up only. You can ignore results from the first pass.

Step 2 - Exercising the Operation

The details of this step may differ, depending on whether the application is running in a production environment or in a test environment. In a test environment, the application owner can carefully stage the test load to contain only the desired operations. For example, testing can focus on newly developed code, or objects that are suspected of leaking memory based on the analysis of the logs or feedback from the IT center where the application is deployed.

It is often useful to use such an operation under test in some kind of a context. For example, if the application requires a user logon, it might be practical to wrap the tested operation by a logon and logout. It is typical for the application to hold the active session information in the heap. In this case, you can dismiss the session information only after a logout.

Alternatively, you can perform the logon before new object tracking is started. In any case, you should arrange the tested operation in such a way that it leaves no permanent footprint in Java memory (adding records to a database is fine). The tested operation can be repeated several times. In the case of simple leaks, a single execution is usually enough. In a production environment, it is impossible to control the load, or to time the new object tracking by starting and stopping to catch only the desired portion of the load. You need to take this into account when analyzing the results.

Heap Walker can display the number and size of the currently tracked objects. These numbers are updated by taking a heap snapshot. You observe the numbers as they change over time. Measurements increase as the application allocates new objects. They decrease as the objects are garbage collected. After the tested operation is complete (or, in the case of a production environment, sufficient time has elapsed), you can click **Stop Tracking New Objects**. At this point, the set of tracked objects is closed. It can no longer grow.

It is normal, however, for several tracked objects to still be alive at this point. They can be present in numerous caches in the application, including the components that you do not own. It is also possible that some tracked objects require finalization. The finalizers are run periodically by the JVM, typically asynchronously to the activities controlled by the application. Objects pending finalization are considered alive, even though the application may hold no references to them.

Step 3 - Flushing Application Caches

Under normal circumstances, if the application remains under load, the caches clear of all the tracked objects eventually, and the pending finalizers run eventually. The JVM also runs garbage collection periodically. This garbage collection removes the tracked object from the heap, provided they are not leaks.

You can sometimes speed up this cleaning process by forcing garbage collection. Clicking **Run Garbage Collection** makes the JVM not only run the full GC cycle, but also run the pending finalizations.

Taking heap snapshots is especially useful at this point. The observed number and the total size of tracked objects should go down over time, as the cache flushing process progresses. Ideally, these numbers should eventually reach zero, meaning that all tracked objects have been garbage collected.

However, if there is a Java memory leak in the tested operation, the numbers stabilize at some non-zero values, and no longer decrease, despite repeated garbage collections and continuous load on the application. When you decide that the tracked objects remaining on the heap should be considered a leak, it is time to capture the **object reference graph**. This action dumps all references present in the heap to a file, and starts an additional (external) process, which sorts the file. The file is used in the next steps.

Step 4 - Analyzing Potential Leaks

After you capture the object reference graph, you can retrieve the list of tracked objects. In most cases, you select just one class of objects to retrieve. This can be accomplished by double-clicking the row with the selected class. It is also possible to retrieve objects for multiple classes. Simply select multiple rows (by holding down the Ctrl key), and then right-click to select **Inspect Selected Tracked Objects**.

For efficiency of operation, there is a limit on the total number of objects that can be retrieved. You can change the limit, using the selector located on the left side of the window. Retrieving a large number of objects rarely makes sense, as it is costly, and it does not necessarily increase your capability to solve the leak problem.

Step 5 - Walking the Heap

You can determine why any of the retrieved objects is alive by clicking the table row describing the object. This action displays an **Object Reference Diagram**. This diagram shows the selected object with a chain of references that are keeping the object alive, and indicates which object is a heap root.

For any object already displayed, it is possible to show all objects directly referencing it by double-clicking the object.

As above, to keep a limit on the overhead, there is a limit selector on the left side of the screen controlling the maximum number of objects to be retrieved and displayed.

An example of an Object Reference Diagram is shown below:

Profiling Since: Tue Oct 27 10:56:01 AM PDT 2009

Memory Analysis

Heap Walker
Step 5 - Walking the Heap

This table is a list of tracked objects of the selected class in the captured object reference graph.

To view a path from a root to a specific object, select it in the table. Any path of references from a root to an object will keep the object alive, making it impossible for the Garbage Collector to remove it from the heap.

You can further explore the graph by double-clicking nodes to view all referers to each node.

Limit the number of objects to: 50

Available Actions:
 Capture Object Reference Graph
 Take another Heap Snapshot
 Return to Step 1

Last Action:
 Path to root found in 6064ms

Class	Bytes	Obj Id
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2603019
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2769445
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2603183
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2766805
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2602887
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2767757
weblogic.server.channels.ServerConnectionRuntimeImpl	24	2766796

All displayed references (links between objects) are based on the captured object reference graph. Additional information, such as object type, size, or reference names are retrieved directly from the heap. Under some circumstances, the additional information cannot be retrieved because some of the objects keeping the specific object alive can cycle over time and be garbage collected. A continuously growing `java.util.Vector` object is a good illustration of this point, as the underlying array is replaced over time.

The objects are color-coded according to their age. There are three distinct object ages:

- **Baseline.** Objects allocated before new object tracking was started.
- **Tracked.** Objects allocated between new object tracking start and new object tracking stop, ostensibly by the tested operation.

- **Fresh.** Objects allocated after new object tracking was stopped.

The toolbar selections in the Object Reference Diagram are similar to the toolbar in topology views, so for toolbar details see "The Topology Toolbar" on page 249.

You may elect to capture a new Object Reference Diagram to obtain a fresh view of the object, and repeat "Step 4 - Analyzing Potential Leaks" on page 987.

Heap Walker Performance Characteristics

Technically, starting or stopping new object tracking, and capturing the object reference graph, uses the JVM heap tagging operations. It may require substantial execution time, which can be up to several minutes for very large heaps. The application is practically paused during this time. Do not use Heap Walker if the nature of the deployed application cannot tolerate such long pauses. If in doubt, always test Heap Walker first in a test environment.

The above steps, and in particular starting new object tracking, also make the JVM allocate extra memory generally proportional to the current heap size. This memory is allocated outside of the Java heap, but within the JVM process. You need to take special care to ensure that such memory can be allocated. Keep in mind that the JVM itself, the application code, the JIT-compiled code, and any native libraries used by the application must fit into this space as well. For 32-bit processes, there is an operating system-dependent limit on the size of the process address space (for example, 2GB for Windows on Intel x86). If almost half (or more) of the available address space is already reserved by the Java heap, the tagging operation can crash the JVM.

Heap Walker gives you the total memory usage estimate when the **Start New Object Tracking** operation is activated for the first time. The estimate is for total system memory. It is based on additional memory needed by the JVM and on memory for the object references sorting program. At this point, you have a chance to quit Heap Walker without affecting the deployed application negatively (no additional memory is allocated). Obviously, in a

production environment (deployed application), it is recommended that you use Heap Walker only if the system capacity is large enough to handle the additional memory pressure. The decision whether to continue with tagging depends not only on the total amount of memory available on the system running the application, but on the impact of a possible JVM crash on the business process as well.

When using Heap Walker in a test environment, it is usually possible to scale down the load and the maximum heap size to match the system capacity. There is no direct CPU overhead on the Java application, other than actually running a Heap Walker command (indicated by the progress bar). This also includes the tracking period. That is, even though tracking start and tracking stop consume large amounts of CPU time, there is no overhead while actually tracking new objects.

However, the increased memory footprint of the JVM may cause serious sluggishness if the JVM no longer fits into main memory, and makes excessive use of the swap area. If, after having tagged the heap, you notice severe application performance degradation while none of the Heap Walker operations are running, you most likely have a swap file thrashing problem.

Tasks

How to Access the Java Diagnostics Profiler

Once you have installed the Java Agent, configured a probe to collect performance data and started the application that is being monitored, you can access the Java Diagnostics Profiler from your browser and view Diagnostics data. You can also access the Java Diagnostics Profiler by drilling down from the views of the HP Diagnostics user interface.

To open the Java Diagnostics Profiler directly (standalone):

- 1 In your browser, go to the Java Diagnostics Profiler URL: http://<probe_host>:<probeport>/profiler.

The probes are assigned to the first available port beginning at **35000**.

Note: You can find the port that a particular probe is using in the probe's **probe.log** file located in **<probe_install_directory>/log/<probe_id>** directory. In the **probe.log** file, find the line that begins with the words **webserver listening on**, for example: **webserver listening on 0.0.0.0:35003**. The port is the number after the colon, in this example 35003.

- 2 Type your username and password.

You are prompted to enter a username and password. The default username is **admin**. The default password is **admin**. You may be prompted again to enter a username and password. Re-enter the same details.

For more information about authentication and usernames and passwords when you have the full Diagnostics product, refer to the *HP Diagnostics Installation and Configuration Guide* section on Authentication and Authorization.

To drill down to the Diagnostics Java Profiler from the main Diagnostics UI:

- 1 From any view in HP Diagnostics that shows probe entities, right-click the probe in the table and select **View Profiler for <probe name>** from the menu.
- 2 If the Profiler fails to open when performing the drill down from the Diagnostics UI, ensure that you have set a default browser within your operating system.

How to Enable LWMD for Memory Analysis and Collections Displays

This task describes how to enable Lightweight Memory Diagnostics (LWMD) for use in analyzing memory leaks.

By default, LWMD is disabled, so the Java Agent does not impose the additional overhead on its host when you are not going to use memory diagnostics metrics. When you detect a memory leak using the Heap Breakdown tab, you can enable LWMD. When you have completed your investigation, you can disable LWMD once more.

Note: LWMD must be enabled in order for you to see any data in the Collections tab of the Java Diagnostics Profiler.

To enable LWMD:

- Turn on the LWMD capture in the **dynamic.properties** file by setting the **lwm.diagnostics.capture** property equal to **true**:

```
lwm.diagnostics.capture=true.
```


- Activate the LWMD point in the **auto_detect.points** file by setting **active** equal to **true** and indicate the scope of the LWMD instrumentation:

```
[Light-Weight Memory Diagnostics]
keyword = lwmd
scope = !only\in\this\Class\.*;!or\in\this\Class\.*
active=true
```

It is very important to limit the scope of the LWMD instrumentation to a particular package to reduce overhead. The syntax for the scope starts with an exclamation point (!) to indicate that a regular expression follows.

How to Enable Allocation Capture

This task describes how to enable allocation capture for the probe.

The Allocation/Lifecycle Analysis tab cannot display allocation objects or their metrics until allocation capture has been enabled for the probe. By default, allocation capture is disabled, so the Java Agent does not impose the additional overhead on its host when you are not going to use memory diagnostics metrics. If you suspect that you may have a memory issue with the way your application manages its object allocations, you can enable allocation capture. When you have completed your investigation, you can disable the allocation capture again.

To enable allocation capture to view data in the Allocation/LifeCycle Analysis Tab:

- 1 In the **auto_detect.points** file located in **<probe_install_directory>\etc**, modify the default settings to match the following:

```
[Allocation]
keyword = allocation
detail = leak
scope = !com\mycompany\mycomponent\.*
active = true
```

If you want to have reflective allocation tracked, you can add the reflection attribute to the detail argument in the Allocation point.

```
[Allocation]
keyword = allocation
detail = leak,reflection
scope = !com\.mycompany\.mycomponent\..*
active = true
```

This instruments the **Class.newInstance**, **Constructor.newInstance**, and **Object.clone** methods. The reflection instrumentation tracks all classes that are created.

- 2 Restart the monitored application, so the probe restarts and can apply the updated instrumentation.

How to Enable Object Lifecycle Monitoring

This task describes how to enable the monitoring of certain types of objects.

To enable object lifecycle monitoring to monitor object lifecycle data in the Allocation/LifeCycle Analysis tab:

- 1 Object lifecycle monitoring in Diagnostics is not enabled by default. The resource monitoring of certain types of objects can be individually enabled in the **etc/inst.properties** file.

For example, to enable the database cursor monitoring set `mercury.enable.resourcemonitor.jdbcResultSet=true` for `details.conditional.properties` property in the **inst.properties** file. This enables object lifecycle monitoring for all resources of this type for a single probe.

- 2 You will need to restart the probe after making changes to the `etc/inst.properties` file.
- 3 Due to higher overhead of caller side instrumentation and possibly large number of objects (resources) to be tracked, it is recommended that this feature is only enabled during development stage. It should be enabled in production environment with great caution and with a very limited 'scope'.

You specify the scope in the object lifecycle monitoring section in the `auto_detect.points` file.

How to Analyze Object Allocation

This task describes how to analyze the object allocations your application is performing.

After you have identified a memory problem using the Heap Breakdown tab, you can analyze the object allocations that your application is performing by examining the allocations while the suspected application functionality is being executed. The following procedure describes how to run an experiment and study the resulting application performance.

To analyze object allocations:

- 1** If you have not already enabled allocation capture for the probe, do so as instructed in "How to Enable Allocation Capture" on page 993.
- 2** Begin tracking allocations by selecting **Start Tracking Allocations** from the Common Tasks menu.

The probe starts collecting the metrics for the objects that are being allocated and de-allocated. No collection metrics are displayed in the tab until you select the **Refresh Allocation Information** or **Stop Tracking Allocations** menu options.

- 3** Execute the application functions that you suspect may be causing a leak, so any objects that are allocated while performing the function can be tracked.
- 4** Select the **Stop Tracking Allocations** menu option to limit the tracked objects to those that were captured while the suspect application functions were being performed.

No additional instances are tracked after you stop tracking. The instances of the objects that were already allocated continue to be tracked as they are de-allocated, so the metrics on the tab can be refreshed with accurate counts of the objects that are alive or de-allocated, as well as with accurate object lifespans.

- 5 Select the **Refresh Allocation Information** menu option to update the tab with the current metrics for the allocated objects.

Each time you select this menu option, the Profiler updates the metrics for the tracked objects in the allocations analysis table with the current counts and lifespans. The trend lines for the metrics in the graph are updated to chart the data points for the metrics at the refresh time.

You should repeat this step as your application continues to run, so you can see what happens to the allocated objects over time.

- 6 If you want to run your experiment again, select the **Clear Allocation Information** menu option to clear the table and graph of all of the objects and metrics currently displayed, and begin this process again from the second step.

How to Enable Heap Breakdown

This task describes how to enable heap breakdown for memory analysis.

Heap breakdown is used when your application is running a JVM version earlier than JVM 1.5. If your application is running JVM 1.5 or higher (see "How to Enable Memory Analysis" on page 997).

To enable Heap Breakdown and display the Heap Breakdown view:

- 1 Add the files in the **directory** `<probe_install_dir>\lib\<platform>` to the OS environment path.

Note: Use the environment variable that is appropriate for your operating system. In Windows, the environment variable is **PATH**.

- 2 Add **-Xrunheapdump** to the command line that starts the JVM and the application server.

How to Enable Memory Analysis

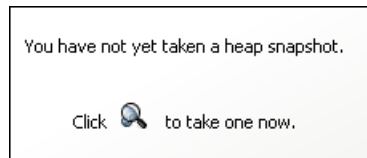
This task describes how to enable memory analysis.

If your application is running a JVM version earlier than JVM 1.5 you enable heap breakdown instead (see "How to Enable Heap Breakdown" on page 996).

By default, the Memory Analysis tab is disabled, so the Java Agent does not impose additional overhead on its host when you do not need the memory diagnostics metrics.

To enable advance memory analysis and display the Heap Walker views:

- 1** You must have Java 1.5 or higher installed, and use the new `-agentpath:<probe_install_dir>/lib/<platform_dir>/jvmti.dll` parameter instead of `-Xrunheapdump` in the application startup script. Replace `jvmti.dll` with the appropriate library name if you run the probe on a non-Windows system.
- 2** Open the Java Diagnostics Profiler for the application, and click the **Memory Analysis** tab.
- 3** Click the icon to take a heap snapshot and open the first Heap Walker view.



Note: You cannot use Heap Walker when running your application with HotSpot 5.0 JVM with CMS enabled (the `-XX:+UseConcMarkSweepGC` option). Remove this option from the Java command if you plan to use Heap Walker.

When both the **-server** and the **-Xgc:parallel** options are selected, some versions of JRockit 5.0 JVM demonstrate instability. In some configurations, both options are selected by default. In such cases, specify the **-client** or the **-Xgc:gencon** option to override the default. This is a known BEA issue (CR334327) and should be resolved in future releases of JRockit.

Reference

Java Diagnostics Profiler User Interface

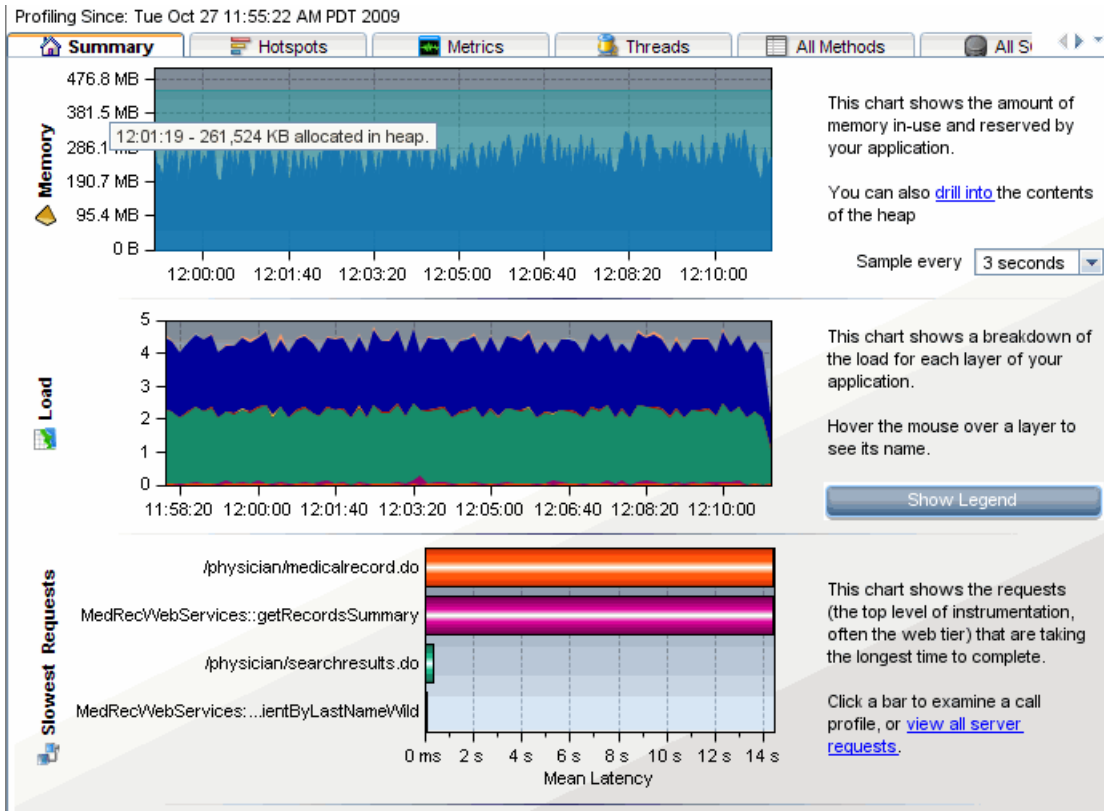
This section includes:

- ▶ Summary Tab Description on page 1000
- ▶ Hotspots Tab Description on page 1002
- ▶ Metrics Tab Description on page 1004
- ▶ Threads Tab Description on page 1007
- ▶ All Methods Tab Description on page 1012
- ▶ All SQL Tab Description on page 1015
- ▶ Collection Leaks Tab Description on page 1016
- ▶ Collections Tab Description on page 1020
- ▶ Exceptions Tab Description on page 1024
- ▶ Server Requests Tab Description on page 1026
- ▶ Web Services Tab Description on page 1029
- ▶ Allocation/LifeCycle Analysis Tab Description on page 1032
- ▶ Heap Breakdown Tab Description on page 1036
- ▶ Memory Analysis Tab Description on page 1038
- ▶ Configuration Tab Description on page 1042

Summary Tab Description

The Summary tab consists of graphs that display information about the memory in use and reserved by your application, the load for each layer of your application and the slowest requests made to your application server.

The following is an example of the Java Profiler Summary Tab display.



To access	In the Java Diagnostics Profiler, select the Summary tab.
Important information	If the message 'Profiling not in progress' is displayed, select the Begin Profiling link in the upper left corner.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

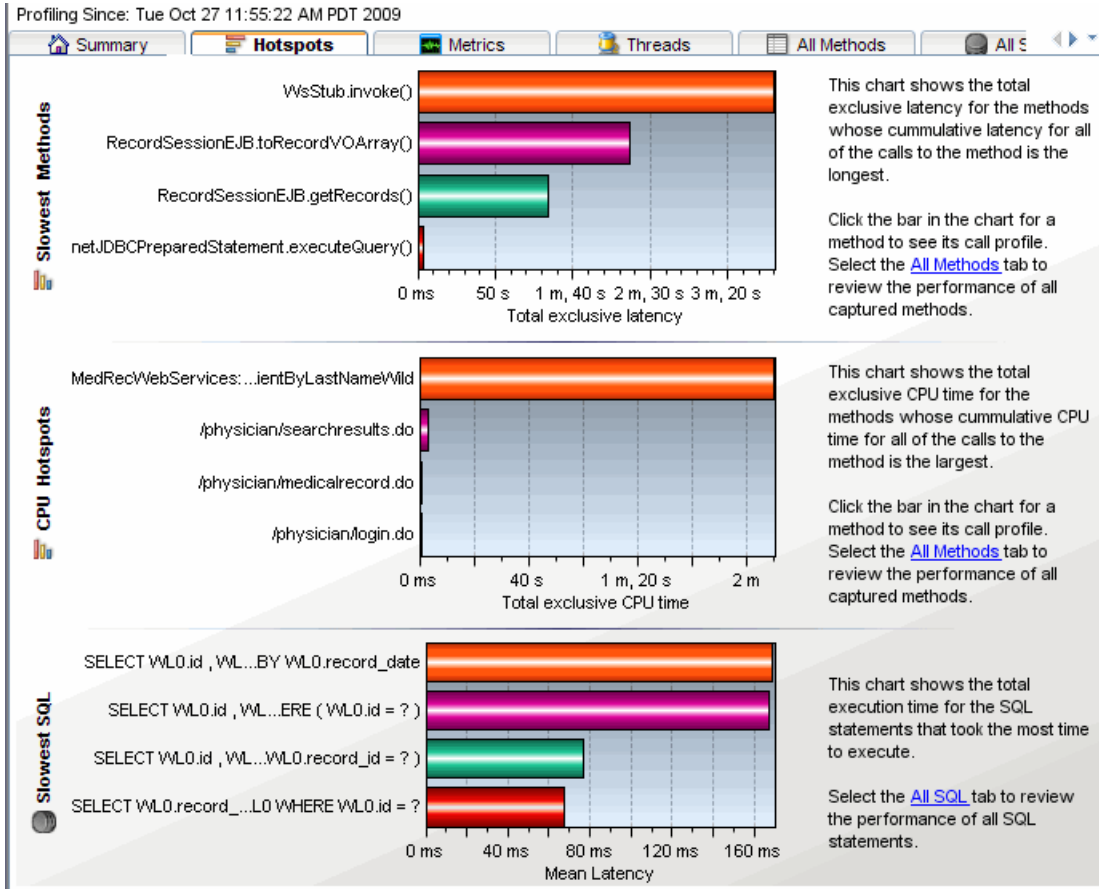
The following user interface elements are included:

UI Elements	Description
Memory Graph	<p>The Memory graph shows the amount of memory allocated in your application and the amount of memory (JVM heap size) reserved by your application.</p> <p>You can see more details about the exact amount of allocated memory or reserved memory in your application, by holding your pointer over various points on the graph to view the tooltip.</p>
Load Graph	<p>The Load graph shows the breakdown of the load for each layer of your application.</p> <p>The performance metrics for classes and methods are grouped into layers based upon the resources that the application invokes to perform the processing. The Java Diagnostics Profiler displays the layers on one level and does not split them into sublayers.</p> <p>You can see the name of each layer by holding your pointer over various points on the graph to view the tooltip.</p> <p>To view a legend of the graph that displays the names of all the layers, click Show Legend.</p>
Slowest Requests Graph	<p>The Slowest Request graph shows the server requests that are taking the longest time to complete.</p> <p>To view the aggregated call profile for a server request in the Slowest Request graph, click the bar for the server request. For more information about the call profile window, see "Analyzing Performance Using the Call Profile Window" on page 969.</p>
Information Pane	<p>The information pane at the bottom of the window displays the following information:</p> <ul style="list-style-type: none"> ▶ The date and time of the last time you refreshed the Profiler data. ▶ The probe ID.

Hotspots Tab Description

The Hotspots tab displays bar charts of the significant metrics that have been captured during the monitoring of your application.

The following is an example of the Java Profiler Hotspots Tab display.



To access

In the Java Diagnostics Profiler, select the Hotspots tab.

Important information	You can view the details for a graphed metric by holding your pointer over the bar for the metric and viewing the tooltip.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

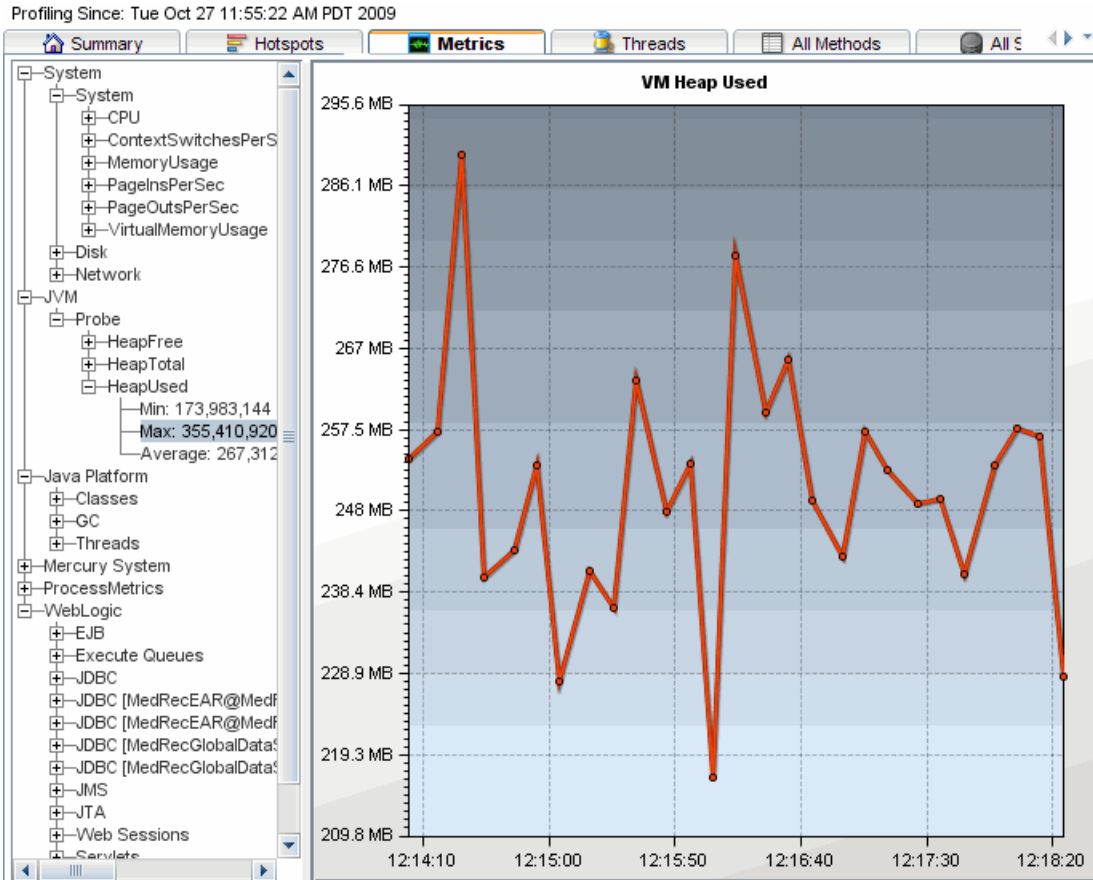
The following user interface elements are included:

UI Elements	Description
Slowest Methods Graph	<p>This chart shows the method calls that are taking the most time exclusively in that method. To view the call profile for a selected method call in the Slowest Methods graph, click the bar for the method. For more information about the call profile window, see "Analyzing Performance Using the Call Profile Window" on page 969.</p> <p>If the method is part of more than one server request, when you double-click the method, the following dialog box opens and asks you to select the particular server request for which you want to see the call profile.</p> <p>Double-click the appropriate server request row to view the call profile.</p>
CPU Hotspots Graph	<p>This chart shows the methods that are using the most CPU.</p> <p>To view the call profile for a particular method, click the bar for the method. For more information about the call profile window, see "Analyzing Performance Using the Call Profile Window" on page 969</p>
Slowest SQL Graph	<p>This chart shows the SQL statements that are taking the most time.</p> <p>To view the SQL statement details for a particular statement in the Slowest SQL graph, click the bar for the SQL statement to select it. For more information about SQL statement details, see "Analyzing Performance Using the Call Profile Window" on page 969.</p>

Metrics Tab Description

The Metrics tab displays information about the Operating System, the JVM and the application server.

The following is an example of the Java Profiler Metrics Tab display.



To access	In the Java Diagnostics Profiler, select the Metrics tab.
Important information	<p>When more than one probe is running on the same host, the System metrics only appear for the probe for which you opened the profiler.</p> <p>If you are using the Profiler without the Diagnostics product, then to preserve memory in the application server, metrics are only measured from the time you access the graph. However, if the probe is connected to a Diagnostics Server, the metrics are measured continuously, regardless of whether you have accessed the graph.</p>
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

UI Elements	Description
Tree Pane	<p>Displays the metrics in an expandable tree.</p> <p>The top three levels displayed in the tree are:</p> <p>System. Metrics about the Operating System</p> <p>JVM. Metrics about the JVM</p> <p><application server>. Metrics about the application server. Depending on the environment, the application servers that will be displayed are WebLogic, WebSphere, or SAP.</p> <p>When you expand each of the top levels, the tree displays the associated metrics for each top level. As you further expand each metric, you arrive at a minimum, a maximum and an average numerical value for each metric.</p>
Graph Pane	<p>Displays a graph of the metrics selected from the tree pane.</p> <p>When you click a specific metric in the tree, the graph pane displays a graph representing the selected metric. You can select more than one metric to display in the graph pane using the Control or Shift keys.</p> <p>The x-axis in the graph represents time. The y-axis in the graph represents the numerical value of the metric. Metrics are displayed for the last five minutes unless the probe is working with another HP Software product, in which case they are displayed for three hours.</p>

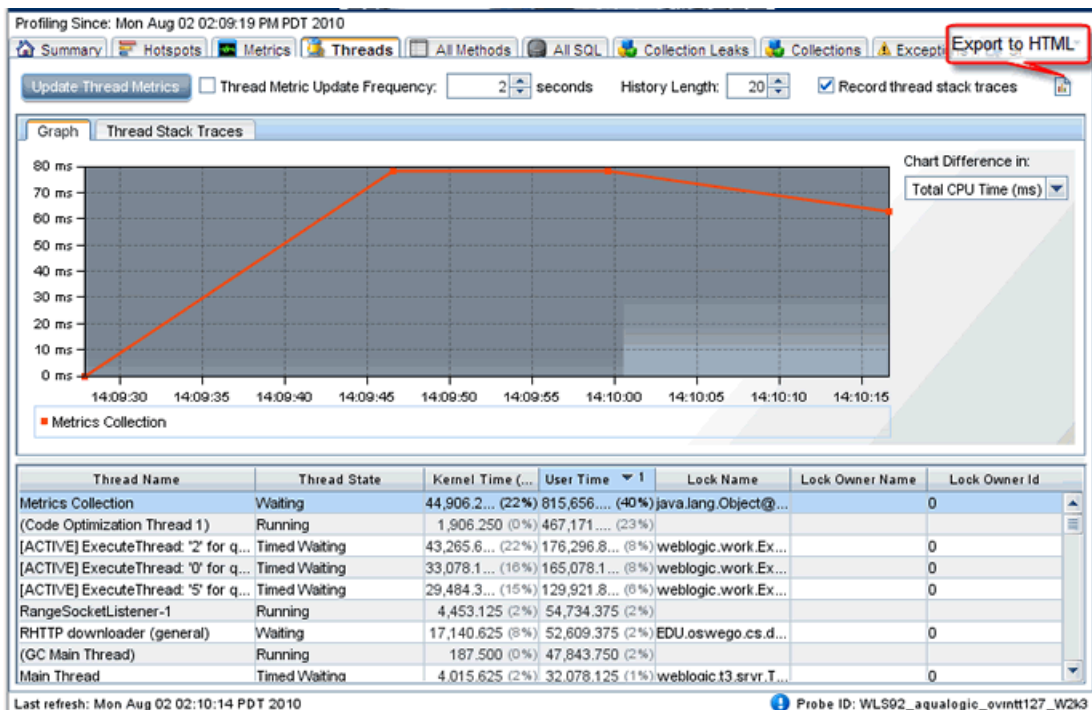
Threads Tab Description

The Threads tab displays thread performance metrics for the Java threads that are captured by the probe and provides a way for you to capture stack traces for the captured threads.

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

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

The following is an example of the Java Profiler Threads Tab display.



To access	In the Java Diagnostics Profiler, select the Threads tab.
Important information	This Threads tab is only displayed when the application is running on JDK 1.5 and above.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

UI Elements	Description
Controls	<p>Used to control how often the thread metrics are updated, whether stack traces are captured for each thread, and what kind of data is displayed for the thread processing in your application.</p> <p>When the Threads tab is updated, the information displayed on the tab is refreshed with the latest thread metrics. If you are capturing stack traces, a stack trace is saved each time that the Profiler updates the metrics on the tab. You control how often the Profiler updates the thread metrics on the Threads tab.</p> <p>Update Thread Metrics. Check this box and the Profiler refreshes the information in the graph and the thread table and captures stack traces.</p> <p>Thread Metric Update Frequency. Check this box to turn automatic updates on. Select the update interval from the spinner. The Profiler immediately begins refreshing the thread metrics displayed in this tab based on the update interval specified.</p> <p>By default, whenever the Profiler updates the Threads tab display, stack traces are captured for each of the threads listed in the thread table. You can control how many stack traces for each thread are displayed in the stack trace history and can turn off the capture of stack traces.</p> <p>History Length. Select the number of stack traces from the spinner. The Profiler lists the indicated number of stack traces for the selected threads on the Thread Stack Traces tab.</p> <p>Record thread stack traces. Check this box to turn thread stack trace collection on or off.</p> <p>Export to HTML. Select this button to export the thread graph and data to an HTML file.</p>

UI Elements	Description
Thread Table	<p>Lists the metrics for each thread.</p> <p>The following columns are displayed:</p> <p>Thread Name. The name of the captured thread.</p> <p>Thread State. The state of the thread at the last thread metric update interval.</p> <p>Kernel Time. The portion of the CPU time during which the thread was executing in kernel mode.</p> <p>User Time. The portion of the CPU time during which the thread was executing in user mode.</p> <p>The table can also include columns for Thread Waited and Blocked Time metrics if you enable them. To enable these metrics, set the threads.contention.monitoring.enabled property to true in the <code><probe_install_dir>/etc/probe.properties</code> file. This setting may cause slightly higher probe overhead.</p>

UI Elements	Description
Graph Tab	<p>Charts the metrics for the selected threads.</p> <p>When you start capturing thread metrics, the Profiler charts the thread with the highest User Time in the thread table by default. You can control which metrics are charted in the graph. You may chart the metrics for one or more of the threads listed in the threads table and you can select the metric that is to be charted for each thread.</p> <p>Chart difference in. To select a metric to be charted for each thread, select the metric from the drop down menu. Diagnostics updates the graph to chart the indicated metric for each of the threads selected in the thread table.</p> <p>To select the threads that are to have their metrics charted, select the row in the thread table that contains the thread whose metrics you would like to chart.</p> <p>Diagnostics removes the metrics for any previously charted threads from the graph and charts the metrics for the selected thread. The graph legend is updated to indicate the color with which the selected thread's metrics were charted.</p> <p>To chart additional metrics in the graph along with any that you have already charted, select additional threads in the thread table.</p> <p>To select each additional thread one at a time, select each row in the thread table using Ctrl-Click.</p> <p>To select a range of threads, select the row in the thread table using Shift-Click. Diagnostics charts the metrics for the selected thread along with the metrics for all of the threads in the thread table that are between the selected threads and the newly selected thread. The graph legend is updated to indicate the colors with which the selected threads metrics were charted.</p> <p>To remove the metrics from the chart for selected threads, use Ctrl-Click to select the row in the thread table that contains the thread whose metrics you'd like to remove from the chart.</p>

UI Elements	Description
Thread Stack Traces Tab	<p>Stack traces for the threads select from the threads table are displayed when you have indicated that you want thread stack traces captured.</p> <p>The Stack Trace Details for drop down allows you to control which thread's stack traces the Profiler displays in the Stack Trace details area.</p> <p>When you select All Threads, the stack traces for all threads are displayed in the stack trace details area. The selections made in the threads table do not impact the stack traces that are displayed in the stack trace details area when All Threads is selected.</p> <p>When you select Selected Threads, the stack traces displayed in the stack trace details area are limited to those for the threads that you select in the threads table for the stack trace capture selected in the stack trace capture list.</p> <p>The Thread Stack Traces tab display is divided into two areas:</p> <p>Captured Stack Traces. List contains a list of the times when stack trace captures occurred.</p> <p>Stack Trace Details. Displays the stack traces that you indicated based on your selections from the stack trace capture list, the scope selection drop down, and the thread table.</p>

All Methods Tab Description

The All Methods tab lists the method calls that your application makes according to the instrumentation in the `auto_detect` points file.

The following is an example of the Java Profiler All Methods Tab display.

Profiling Since: Tue Oct 27 11:55:22 AM PDT 2009

Summary Hotspots Metrics Threads **All Methods** All SQ

Show Only Outbound Method Calls

Method Name	Total ti...	Avg time...	Count	Excepti...	Total CPU...	Avg CPU...	Layer ...
MedRecWebServices::getRecordsSummary	230,003.0	14,375.2	16	0		0.0	CrossVM
com.bea.medrec.value.Record[] com.bea.medrec...	136,608.9	8,538.1	16	0		0.0	Business ...
com.bea.medrec.value.Record[] com.bea.medrec...	84,449.3	5,278.1	16	0		0.0	Business ...
java.sql.ResultSet com.pointbase.net.net.JDBCPre...	3,553.3	142.1	25	0		0.0	Database/...
String com.bea.medrec.entities.RecordEJB.toStrin...	2,716.2	55.4	49	0		0.0	Business ...
MedRecWebServices::findPatientByLastNameWild	2,148.5	134.3	16	0		0.0	CrossVM
com.bea.medrec.webservices.client.MedRecWe...	2,124.1	132.8	16	0		0.0	Business ...
com.bea.medrec.webservices.client.PhysicianWV...	1,380.6	86.3	16	0		0.0	Business ...
void weblogic.wsee.server.servlet.BaseWSServ...	1,044.1	16.3	64	0		0.0	Web Tier/S...
com.bea.medrec.value.Patient[] com.bea.medrec....	1,043.8	94.9	11	0		0.0	Business ...
java.sql.PreparedStatement com.pointbase.net.ne...	935.6	233.9	4	0		0.0	Database/...
void com.bea.medrec.filters.RequestEncodingFilt...	539.0	6.7	80	0		0.0	Web Tier/S...
String com.bea.medrec.entities.RecordEJB.presc...	311.1	31.1	10	0		0.0	Business ...
java.sql.Connection weblogic.jdbc.jta.DataSource...	261.5	261.5	1	0		0.0	Database/...
com.bea.medrec.value.RecordsSummary com.be...	251.2	15.7	16	0		0.0	CrossVM
java.sql.ResultSet weblogic.jdbc.wrapper.Prepar...	232.9	2.6	88	0	21.5	0.3	Database/...
com.bea.medrec.value.Patient[] com.bea.medrec....	221.6	13.9	16	0		0.0	Web Servi...
java.sql.ResultSet weblogic.jdbc.wrapper.Statem...	209.1	5.8	36	0	22.5	0.6	Database/...
void org.apache.struts.action.ActionServlet.doGe...	184.2	5.8	32	0		0.0	Web Tier/S...
org.apache.struts.action.ActionForward org.apa...	171.7	10.7	16	0		0.0	Web Tier/S...
org.apache.struts.action.ActionForward com.bea...	162.0	10.1	16	0		0.0	Web Tier/S...

Double-click a method to view that method in Aggregate Call Profile(s). All times shown are exclusive (not including time spent in profiled chil...

To access	In the Java Diagnostics Profiler, select the All Methods tab.
Important information	All CPU times shown are exclusive (not including time spent in profiled children). All of the metrics in the All Methods tab are counted from the time you enter the system or click the Reset button
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

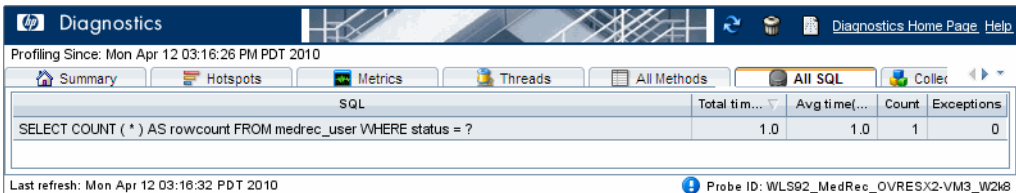
UI Elements	Description
Table	<p>The All Methods tab displays a table that contains the following columns:</p> <p>Method name. The names of the methods that were called. The Method name has the following syntax: <code><package name>.<class name>.<method name></code>.</p> <p>Total Time. The aggregate latency for all of the calls to the method. The total latency is shown in milliseconds.</p> <p>Avg Time. The average latency for all of the calls to the method. The average latency is shown in milliseconds.</p> <p>Count. The number of times that the method was invoked.</p> <p>Exceptions. The number of times that the method generated an exception.</p> <p>Total CPU. The total amount of CPU time that all invocations of the listed method used.</p> <p>Avg CPU. The CPU time that the method used during an average invocation.</p> <p>If CPU time metrics are not being displayed, CPU Timestamp collection for methods can be configured.</p> <p>Layer. The Layer associated with this method according to the instrumentation in the <code>auto_detect points</code> file. The layers are displayed on one level and there is no distinction made between layers and sub-layers.</p> <p>To view the call profile for a method call, double-click the appropriate row. For more information about the call profile see "Analyzing Performance Using the Call Profile Window" on page 969.</p>

UI Elements	Description
Table (continued)	<p>If the method is part of more than one server request, when you double-click the method, a dialog box opens for you to select the relevant server request.</p> <p>To create call profiles from more than one server requests select the first server request with a single click and select subsequent server request using control click. When you have finished making your selections, click OK to instruct the Profiler to create the call profiles. The call profile for each selected server request is displayed in a separate window.</p>

All SQL Tab Description

The All SQL tab displays the SQL statements in a table.

The following is an example of the Java Profiler All SQL Tab display.



SQL	Total tim...	Avg time(...	Count	Exceptions
SELECT COUNT (*) AS rowcount FROM medrec_user WHERE status = ?	1.0	1.0	1	0

To access	In the Java Diagnostics Profiler, select the All SQL tab.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

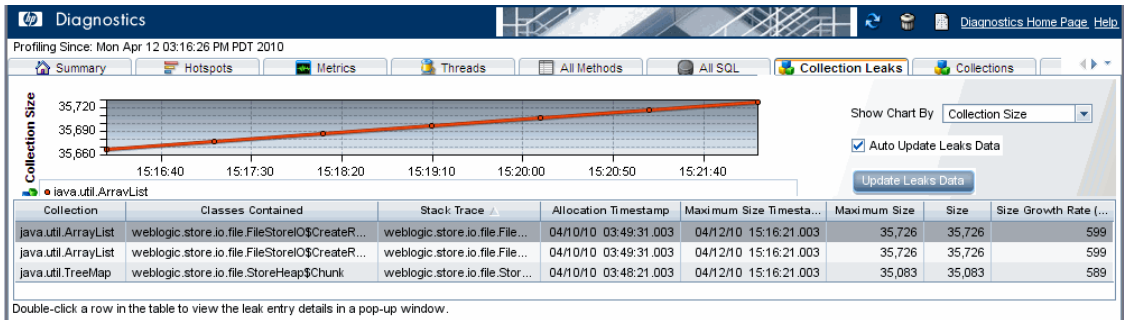
The following user interface elements are included:

UI Elements	Description
Table	<p>The All SQL tab displays the SQL Statement table, which contains the following columns.</p> <p>SQL. The name of the SQL statement that was invoked by the application server.</p> <p>Total Time. The total latency of all invocations of the SQL statement.</p> <p>Avg Time. The average latency of all invocations of the SQL statement.</p> <p>Count. The number of times the SQL statement was invoked by the application server.</p> <p>Exceptions. The number of times that the statement generated an exception.</p> <p>To view the SQL statement details, double-click the relevant statement. The SQL statement details dialog box opens, displaying all the information shown in the SQL table for each statement.</p> <p>The SQL statement details dialog box enables you to view the full string of the SQL statement and to copy the text.</p>

Collection Leaks Tab Description

The Collection Leaks tab displays information on the probe's currently leaking collection objects in a table and a chart of collection size or collection size growth.

The following is an example of the Java Profiler Collection Leaks Tab display.



To access	In the Java Diagnostics Profiler, select the Collection Leaks tab.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991
Important Information	<p>Collection leak pinpointing must be enabled for this view, but note that it is enabled by default. The [Collection Leak Pinpointing] keyword = clp point in the probe's etc/auto_detect.points file is set to active by default.</p> <p>For this feature you also need to enable Collection Leak Pinpointing (CLP) instrumentation by running the JRE instrumenter to pre-instrument the Java Collection classes in the JRE your application/application server will run with; and copy the java parameter to include them in your java options.</p>
See also	See <i>HP Diagnostics Installation and Configuration Guide</i> chapter on "Custom Instrumentation for Java Applications" and the chapter on "Advanced Java Agent and Application Server Configuration" for more information on configuring collection leak pinpointing and for how to enable/disable and configure CLP reporting.

The following user interface elements are included:

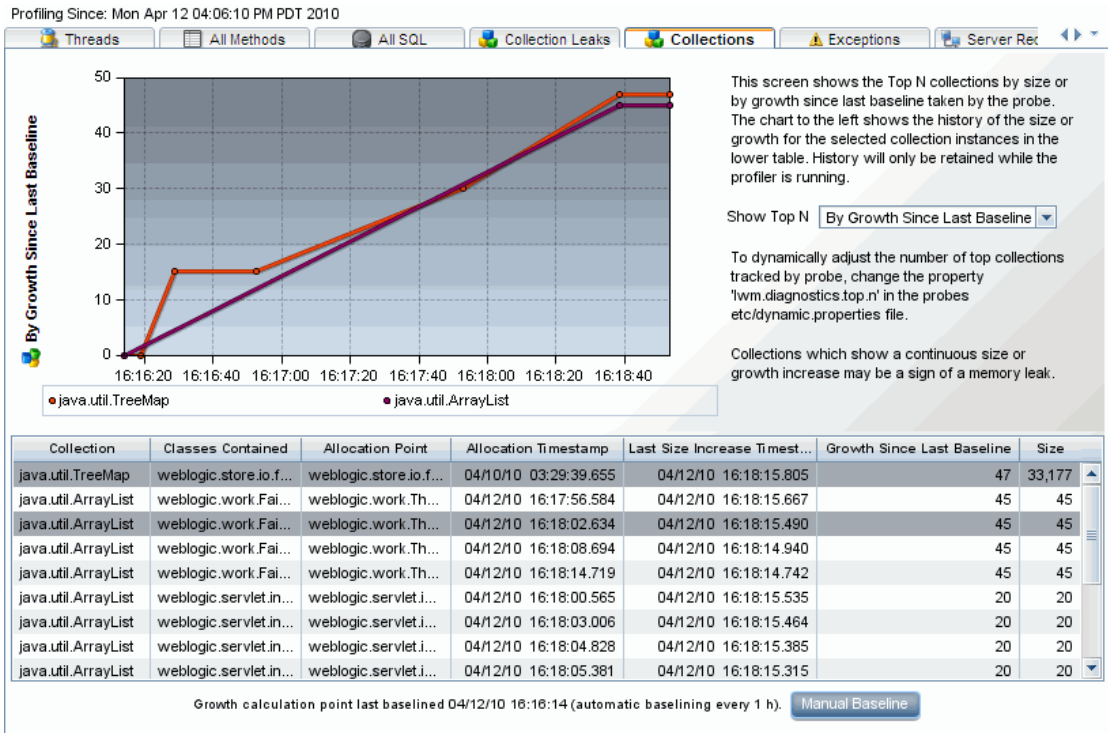
UI Elements	Description
Collection Leak graph	<p>When you click the row in the collections table, the graph is updated to show a trend line for the collection leak. The trend line shows either the Collection Size Growth, or the Collection Size, depending on the selection you make from the Show Chart By drop down list.</p>
Collection Leaks Table	<p>The collection table lists the probe's currently flagged leak collection objects. The collections can be sorted by various columns in the table.</p> <p>Check the Auto Update Leaks Data checkbox to automatically update the data display. Click the Update Leaks Data button to update the data.</p> <p>The Collections Table contains the following columns:</p> <p>Collection. The collection type.</p> <p>Classes Contained. The type of the objects contained within the collection. If there are multiple types of objects found within the collections, the value in the table appears as Unknown.</p> <p>Stack Trace. Leak location stack trace.</p> <p>Allocation Timestamp. The time at which the collection was allocated.</p> <p>Maximum Size Timestamp. The time when the maximum size was captured.</p> <p>Maximum Size. The maximum size of the collection ever observed by the Java agent (in number of elements).</p> <p>Size. The average size of the collection (in number of elements).</p> <p>Size Growth Rate (per hour). The average growth rate for the collection, measured over the period of time since the collection creation until now (in number of elements per hour)</p> <p>To view the collection leak details, double-click the relevant collection and a dialog box opens with the collection leak details including stack trace information.</p>

Collections Tab Description

The Java Diagnostics Profiler monitors your applications' memory usage with Light Weight Memory Diagnostics (LWMD). LWMD monitors the memory used by your applications by tracking collection objects.

The Collections tab shows the metrics for the collections in your application in a graph and corresponding table. The table lists the collections, information about the allocation of the collections, and the metrics for their growth rate and size. The graph contains the metrics charted for the collections that you selected. The growth rate of the collections are calculated from a baseline. The Profiler updates the baseline periodically. If you want, you can update it manually.

The following is an example of the Java Profiler Collections Tab display.



To access	In the Java Diagnostics Profiler, select the Collections tab.
Important information	LWMD must be enabled to view data in the Collections tab and do Memory Analysis using the Heap Breakdown.
Relevant tasks	"How to Enable LWMD for Memory Analysis and Collections Displays" on page 992

The following user interface elements are included:

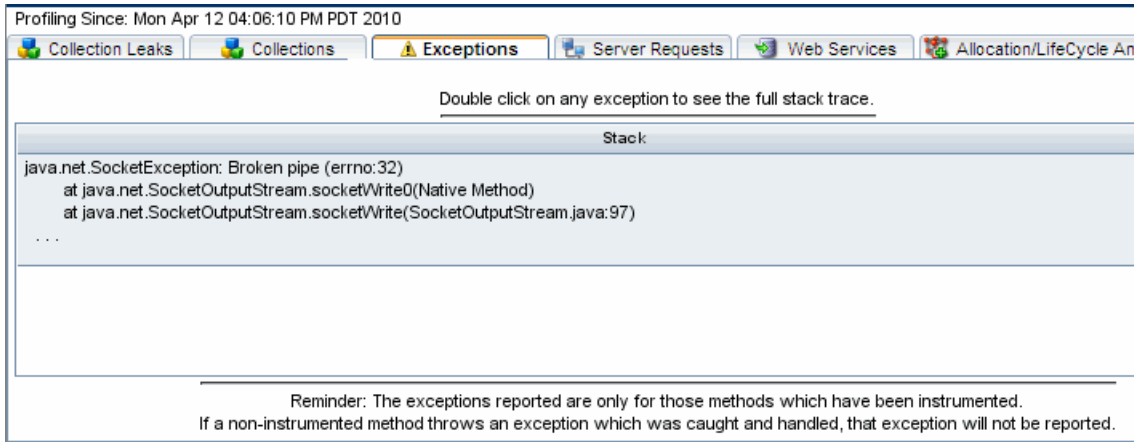
UI Elements	Description
Collections Table	<p>The collection table lists the collections. The collections are sorted by either the amount of growth since the last baseline, or by the size of the collection, depending on the selection you make from the Show Top N box to the right of the graph.</p> <p>Your selection from the Show Top N box controls the metrics that are charted in the collections graph as well as the sort order of the rows in the collections tables.</p> <p>When you choose By Size, the collection table is sorted in descending order by collection size. The size metrics for the selected collections are charted in the collections graph.</p> <p>When you chose By Growth in Last Baseline, the collection table is sorted in descending order by the amount of growth in the collection since the last baseline. The growth metrics for the selected collections are charted in the collections graph.</p> <p>The Collections Table contains the following columns:</p> <p>Collection. The collection type.</p> <p>Classes Contained. The type of the objects contained within the collection. If there are multiple types of objects found within the collections, the value in the table appears as Unknown.</p> <p>Allocation Point. The location where the collection is allocated in the code.</p> <p>Allocation Timestamp. The time at which the collection was allocated.</p> <p>Last Size Increase Timestamp. The last time that a size increase was captured.</p> <p>Growth Since Last Baseline. The increase or decrease in the number of objects within the collection since the last baseline.</p> <p>Size. The number of objects in the collection.</p>

UI Elements	Description
Collections Graph	<p>When you click the row for a collection in the collections table, the collections graph is updated to chart either the size or the growth of the collection since the last baseline, depending on the selection you make from the Show Top N box to the right of the graph. You may chart the metrics for more than one of the collections by selecting subsequent rows with a CTRL-click.</p>
Baseline Information	<p>The baseline determines the time from which the growth in the size of the collections is measured. You can view the time that the last baseline was set at the bottom of the Collections display.</p> <p>The Profiler automatically sets a new baseline at preset periodic intervals. You can also set a new baseline manually.</p> <p>To set a new baseline manually, click Manual Baseline. The Profiler resets the Growth Since Last Baseline metric for each collection, and refreshes the charted metrics in the graph.</p> <p>By default, a new baseline is set automatically every hour. You can change the automatic baselining interval in the dynamic.properties file.</p> <p>You do not need to stop the application server when you change the automatic baselining interval.</p> <p>You can change the automatic baselining interval in the <probe_install_dir>\etc\dynamic.properties file. Locate the line: lwm.diagnostics.auto.baseline.interval=60m.</p> <p>Change the time interval according to your needs as explained in the comments of the file.</p> <p>If you want to stop automatic baselining, enter 0 for the time interval.</p>

Exceptions Tab Description

The Exceptions tab displays all the exceptions that were generated in the application server for methods that have been instrumented.

The following is an example of the Java Profiler Exceptions Tab display.



Profiling Since: Mon Apr 12 04:06:10 PM PDT 2010

Collection Leaks Collections **Exceptions** Server Requests Web Services Allocation/LifeCycle An

Double click on any exception to see the full stack trace.

Stack

```

java.net.SocketException: Broken pipe (errno:32)
  at java.net.SocketOutputStream.socketWrite0(Native Method)
  at java.net.SocketOutputStream.socketWrite(SocketOutputStream.java:97)
  . . .
    
```

Reminder: The exceptions reported are only for those methods which have been instrumented.
If a non-instrumented method throws an exception which was caught and handled, that exception will not be reported.

To access	In the Java Diagnostics Profiler, select the Exceptions tab.
Important information	If a non-instrumented method throws an exception which was caught and handled, that exception will not be reported.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

UI Elements	Description
Table	<p>The Exceptions tab displays the Exceptions table which contains the following columns:</p> <p>Stack. Shows the first three lines of the exception stack trace.</p> <p>Count. The number of times the exception was generated.</p> <p>To see the full stack trace of the exception, double-click the row containing the exception to open the Exception dialog box.</p>

Server Requests Tab Description

The Server Requests tab displays information about the server requests made to the application server.

The following is an example of the Java Profiler Server Requests Tab display.

Profiling Since: Mon Apr 12 03:16:26 PM PDT 2010

Filter by Server Request Type

Server Request	Total time(ms) ▾	Avg time(ms)	Count	Avg CPU(ms)
/physician/medicalrecord.do	4,235,928.5	33,887.4	125	20.8
MedRecWebServices::getRecordsSummary	4,218,816.9	33,750.5	125	16,260.9
/physician/searchresults.do	35,249.2	282.0	125	91.2
Background - Database	14,764.3	3.7	3,989	0.0
MedRecWebServices::findPatientByLastNameWild	8,364.0	66.9	125	16.8
/physician/login.do	2,337.1	9.4	248	4.7
/ws_medrec/MedRecWebServices	763.3	6.1	125	4.9
/physician/search.do	641.8	5.2	124	0.4
Static Content	621.2	0.6	992	0.3
/ws_phys/PhysicianWebServices	429.8	3.4	125	2.8
Background - Directory Service	355.7	0.5	684	0.0
RmiDataSource.getConnection()	334.9	0.5	684	0.1
Static Content	242.6	0.3	744	0.2
BasicNamingNode.lookup()	193.0	0.3	684	0.0

Select a row to view the worst instances for that Server Request, or Double-click a row to view the Aggregate Call Profile for that Server ...

To access	In the Java Diagnostics Profiler, select the Server Requests tab.
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

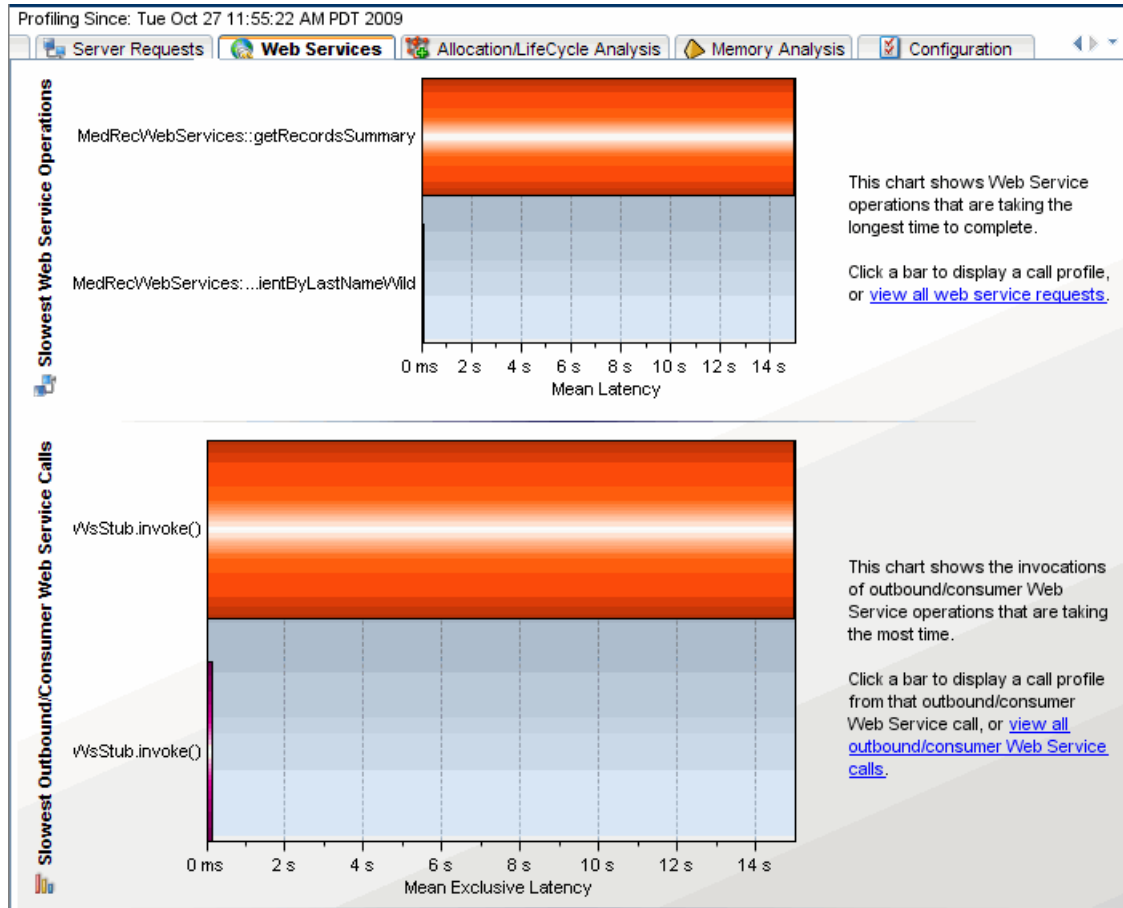
UI Elements	Description
<p>Table (Server Requests)</p>	<p>The server request table at the top of the display lists the aggregated performance information for all instances of the server requests.</p> <p>When you select a server request in this table by clicking the row, a table at the bottom of the tab is populated with the three server request instances that have the worst total time.</p> <p>When you double-click a server request in this table, the Profiler displays the call profile for the selected aggregated server request in a new window. For more information about the call profile window, see "Analyzing Performance Using the Call Profile Window" on page 969.</p> <p>The aggregated Server Requests table contains the following columns:</p> <p>Server Request. The URI or the root method for the server request. The URI parameters are trimmed. To break down server requests according to URI parameters, contact Customer Support.</p> <p>Total Time. The total latency of all invocations of the server request.</p> <p>Average Time. The Average latency of all invocations of the server request.</p> <p>Count. The number of times this server request was invoked.</p> <p>Avg CPU. The CPU time that the method used during an average invocation.</p> <p>If CPU time metrics are not being displayed, CPU Timestamp collection for methods can be configured.</p> <p>Layer. Displays the layer for server requests that were invoked by root methods that are not part of an HTTP request. HTTP server requests do not have a layer.</p>

UI Elements	Description
Table (Slowest Instances)	<p>When you click a server request, the bottom section of the window displays a table containing the three slowest instances of the server request.</p> <p>The table contains the following columns:</p> <p>Server Request. The name of the server request.</p> <p>Start Timestamp. Point in time when the server request instance was invoked.</p> <p>End Timestamp. Point in time when the server request ended.</p> <p>Total Time. Total amount of time the server request took to execute.</p> <p>Threw Exception. Indicates whether or not an exception was thrown during the processing of this server request instance.</p> <p>To view the instance call profile for an instance of a server request, double-click a server request instance. For more information about the call profile see "Analyzing Performance Using the Call Profile Window" on page 969.</p>

Web Services Tab Description

The Web Services tab contains graphs displaying the slowest Web service operations (inbound Web service calls) received and processed in your monitored environment and the slowest outbound Web service calls made from within your monitored environment.

The following is an example of the Java Profiler Web Services Tab display.



To access	In the Java Diagnostics Profiler, select the Web Services tab.
Important information	Web service operations and calls are displayed in the graphs, in the following format: <Web-service-name>::<operation-name>. For example, MedRecWebServices::getRecordsSummary .
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

UI Elements	Description
<p>Slowest Web Service Operations Graph</p>	<p>The Slowest Web Service Operations graph displays the slowest Web service operations (inbound Web service calls) received and processed in your monitored environment.</p> <p>The Java Diagnostics Profiler displays Web service operations as a type of server request.</p> <p>You can view the call profile for a Web service operation displayed in the graph, by clicking the bar representing the relevant Web service operation. For more information about the call profile window, see "Analyzing Performance Using the Call Profile Window" on page 969.</p> <p>You can view a list of all the Web service operations in the Server Requests tab, by clicking the view all web service requests link to the right of the graph. For more information about the Server Requests tab, see "Server Requests Tab Description" on page 1026.</p>

UI Elements	Description
<p>Slowest Outbound Web Service Calls Graph</p>	<p>The Slowest Outbound Web Service Calls graph displays the slowest outbound Web service calls made from within your monitored environment.</p> <p>The Java Diagnostics Profiler displays outbound Web service calls as remote calls within a server request.</p> <p>You can view the call profile for the server request containing a particular outbound Web service call displayed in the graph. To view the call profile, click the bar representing the relevant Web service call. For more information about the call profile window, see "Analyzing Performance Using the Call Profile Window" on page 969.</p> <p>If the remote call is part of more than one server request, when you double-click the method, a dialog box opens and asks you to select the relevant server request. Double-click the appropriate server request row to view the call profile.</p> <p>You can view all the outbound Web service calls in the All Methods tab, by clicking the view all outbound Web service calls link to the right of the graph. For more information about the All Methods tab, see "All Methods Tab Description" on page 1012.</p>

Allocation/Lifecycle Analysis Tab Description

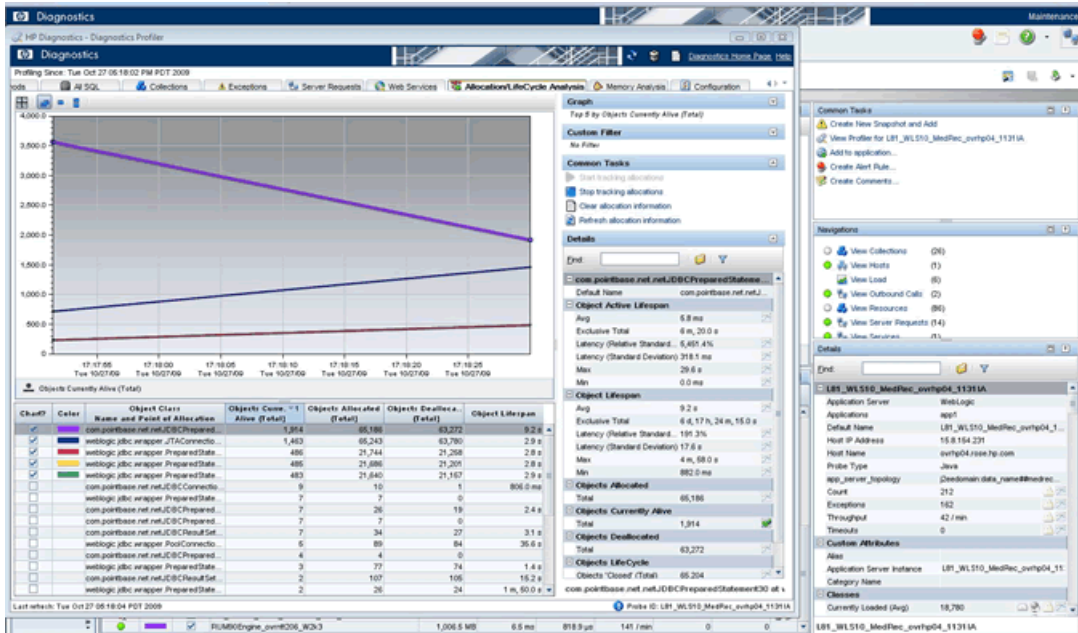
The Allocation/Lifecycle Analysis tab shows the metrics for the objects that have been allocated by your application in a graph and a corresponding table. The table lists the allocated objects, along with the number of allocated instances and their lifespan. The graph contains the charted metrics for the selected allocated objects.

To analyze allocations, you must use the controls in the Common Tasks menu to track allocations and refresh the displayed metrics as you exercise the application functionality that you believe may be experiencing leaks.

The Allocation/Lifecycle Analysis tab can be used for:

- **Allocation Analysis.** Use the information displayed to investigate a memory leak that you have observed in the Heap Breakdown tab by examining the allocation and de-allocation of objects while the leak is happening.
- **Lifecycle Analysis.** Use the information displayed to monitor object lifecycles. This feature can be used for resource monitoring of certain database resources.

The following is an example of the Java Profiler Allocation/LifeCycle Analysis Tab display.



To access	In the Java Diagnostics Profiler, select the Allocation/LifeCycle Analysis tab.
Important information	The Allocation/Lifecycle Analysis tab is similar to the views with a detail layout in the Diagnostics views. Instead of appearing in the view title, the view filters appear in view filter menus, along the side of the graph. The Common Tasks menu controls the tracking of the allocations as well as the refreshing of the information that is displayed for the entire view.
Relevant tasks	Allocation capture must be enabled to view allocation data. See "How to Enable Allocation Capture" on page 993. Object lifecycle monitoring must be enabled to view object lifecycle data. See "How to Enable Object Lifecycle Monitoring" on page 994.

The following user interface elements are included:

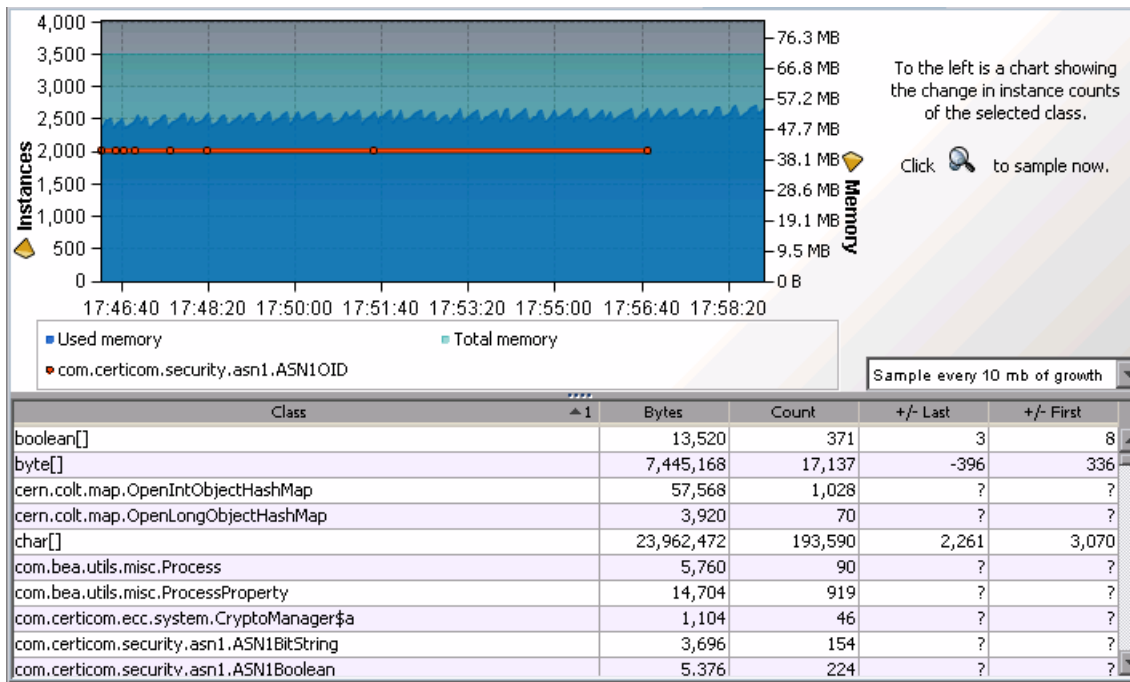
UI Elements	Description
Object Table	<p>The object table lists the objects that have been allocated since you started tracking allocations. You can customize this table to adjust the sort order and the columns that appear in the table, just like the graph-entity tables in other views with the detail layout. By default, the table is sorted in order by Objects Currently Alive. It displays the following columns:</p> <p>Chart. Allows you to indicate if the metrics for the allocated object are to be charted in the graph. You can select objects to be charted by clicking on the box in this column manually. Or you can let the Profiler select the objects to chart dynamically, using the criteria that you specify in the Graph filter.</p> <p>Color. Indicates the color that the Profiler uses to chart the metrics for the allocated object. No color is shown for metrics that are not charted.</p> <p>Objects Currently Alive. A count of the total number of allocated objects that have not yet been garbage collected.</p> <p>Objects Allocated. A count of the total number of objects that have been allocated whether they have been garbage collected or not.</p> <p>Objects Deallocated. A count of the total number of objects that have been garbage collected.</p> <p>Object Lifespan. The average duration of the life of all de-allocated objects. If no objects have been de-allocated, this column is blank.</p> <p>In the Details pane, metrics for Objects Lifecycle and Object Active Lifespan are also available. See "Object Lifecycle Monitoring" on page 980 for more information.</p>
Graph	<p>The graph charts the metrics that you selected from the details table for each of the objects selected in the allocation/lifecycle analysis table.</p> <p>Using the controls in the views with a detail layout, you control which metrics are charted and which entities have their metrics charted.</p>

🔗 Heap Breakdown Tab Description

The Profiler displays the Heap Breakdown tab instead of the Memory Analysis tab when your application is running a JVM version earlier than JVM 1.5.

The Heap Breakdown tab allows you to monitor your applications' memory usage by performing Heap Breakdown analysis. The Heap Breakdown tab displays the memory metrics in a table and in a corresponding graph.


The following is an example of the Java Profiler Heap Breakdown Tab display.



To access	In the Java Diagnostics Profiler, select the Heap Breakdown tab.
Important information	If your application runs in JVM 1.5 or later you will see the Memory Analysis tab instead of the Heap Breakdown tab. See "Memory Analysis Tab Description" on page 1038. Heap Breakdown has been known to be unstable and to crash when used with IBM JVM 1.3.x.
Relevant tasks	By default, Heap Breakdown is disabled, so the Java Agent does not impose additional overhead on its host when you do not need the memory diagnostics metrics. You must enable heap breakdown (see "How to Enable Heap Breakdown" on page 996).

The following user interface elements are included:

UI Elements	Description
Heap Metrics Table	The Heap Metrics table contains the following columns: Class. The name of the class. Bytes. Actual amount of memory, in bytes, that has been allocated by objects of this class. Count. The number of object instances of this class that are allocated in the JVM. +/-Last. The count change since the most recent time a heap snapshot was taken. +/-First. The count change since the initial heap snapshot was taken

UI Elements	Description
Heap Breakdown Graph	When you click a class name in the Heap Breakdown table, the Heap Breakdown graph shows the count over time of objects belonging to that class. You can select more than one class to display on the graph by selecting subsequent rows with a CTRL-click .
	The data displayed in the Heap Breakdown tab is a snapshot of the system. You can choose how frequently the system takes an automatic sample by selecting one of the autosample options in the autosampling list box next to the graph. You can take a sample snapshot manually by clicking the sample button.

Memory Analysis Tab Description

The Profiler displays the Memory Analysis tab instead of the Heap Breakdown tab when your application is running JVM 1.5 or higher.

Data displayed in the Heap Breakdown view in the Memory Analysis Tab helps users of applications with Java 1.5 or higher to find memory leaks. The Memory Analysis Tab includes a wizard that guides you through the process of diagnosing a memory leak. See "Heap Walker Memory Analysis Execution Steps" on page 984 for details about the heap walker process you can use to diagnose memory leaks.

The following is an example of the Java Profiler Memory Analysis Tab display.

Server Requests | Web Services | Allocation/LifeCycle Analysis | **Memory Analysis** | Configuration

Heap Walker
Step 1 - Establishing a baseline

The Heap Breakdown summarizes the contents of the Heap Snapshots by class and helps you detect memory leaks.

If you observed a reproducible memory leak you are welcome to use this wizard to figure out why the leaking objects are not garbage collected:

First, determine an operation (such as "login, buy book, logout") that you would like to test for memory leaks.

To avoid false positives, ensure that the application has completely initialized by exercising this operation once or twice.

Once you have this good baseline to test against, press [Start Tracking New Objects](#).

Any object that the application allocates after this point will be tracked as a 'potential leak'.

Available Actions:

- Take another Heap Snapshot
- Start Tracking New Objects
- Run Garbage Collection

Last Action:
Heap snapshot taken in 2286ms

Class	Bytes	C	I	+/- Last	+/- First
char[]	115,...	634,...		?	?
String	14,9...	621,...		?	?
java.util.HashMap\$Entry	4,94...	206,...		?	?
Object[]	9,55...	148,...		?	?
java.util.HashMap\$Entry[]	9,12...	101,...		?	?
java.util.ArrayList	2,32...	96,9...		?	?
weblogic.ejb.container.cache.CacheKey	2,30...	96,2...		?	?
java.util.HashMap	3,82...	95,5...		?	?
Integer	1,36...	85,5...		?	?
java.util.HashSet	1,21...	75,8...		?	?
java.util.TreeMap\$Entry	2,23...	69,7...		?	?
com.mercury.opal.capture.inst.InstrumentationControl\$MethodSi...	2,14...	67,1...		?	?
weblogic.store.io.file.FileStoreIO\$DeleteRecord	1,41...	58,8...		?	?
com.mercury.opal.capture.inst.MethodData	2,72...	56,6...		?	?
String[]	1,81...	51,0...		?	?
weblogic.store.io.file.FileStoreIO\$Handle	1,21...	50,7...		?	?

Don't auto-sample

Last refresh: Tue Jun 23 12:51:32 PDT 2009

Probe ID: LB1_WLS10_MedRec_ovrhp04_11311A

To access	In the Java Diagnostics Profiler, select the Memory Analysis tab.
Important information	<p>You must have Java 1.5 or higher installed. If your application runs in a JVM version earlier than JVM 1.5 the Profiler displays the Heap Breakdown tab instead of the Memory Analysis tab. See "Heap Breakdown Tab Description" on page 1036.</p> <p>Heap Walker has JVM and memory requirements as described in this topic.</p> <p>Also See "Heap Walker Performance Characteristics" on page 989.</p>
Relevant tasks	<p>By default, the Memory Analysis tab is disabled, so the Java Agent does not impose additional overhead on its host when you do not need the memory diagnostics metrics. You must enable memory analysis (see "How to Enable Memory Analysis" on page 997).</p> <p>See "Heap Walker Memory Analysis Execution Steps" on page 984 for a description of how to use the heap walker wizard.</p>

The following user interface elements are included:

UI Elements	Description
Heap Metrics Table	<p>The Heap Metrics table contains the following columns:</p> <ul style="list-style-type: none">Class. The name of the class.Bytes. Actual amount of memory, in bytes, that has been allocated by objects of this class.Count. The number of object instances of this class that are allocated in the JVM.+/-Last. The count change since the most recent time a heap snapshot was taken.+/-First. The count change since the initial heap snapshot was taken

UI Elements	Description
Heap Breakdown Graph	When you click a class name in the Heap Breakdown table, the Heap Breakdown graph shows the count over time of objects belonging to that class. You can select more than one class to display on the graph by selecting subsequent rows with a CTRL-click .
Heap Walker	<p>The Memory Analysis Tab includes a wizard that guides you through the process of diagnosing a memory leak. See also "Heap Walker Memory Analysis Execution Steps" on page 984 for how to use the heap walker wizard. Heap Walker requires the following:</p> <p>JVM Requirements: Heap Walker uses the JVM Tool Interface (JVM TI), which was introduced in Java 5.0. As a result, the profiled application must run on a Java 5 VM that implements JVM TI, including the optional JVM TI capability <code>can_tag_objects</code>.</p> <p>Sun HotSpot JVM, version 5.0, for Linux and Windows on Intel x86, and HP-UX HotSpot JVM, version 5.0, for PA-RISC, are examples of compatible JVMs.</p> <p>Memory Requirements. Tagging the heap, and processing the object reference graph, requires large amounts of memory (total physical memory installed on the system). The amount of memory required depends on the size of the heap used by the application. You will see an error message if there isn't enough memory on the system based on the heap size.</p>

Configuration Tab Description

The Configuration tab in the Java Diagnostics Profiler provides a way for you to maintain the instrumentation points and some of the probe configuration without having to manually edit the capture points file or property files.

The following is an example of the Java Profiler Configuration Tab display.

HP Diagnostics - Diagnostics Profiler

Profiling Since: Thu May 13 09:23:39 AM PDT 2010

Server Requests | Web Services | Allocation/LifeCycle Analysis | Heap Breakdown | **Configuration**

Instrumentation

Currently Used Instrumentation: [View...](#)

Change Probe Instrumentation Plan

Shared Instrumentation: [Edit...](#) [326 points] Used by all probes running from this probe installation.

Instance Instrumentation: [Edit...](#) [0 points] Used by probes with Id: P81_WAS5_Plants_ovrntt152_...

Probe Settings

Sample Server Requests

Sample percent of server requests

Thread Stack Trace Sampling Maximum Stack Trace Depth

Sampling Interval ms Tardy Method Latency Threshold ms

Collect CPU Timestamps

Report Collection Leaks

Collection Leaks Flag Threshold minutes Collection Leaks Unflag Threshold minutes

[Apply Changes](#) [Cancel Changes](#)

Last refresh: Thu May 13 09:23:39 PDT 2010 Probe ID: P81_WAS5_Plants_ovrntt152_W2k

To access	In the Java Diagnostics Profiler, select the Configuration tab. You can use this page whether profiling has been started for the probe or not.
Important information	See the <i>HP Diagnostics Installation and Configuration Guide</i> chapter on "Custom Instrumentation for Java Applications" and the chapter on "Advanced Java Agent and Application Server Configuration" for more information on the properties configured in this page. In VMware, the CPU time metric is from the perspective of the guest operating system and is affected by the VMware virtual timer. See the VMware whitepaper on timekeeping at http://www.vmware.com/pdf/vmware_timekeeping.pdf and see "Time Synchronization for Probes Running in VMware in the <i>HP Diagnostics Installation and Configuration Guide</i> ."
Relevant tasks	"How to Access the Java Diagnostics Profiler" on page 991

The following user interface elements are included:

UI Elements	Description
Instrumentation Section	The Instrumentation part of the Configuration screen allows you to view and update the instrumentation for the application the probe is monitoring.
Probe Settings Section	The Probe Settings part of the Configuration screen allows you to configure server request sampling, thread stack trace sampling, and CPU timestamping the probe uses as it captures the performance metrics for your application. Changes you make take affect immediately after you click Apply Changes . There is no need to restart the application.

UI Elements	Description
<p>Sample Server Requests</p>	<p>Server request sampling allows Diagnostics to gather detailed information for a subset of all the server requests executed in your application. This reduces the overhead of the probe because no information is captured for server requests that are not sampled. As long as the sampling rate is not too low, the average results from all sampled server requests should provide results that are reasonably accurate and meaningful for understanding the performance of your applications.</p> <p>Sample Server Requests. Check this box to enable server request sampling.</p> <p>Sample. Select the sampling percentage.</p> <p>You can also set the following properties in the <code>dynamic.properties</code> file: <code>enable.server.request.sampling</code>, <code>server.requests.sampling.rate</code>, <code>server.request.sampling.method</code>, <code>sample.colored.requests</code>.</p>

UI Elements	Description
<p>Thread Stack Trace Sampling</p>	<p>When asynchronous thread sampling is enabled, you can see, in the Call Profile view, which methods were executed during long running fragments even if no instrumented methods were hit during this time.</p> <p>You can enable and configure the following properties.</p> <p>Thread Stack Trace Sampling. Enables or disables asynchronous thread stack trace sampling; possible values are false, auto (the default), and true.</p> <p>When set to auto, stack trace sampling is enabled IF the probe is running on selected (certified) platforms and JVMs. For other JVMs, the setting must be set to Enable explicitly. Use caution because the JVM could generate errors or abort. See Diagnostics Readme for limitations.</p> <p>Sampling Interval. The time that must elapse before the next consecutive sampling attempt is made. Small values cause frequent sampling and provide rich data but at the cost of increased overhead.</p> <p>The overhead caused by frequent sampling affects primarily the latency of server requests. The overall CPU usage by the probe can go up as well, but this effect is not as profound as the latency increase. For systems with many CPUs, the process CPU consumption can actually go down (not a good thing).</p> <p>Maximum Stack Trace Depth. The limit for the depth of stack traces obtained from the JVM. You will most likely not need to adjust this value.</p> <p>Tardy Method Latency Threshold. The minimum time an instrumented method must run without hitting any instrumentation points before stack trace sampling is attempted for this method. The purpose of this property is to control the overhead of sampling by limiting the stack trace collection to only the most interesting cases.</p> <p>These properties can also be set in the dynamic.properties file. And additional configuration can be done in dispatcher.properties for enable.stack.trace.aggregation, aggregated.stack.trace.validity.threshold.</p>

UI Elements	Description
Collect CPU Timestamps	<p>The CPU timestamps calculate the amount of exclusive CPU time that a method uses. You can view this information on the Hotspots tab in the Java Diagnostics Profiler.</p> <p>By default, collection of CPU time metrics is enabled for server requests. Select an option:</p> <p>None. No CPU timestamps.</p> <p>For Server Requests Only. CPU timestamps are only collected for server requests. This is the default setting.</p> <p>For Server Requests and Portlet Methods. CPU timestamps are collected for all server requests and the lifecycle methods instrumented for portal components (layertype=portlet).</p> <p>For Server Requests and ALL Methods. CPU timestamps are collected for all server requests and ALL methods.</p> <p>You can also set these properties in the capture.properties file (use.cpu.timestamps) and dynamic.properties file (cpu.timestamp.collection.method).</p>

UI Elements	Description
Report Collection Leaks	<p>Data collection and reporting for collection leak pinpointing is enabled by default for the probe. The [Collection Leak Pinpointing] keyword = clp point in the probe's etc/auto_detect.points file is set to true by default.</p> <p>You must initially run the 9.0 JRE Instrumenter if you want to use the collection leaks pinpointing (CLP) feature in the Java Agent.</p> <p>Report Collection Leaks. You can enable and disable reporting in the UI by checking or unchecking this box.</p> <p>Collection Leaks Flag Threshold. The threshold of time duration in which the collection has size growth. If a collection's size growth period exceeds this threshold, it will be flagged as a memory leak by the probe.</p> <p>Collection Leaks Unflag Threshold. For an already flagged leaking collection, if its size stops growing continually for this threshold time period, that probe will unflag it as a leak.</p> <p>These same values can also be set in the dynamic.properties file for the probe: clp.diagnostics.reporting, clp.diagnostics.growth.time and clp.diagnostics.nongrowth.time.</p>

38

Diagnostics Profiler for .NET

This chapter describes how to use the Diagnostics Profiler for .NET.

This chapter includes:

Concepts

- ▶ About the .NET Diagnostics Profiler on page 1050
- ▶ How the .NET Agent Provides Data for the .NET Profiler on page 1051
- ▶ .NET Diagnostics Profiler UI Navigation and Display Controls on page 1052
- ▶ .NET Diagnostics Profiler Inactivity Timeout on page 1054

Tasks

- ▶ How to Access the .NET Diagnostics Profiler on page 1055
- ▶ How to Enable and Disable the .NET Diagnostics Profiler on page 1056

Reference

- ▶ .NET Diagnostics Profiler User Interface on page 1058

Concepts

About the .NET Diagnostics Profiler

The Diagnostics Profiler for .NET is installed with the .NET Agent. The Profiler runs in a separate UI and provides near real-time data, enabling you to pinpoint application performance bottlenecks.

Note: The Diagnostics Profiler for .NET operates in an unlicensed mode with load restrictions until the probe is able to connect to a Diagnostics Server that has been properly licensed. In unlicensed mode, the .NET Profiler is limited to capturing data from 5 concurrent threads.

If you installed the unlicensed trial software agent from the HP Software Web site and you want to use it with a Diagnostics Server, contact HP Software Support to purchase HP Diagnostics.

If you are using Diagnostics with LoadRunner or Performance Center you will be prompted to enter the Diagnostics User Name and Password when selecting the .NET Profiler from the Diagnostics UI.

You can use the different tabs in the .NET Profiler to analyze method latency for the selected application. And you can analyze memory problems for the selected application using the memory diagnostics metrics displayed in the .NET Profiler.

Some of the information presented in the .NET Profiler is also available in the Diagnostics enterprise UI.

How the .NET Agent Provides Data for the .NET Profiler

This section describes the way in which the .NET Agent monitor your application and how this data is displayed in the .NET Diagnostics Profiler.

Monitoring Method Latency and Call Stacks

The Diagnostics Agent for .NET (.NET Probe) runs probes to monitor your application and keep track of the metrics for all of the instrumented methods that your application calls. As probes are monitoring, they capture the call stack for the three slowest instances and the single fastest instance of each server request.

When a new server request instance is encountered that is slower than one of the currently captured instances for the server request, it replaces one of the previously captured instances. In the same manner the captured call stack for the fastest instance is replaced when an instance that is even faster is encountered.

The .NET Diagnostics Profiler displays metrics for all of the instrumented methods. The .NET Profiler ignores all configured trim settings, for example, latency trimming, depth trimming or throttling. For details about trim configuration refer to the section about advanced .NET Probe Configuration in the *HP Diagnostics Installation and Configuration Guide*. You can drill down to the instances of the methods that were included in one of the four server request call stacks that were captured when you accessed the .NET Diagnostics Profiler user interface.

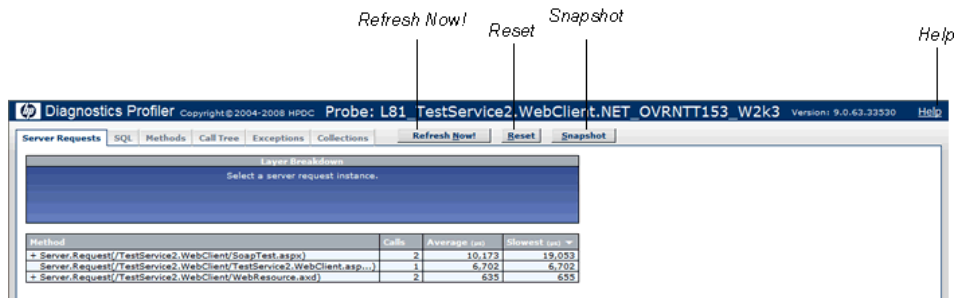
While you are analyzing the information displayed on the various tabs of the .NET Diagnostics Profiler, you are working with the methods and call stacks captured from the time that the .NET Profiler was started/reset to the time that the user interface was started/refreshed. In the meantime the probe continues to monitor your application, capture method metrics, and capture call stacks. These changes are not sent automatically to the user interface, you must request them via the **Refresh Now** button. This is so the underlying data will not change unexpectedly while you are investigating something of interest.

Monitoring Application Memory

The .NET Diagnostics Profiler allows you to monitor your application's memory usage using Light Weight Memory Diagnostics. Light Weight Memory Diagnostics allows you to monitor the collections that your application has created, and to identify the largest collections and the fastest growing collections. For more information about Light Weight Memory Diagnostics, see "Collections Tab Description" on page 1075.

.NET Diagnostics Profiler UI Navigation and Display Controls

This section describes the following features and controls that are common to all of the .NET Profiler tabs: **Refresh now**, **Reset**, **Snapshot** and **Help**:



Refresh Metrics

Click **Refresh Now** to refresh the information displayed on the tabs with the latest metrics and call stacks.

After you refresh the metrics, the .NET Diagnostics Profiler continues to monitor and collect metrics using the same baseline for the calculations of instance counts, average latency, and slowest latency. It also continues to use the captured call stacks as a basis of comparison for finding new call stacks to capture.

Reset Metrics

You can force the .NET Diagnostics Profiler to use new baselines for the calculation of instance counts, average latency, and slowest latency, and to force-drop all captured call stacks, by clicking **Reset**.

After you reset the metrics, the .NET Diagnostics Profiler begins collecting data with new baselines and starts processing the instance trees as though the profiler had just been started.

Note: You may want to click **Reset** once your system has warmed up so that you can do your performance analysis using metrics that are more representative of the processing that takes place when your application is running in steady state.

Take a Snapshot

You can capture a snapshot of the data from your profiler session into an .xml formatted file, by clicking the **Snapshot** button.

The resulting snapshot can be used, for example, as a report that is distributed to your colleagues or as a point of reference when you are about to make changes to your applications. The snapshot includes the profiler tabs so that you can review and analyze the data in the snapshot in the same way that you would view it in the Profiler.

The Profiler displays a dialog box that indicates the path to where the .xml file is stored. When you open the snapshot, the saved profiler data is displayed in your browser.

Access Help

When you click **Help**, on the top right hand corner of the screen, you access the on-line help manual for the Diagnostics Profiler for .NET.

.NET Diagnostics Profiler Inactivity Timeout

By default, the .NET Diagnostics Profiler is not started until you display the Profiler UI. When you close the Profiler UI, the profiler continues to run for a period of time specified by the **inactivitytimeout** attribute in `<probe_install_dir>/etc/probe_config.xml`. If you reopen the Profiler UI before the profiler times out, the profiler displays the data for the time period since the profiler was started. If you reopen the Profiler UI after the timeout has occurred, the profiler is restarted and only the data for the new profiler session is displayed. As long as the Profiler UI is open, the profiler session remains active. The count down for the inactivity timeout begins when you close the Profiler UI.

Tasks

How to Access the .NET Diagnostics Profiler

Once you have installed the .NET Agent, configured a probe to collect performance metrics and started the application that is being monitored, you can access the .NET Diagnostics Profiler from your browser and view diagnostics data. You can also access the .NET Diagnostics Profiler by drilling down from the views of the HP Diagnostics user interface.

Remote access to the .NET Profiler can be disabled with the profiler element in the probe_config.xml file.

To open the .NET Diagnostics Profiler directly (standalone):

- 1 In your browser, go to the .NET Diagnostics Profiler URL: http://<probe_host>:<probeport>/profiler.

The probes are assigned to the first available port beginning at **35000**.

- 2 Type your username and password.

Depending on your authentication settings, you may be prompted to enter a username and password. The default username is **admin**. The default password is **admin**.

For more information about authentication, usernames and passwords when you have the full Diagnostics product, refer to the *HP Diagnostics Installation and Configuration Guide* section on Authentication and Authorization.

To drill down to the Diagnostics .NET Profiler from the main Diagnostics UI:

- 1 From any view in HP Diagnostics that shows probe entities, right-click the probe in the table and select **View Profiler for <probe name>** from the menu.

If you are using Diagnostics with LoadRunner or Performance Center you will be prompted to enter the Diagnostics User Name and Password when selecting the .NET Profiler from the Diagnostics UI.

- 2 If the Profiler fails to open, ensure that you have set a default browser within your operating system.

How to Enable and Disable the .NET Diagnostics Profiler

This task describes how to disable and re-enable the .NET Profiler to start.

When the .NET Probe is installed and probes configured to work with a Diagnostics Server, the probe data collection starts automatically when a Web page in the monitored application is accessed.

By default the .NET Diagnostics Profiler isn't started until you access the Profiler UI. You may configure the agent so that the .NET Profiler is started at the same time that the probe data collection is started or so the .NET Profiler cannot be started.

To configure the probe to automatically start the profiler:

You may want to start the .NET Profiler at the same time that the probe is started if you are trying to understand the performance of your application when it is first invoked.

Set the **pro** attribute in `<probe_install_dir>/etc/probe_config.xml` to true.

```
<modes enterprise="true" pro="true"/>
```

To configure the probe to prevent the Profiler from starting:

You may want to prevent someone from starting the .NET Profiler for a probe that is monitoring an application where you do not want to incur the additional overhead from the .NET Profiler.

Set the **pro** attribute in `<probe_install_dir>/etc/probe_config.xml` to false

```
<modes enterprise="true" pro="false"/>
```

To configure the probe to start the Profiler when you access the UI:

By default, the probe starts the Profiler when you bring up the Profiler UI. If you have altered the setting for the probe, you may want to reset the behavior of the probe to the default behavior.

Set the **pro** attribute in `<probe_install_dir>/etc/probe_config.xml` to auto:

```
<modes enterprise="true" pro="auto"/>
```

Note: If you do not include the **pro** attribute, the probe defaults to the behavior when **pro** is set to auto.

Reference

.NET Diagnostics Profiler User Interface

This section includes:

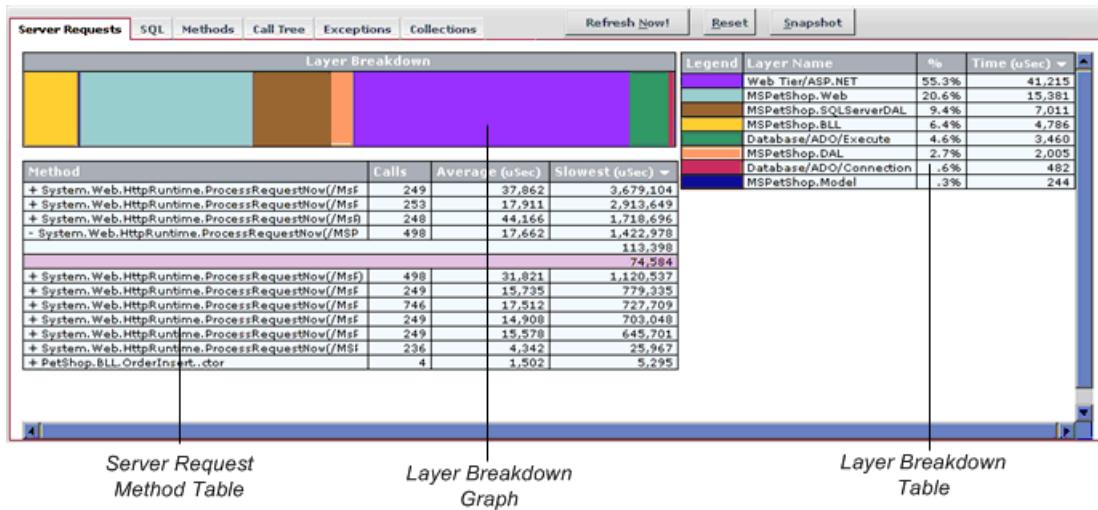
- ▶ Server Requests Tab Description on page 1058
- ▶ SQL Tab Description on page 1062
- ▶ Methods Tab Description on page 1065
- ▶ Call Tree Tab Description on page 1067
- ▶ Exceptions Tab Description on page 1071
- ▶ Collections Tab Description on page 1075

Server Requests Tab Description

The .NET Diagnostics Profiler keeps track of all of the method calls made by your application. The Server Requests tab displays information about the server request methods. The server request methods are listed in a table that shows the number of times that each method was executed, along with the average latency and the slowest execution time for all of the calls to the method. You can expand each server request listed in the table, to reveal the latency for the three slowest instances of the server request along with the single fastest instance.

Note: The .NET Diagnostics Profiler captures call trees for the three slowest instances and the single fastest instance of each server request. The .NET Diagnostics Profiler lets you drill into the captured call trees from the Server Requests tab.

The following is an example of the .NET Profiler Server Requests Tab display.



To access	In the .NET Diagnostics Profiler, select the Server Requests tab.
Relevant tasks	"How to Access the .NET Diagnostics Profiler" on page 1055

The following user interface elements are included:

UI Elements	Description
Server Request Method Table	<p>The Method table lists the server requests that have been called. You can sort the table by clicking the column headers.</p> <p>The following columns are included in the table:</p> <p>Method. The server request methods that were called.</p> <p>If a server request method was called more than once, the method name is preceded by a plus sign (+) or a minus sign (-) to indicate that the instance specific latency information is available for the server request.</p> <p>Calls. The number of times that the server request method was invoked.</p> <p>Average. The average latency for all of the calls to the server request method. The average latency is shown in microseconds.</p> <p>Slowest. The response time of the instance with the longest latency. The slowest response time is shown in microseconds.</p> <p>If a server request method was called more than once, the method name is preceded by a plus sign (+) or a minus sign (-). When you click the plus sign, the entry is expanded to reveal the three slowest instances of the method along with the single fastest method. Click the minus sign to the collapse instances shown.</p> <p>If a server request method was called only once, the entry itself represents the single instance of the method call. The value in the Slowest column is the instance's latency.</p> <p>You can view the call tree for a server request instance by clicking on any row that contains a server request instance (a row that does not have a plus sign (+) or a minus (-) sign before the method name or that only contains a latency value is a server request instance).</p> <p>The Profiler switches to the Call Tree tab and displays the call tree for the selected server request instance. The method call for the selected server request is highlighted in blue in the call tree.</p>

UI Elements	Description
Layer Breakdown Graph	<p>The Layer Breakdown graph shows the amount of processing time that was spent in each layer while executing a selected instance of a method call. It is a graphical representation of the information shown in the Layer Breakdown table.</p> <p>You can view the Layer Breakdown for a server request instance by hovering the mouse pointer on any row in the Method table that contains a server request instance (a row that does not have a plus sign (+) or a minus (-) sign before the method name, or that only has a latency value, is a server request instance).</p> <p>The Profiler shows the layer breakdown for the indicated instance in both the Layer Breakdown Graph and Layer Breakdown Table.</p> <p>The graph is divided so that each layer is depicted as an area on the graph that is proportional to the percentage of processing that was performed in the layer. Each layer is displayed in a different color, as shown in the Legend column in the Layer Breakdown table.</p>
Layer Breakdown Legend	<p>The Legend shows the amount of processing time that was spent in each layer while executing a selected instance of a method call. The table can be sorted by clicking the column headers.</p> <p>The following columns are included in the table:</p> <p>Legend. The color that is used in the Layer Breakdown graph to depict the processing that took place in the layer.</p> <p>Layer Name. The name of the layer where the processing for the server request took place.</p> <p>%. The percentage of processing time that was spent in each layer, for a selected server request.</p> <p>Time. The latency measured for the processing that took place in the layer, for a selected server request. The time is shown in microseconds.</p>

SQL Tab Description

The .NET Diagnostics Profiler keeps track of all of the method calls that your application makes. The SQL tab displays the SQL methods only. The SQL methods are listed in the Method table which shows the number of times that each method was executed, along with the average latency and the slowest execution time for all of the calls to the method. The Method table also shows the actual SQL statement when it was included in the SQL method call.

Each SQL method listed in the table can be expanded to reveal the latency for each instance of the method that was included in a captured call tree.

The following is an example of the .NET Profiler SQL Tab display.

Method	Calls	Average (µs)	Slowest (µs)	SQL
+ PetShop.SQLServerDAL.SQLE	998	4,403	1,401,536	
+ System.Data.SqlClient.SqlCon	249	7,805	1,400,624	SELECT Account.Emat.L...
- System.Data.SqlClient.SqlCorr	498	1,996	85,143	SELECT Item.ItemIdte...
				1,327
				1,203
				1,149
				933
System.Data.SqlClient.SqlCon	1	41,496	41,496	SELECT ProductId, Nategory...
+ PetShop.SQLServerDAL.SQLE	249	2,340	28,950	
+ System.Data.SqlClient.SqlCon	249	1,276	21,984	SELECT Qty FROM ItemId
+ System.Data.SqlClient.SqlCon	249	1,461	16,101	SELECT Account.Firstoun...
System.Data.SqlClient.SqlCon	1	1,558	1,558	SELECT ItemId, Attr: IN...

To access	In the .NET Diagnostics Profiler, select the SQL tab.
Important information	The .NET Diagnostics Profiler captures call trees for the three slowest instances and the single fastest instance of each server request. You can drill down to the captured call trees from the SQL tab.
Relevant tasks	"How to Access the .NET Diagnostics Profiler" on page 1055
See also	For more information on the Call Tree tab, see "Call Tree Tab Description" on page 1067.

The following user interface elements are included:

UI Elements	Description
Table	<p>This table lists the SQL methods that have been called, and displays latency information for instances of the SQL method calls that were included in the captured call trees. The table can be sorted by clicking the column headers.</p> <p>The following columns are included in the table:</p> <p>Method. The SQL methods that were called. If an SQL method has two or more instances in the captured call trees, the method name is preceded by a plus sign (+) or a minus sign (-) to indicate additional instance specific latency information can be viewed for the SQL call.</p> <p>Calls. The number of times that the SQL method was invoked. This count includes all instances, whether or not they are included in the captured call trees.</p> <p>Average. The average latency for all of the calls to the SQL method. The average latency is shown in microseconds.</p> <p>Slowest. The response time for the instance with the longest latency. The slowest response time is shown in microseconds.</p> <p>SQL. The first part of the SQL statement that was executed by the SQL method call.</p> <p>You can display a tooltip containing the entire SQL statement by holding the mouse pointer over a row in the SQL column.</p> <p>The latencies for instances of SQL methods can be displayed if they are included in one of the captured call trees.</p> <p>If two or more instances of an SQL method are included in the captured call trees, that method's name is preceded by a plus sign (+) or a minus sign (-) in the Method table. The entry can be expanded to reveal the latency for each of the captured instances for the selected method. Click the minus sign to collapse the visible instances.</p>

UI Elements	Description
<p>Table (continued)</p>	<p>if only one instance of an SQL method was included in the captured call trees, the method name in the SQL Method table is not preceded by a plus sign or minus sign and the table entry itself represents the single instance of the method call, and the value in the Slowest column is the instance's latency.</p> <p>If no instances of a SQL method were included in the captured call trees, the method is not preceded by a plus sign or minus sign, and when you click the method, you get a message indicating that although this method was called there is no data captured for it.</p> <p>You can view the call tree for an SQL method instance listed in the SQL Method table by clicking on any row that contains an instance of an SQL method call. (A row that does not have a plus sign (+) or a minus (-) sign before the method name, or that only contains a latency value, is an SQL instance.)</p> <p>When you select a row with an SQL method instance, the Call Tree tab opens, and displays the call tree for the selected SQL method instance. The method call for the selected SQL method is highlighted in blue in the call tree.</p> <p>For information on the Call Tree tab, see "Call Tree Tab Description" on page 1067.</p>

Methods Tab Description

The .NET Diagnostics Profiler keeps track of all of the method calls that your application makes. The Methods tab is used to list all of the methods. The methods are listed in the Method table, which shows the number of times each method was executed, along with the average latency and the slowest execution time for all of the calls to the method. The methods listed in the Methods tab include the server requests methods listed in the Server Requests tab, the SQL methods listed in the SQL tab, and the methods that generated exceptions shown in the Exceptions tab.

Each method listed in the table can be expanded to reveal the latency for each instance of the method that was included in one of the captured call trees. The .NET Diagnostics Profiler captures call trees for the three slowest instances and the single fastest instance of each server request. The .NET Diagnostics Profiler lets you drill down to the captured call trees from the Methods tab.

The following is an example of the .NET Profiler Methods Tab display.

Method	Calls	Average (µs)	Slowest (µs)
+ System.Web.HttpRuntime.ProcessRequestNow	3475	21,627	3,679,104
+ PetShop.Web.Global.Application_Error	248	14,396	1,701,324
+ PetShop.Web.OrderProcess.OnLoad	248	23,314	1,556,395
+ PetShop.Web.ProcessFlow.CartController.PurchaseCart	248	23,061	1,550,664
+ PetShop.Web.SignIn.SubmitClicked	249	11,543	1,405,137
+ PetShop.Web.ProcessFlow.AccountController.ProcessLogin	249	10,951	1,404,444
+ PetShop.BLL.Account.SignIn	249	10,094	1,403,582
+ PetShop.SQLServerDAL.Account.SignIn	249	9,545	1,403,022
+ PetShop.SQLServerDAL.SQLHelper.ExecuteReader	998	4,403	1,401,536
+ System.Data.SqlClient.SqlCommand.ExecuteReader	998	3,351	1,400,624
+ PetShop.BLL.Cart.GetOrderLineItems	248	3,466	720,331

To access	In the .NET Diagnostics Profiler, select the Methods tab.
Relevant tasks	"How to Access the .NET Diagnostics Profiler" on page 1055

The following user interface elements are included:

UI Elements	Description
<p>Table</p>	<p>This table lists the methods that have been called, and displays latency information for instances of the method calls that are included in the captured call trees. This table can be sorted by clicking the column headers.</p> <p>The following columns are included in the table:</p> <p>Method. The name of the methods that were called. If a method has two or more instances included in the captured call trees, the method name is preceded by a plus sign (+) to indicate additional instance specific latency information can be viewed for the method call.</p> <p>Calls. The number of times that the method was invoked. This count includes all instances, whether or not they are included in the captured call trees.</p> <p>Average. The average latency for all of the calls to the method. The average latency is shown in microseconds.</p> <p>Slowest. The response time for the instance with the longest latency. The slowest response time is shown in microseconds.</p> <p>You can view the latency for instances of methods if they are included in one of the captured call trees.</p> <p>If two or more instances of a method are included in the captured call trees, the method name in the Method table is preceded by a plus sign (+) or a minus sign (-). The plus sign indicates that you can expand the entry to reveal the latency for each of the captured instances for the selected method. Click the minus sign to collapse the visible instances.</p> <p>If no instances of a method were included in the captured call trees, the method is not preceded by a plus sign or minus sign, and when you click the method, you get a message indicating that although this method was called there is no data captured for it.</p>

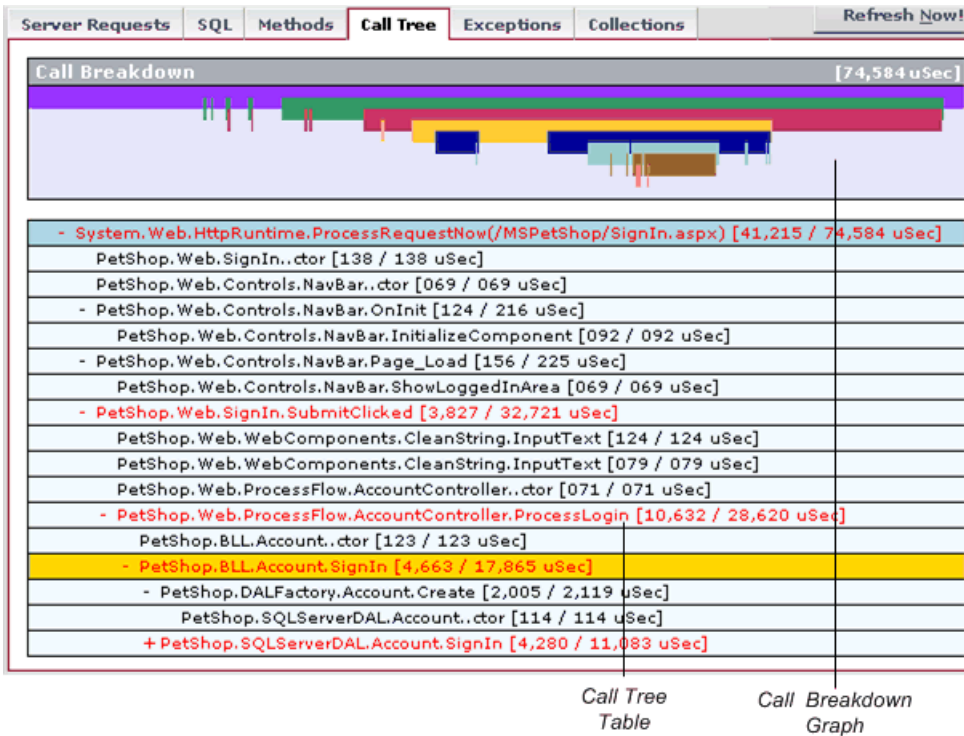
UI Elements	Description
Table (continued)	<p>You can view the call tree for a method instance listed in the Method table by clicking on any row that contains an instance of a method call. (A row that does not have a plus sign (+) or a minus (-) sign before the method name, or that only contains a latency value, is a method instance.)</p> <p>When you click a row with a method instance, the Call Tree tab opens and displays the call tree for the selected method instance. The method call for the selected method is highlighted in blue in the call tree.</p> <p>For information on the Call Tree tab, see "Call Tree Tab Description" on page 1067.</p>

Call Tree Tab Description

The .NET Diagnostics Profiler captures call trees for the three slowest instances and the single fastest instance of each server request. The captured server request call trees are displayed on the Call Tree tab, in the Call Breakdown graph and in the Call Tree table.

As you analyze the methods presented on the Server Requests, SQL, Exceptions, and Methods tabs, you navigate to the Call Tree tab to understand the context of the processing associated with particular instances of the method's execution. The call tree allows you to see the calling and the callee methods for the method of interest as well as the contribution of those methods to the measured latency.

The following is an example of the .NET Profiler Call Tree Tab display.



To access	In the .NET Diagnostics Profiler, select the Call Tree tab. You can also access a Call Tree by clicking one of the method instances listed on the Server Requests, SQL, Exceptions, and Methods tabs.
Relevant tasks	"How to Access the .NET Diagnostics Profiler" on page 1055

The following user interface elements are included:

UI Elements	Description
<p>The Call Breakdown Graph</p>	<p>The Call Breakdown graph shows the processing time that was spent at each level of the call tree hierarchy.</p> <p>Each level in the graph represents the processing at the corresponding level in the call stack. The length of the bar is proportional to the length of time spent in performing the methods at that level of the call stack. The positions where a bar starts and stops indicates the relative time, in relationship to the other levels, that the processing for the level began and ended. A gap in a bar, where the bar ends and then resumes again, indicates that the processing returned to a higher level in the hierarchy before once again proceeding at the lower level.</p> <p>There are two ways that you may identify the method associated with a particular location on the Call Breakdown graph as you mouse over the bars in the graph.</p> <p>As you slide the pointer along a bar in the graph, a tooltip is displayed with the name of the method associated with each segment of the graph bar.</p> <p>As you slide the pointer along a bar in the graph, the Call Tree table scrolls so that the method associated with the selected location in the graph is displayed in the table. The row that contains the selected method is highlighted in gold.</p>

UI Elements	Description
Call Tree Table	<p>The Call Tree table lists method calls that are part of a captured server request call tree in a hierarchical structure.</p> <p>Each method in the call tree is depicted on a separate line containing two parts: the method name and the latency.</p> <p>The latency for each method is shown in brackets following the method name. There are two numbers in the brackets separated by a slash: the exclusive latency and the total latency.</p> <p>Exclusive Latency is the amount of latency that is attributable to just the processing in the selected method.</p> <p>Total Latency is the amount of latency that is attributable to the selected method and all of its callee methods.</p> <p>In the following example the exclusive latency is 156: - PetShop.Web.Controls.NavBar.PageLoad [156/225 uSec]</p> <p>To see a captured call tree on the Call Tree tab you must select a method instance from one of the other .NET Diagnostics Profiler tabs. The Call Tree tab opens with the call tree that contains the selected instance visible and the selected method instance highlighted in blue.</p> <p>The method of interest will remain highlighted until a different method is selected on one of the other tabs.</p> <p>You may identify the method associated with a particular location on the Call Breakdown graph by mousing over the bars in the graph. As you slide the pointer along a bar in the graph, the Call Tree table scrolls so that the method associated with the selected location in the graph is displayed in the table. The row that contains the selected method is highlighted in gold.</p> <p>The path through the call tree that has the longest latency is called the critical path. Methods in the Call Tree table that are on the critical path are written using a red font.</p>

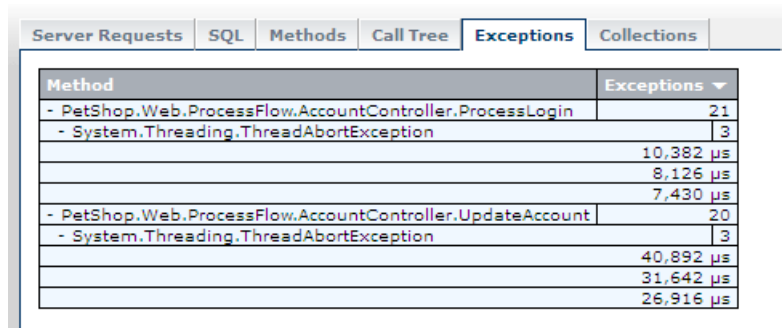
Exceptions Tab Description

The .NET Diagnostics Profiler keeps track of all of the method calls that your application makes. The Exceptions tab is used to list only the methods that generated exceptions. The calling methods that generated exceptions are listed in a table that shows the number of times that each method threw an exception. This information allows you to quickly determine if your application is throwing exceptions, and exactly what those exceptions are.

If the exception was included in one of the captured call trees, the exception class will also be listed in the table along with the latency for each instance of an exception.

Note: The .NET Diagnostics Profiler captures call trees for the three slowest instances and the single fastest instance of each server request. You can drill down to the captured call trees from the Exceptions tab.

The following is an example of the .NET Profiler Exceptions Tab display.



Method	Exceptions
- PetShop.Web.ProcessFlow.AccountController.ProcessLogin	21
- System.Threading.ThreadAbortException	3
	10,382 µs
	8,126 µs
	7,430 µs
- PetShop.Web.ProcessFlow.AccountController.UpdateAccount	20
- System.Threading.ThreadAbortException	3
	40,892 µs
	31,642 µs
	26,916 µs

To access	In the .NET Diagnostics Profiler, select the Exceptions tab.
Important information	Exceptions are only captured by the probe if the exception causes the termination of a method. If the instrumented method handles the exception, no exception information is gathered by the probe.
Relevant tasks	"How to Access the .NET Diagnostics Profiler" on page 1055

The following user interface elements are included:

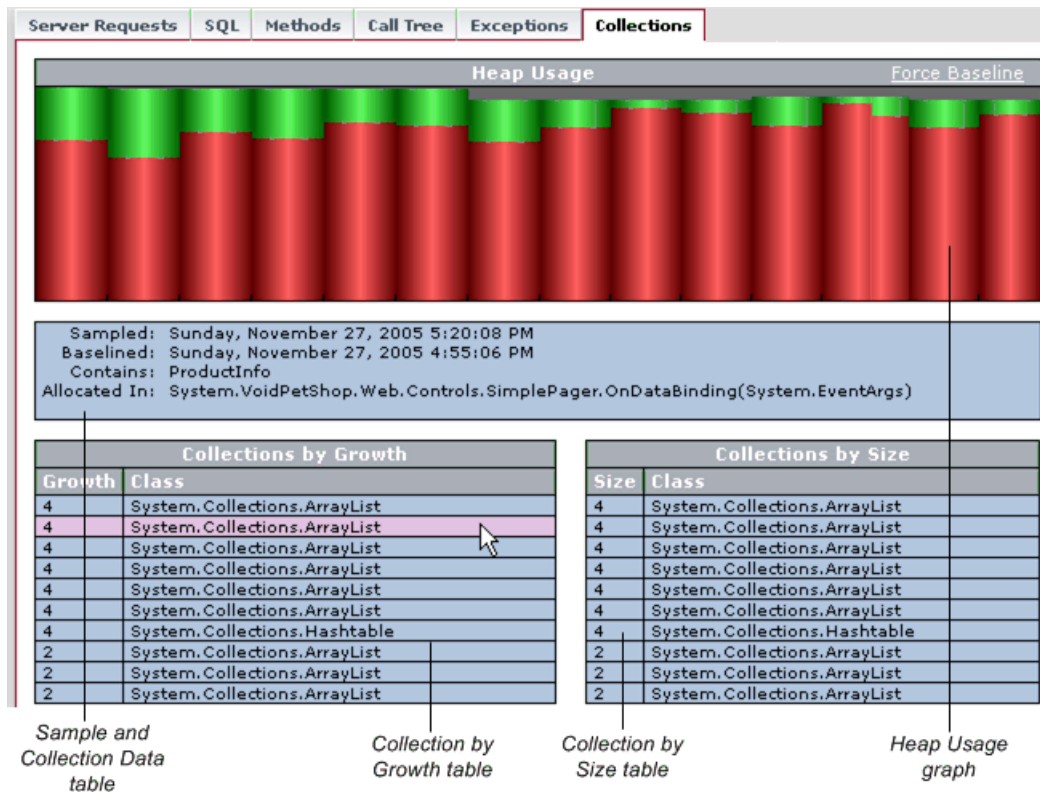
UI Elements	Description
Table	<p>This table lists the methods calls that generated exceptions and allows you to view latency information for instances of the exceptions that were included in the captured call trees. The rows in this table can be sorted by clicking the column headers.</p> <p>The table includes the following columns:</p> <p>Method. The name of the methods generated exceptions. If a method generated two or more exceptions and they were included in the captured call trees, the method name is preceded by a plus sign (+) or a minus sign (-) to indicate that additional instance-specific latency information can be viewed for the exception.</p> <p>Exceptions. The number of times that the method generated an exception. This count includes all instances of all classes of exceptions, whether or not they are included in the captured call trees.</p> <p>The latency for instances of exceptions are available to be displayed if they are included in one of the captured call trees.</p> <p>If an instance of an exception for a particular method call was included in one of the captured call trees, the method name in the Exceptions table is preceded by a plus sign (+) or a minus sign (-). The plus sign indicates that when you click the row in the table, the entry expands to reveal additional rows with the exception class for each of the captured instances of the exception. The minus sign indicates that when you click the row in the table, the entry contracts so that the exception class row is hidden.</p>

UI Elements	Description
<p>Table (continued)</p>	<p>If two or more instances of an exception class were included in the captured call trees, the exception class name in the Exceptions table is preceded by a plus sign (+) or a minus sign (-). The plus sign indicates that when you click the row in the table, the entry expands to reveal the latency for each of the captured instances for the selected exception class. The minus sign indicates that when you click the row in the table, the entry contracts so that the latency for the captured exception class is hidden.</p> <p>If only one instance of an exception class was included in the captured call trees, the exception class in the Exceptions table is not preceded a plus sign or minus sign. In this case, the table entry itself represents the single instance of the exception class and the value in the latency for the exception can be determined from the Call Trees tab.</p> <p>You can view the call tree for an exception listed in the Exceptions table by clicking on any row that contains an instance of an exceptions class. (A row that does not have a plus sign (+) or a minus (-) sign before the exception class or that only contains a latency value is an exception class instance.)</p> <p>When you click a row with an exception class instance, the profiler switches to the Call Tree tab and displays the call tree for the selected exception instance. The method call that generated the exception for the selected exception class is highlighted in blue in the call tree.</p> <p>For information on the Call Tree tab, see "Call Tree Tab Description" on page 1067.</p>

Collections Tab Description

The .NET Diagnostics Profiler can monitor your applications' memory usage using Light Weight Memory Diagnostics (LWMD). LWMD monitors the memory used by your applications by tracking the collections. The metrics from LWMD are displayed on the Collections tab. The memory metrics are shown in a graph of heap usage, and in tables that list the collections that are growing the fastest and that have become the largest. The Collections tab displays these problems, enabling identification of memory issues.

The following is an example of the .NET Profiler Collections Tab display.



To access	In the .NET Diagnostics Profiler, select the Collections tab.
Relevant tasks	"How to Access the .NET Diagnostics Profiler" on page 1055

The following user interface elements are included:

UI Elements	Description
<p>Heap Usage Graph</p>	<p>The Heap Usage graph shows the memory that was committed and used at periodic sample intervals. (The default sample interval is 1 minute.) For each sample interval, a bar is displayed on the graph.</p> <ul style="list-style-type: none"> ▶ The height of the bar indicates the total amount of heap that was committed when the sample was taken. ▶ The red portion of the bar indicates the amount of the heap that was committed and used when the sample was taken. ▶ The green portion of the bar indicates the amount of the heap that was committed, but not used, when the sample was taken. <p>Hold the mouse pointer over a sample's bar on the graph to display a tooltip showing the size of the heap that was used, followed by the size of the heap that was committed for the selected sample.</p> <p>By default, the LWMD process establishes a new baseline for measuring the growth of collections every hour. You can force a new baseline to be set by clicking the Force Baseline link at the upper-right corner of the Heap Usage graph.</p> <p>When the .NET Diagnostics Profiler establishes a new baseline, a green line is inserted between the last sample of the previous baseline and the first sample of the next baseline to mark the point where the baseline was set.</p> <p>The calculation for the growth of collections that is used to determine which collections are included in the Collections by Growth table, is based on the number of collections added since the last baseline.</p>

UI Elements	Description
Samples and Collections Details Pane	<p>Displays additional information about the sample selected in the Heap Usage graph, and about the collection selected from the collection tables.</p> <p>It contains the following information:</p> <p>Sampled. The date and time when the selected Heap Usage sample was taken.</p> <p>Baselined. The date and time of the last baseline prior to the sample being taken.</p> <p>Contains. The type of object contained in the selected collection. This information is displayed when you mouse over the Collections by Growth or Collections by Size tables.</p> <p>Allocated In. The method that allocated the selected collection. This information is displayed when you mouse over the Collections by Growth or Collections by Size tables.</p>

UI Elements	Description
Collections by Growth Table	<p>The Collections by Growth table lists the top ten collections in relation to the growth in the number of objects contained in the collection since the last baseline. The top-ten list of collections changes from sample to sample as the growth rates for each collection fluctuate. When a new baseline is established, the growth rate is calculated in relation to the new baseline, so the list of collections can change significantly.</p> <p>The table contains the following information:</p> <p>Growth. The number of objects that were added to the collection since the last baseline.</p> <p>Class. The class name for the collection.</p> <p>To see details for the collection, hold the mouse pointer over the row in the table for the collection. The row is highlighted in pink and the details are displayed in the Samples and Collections Details pane.</p>
Collections by Size Table	<p>The Collections by Size table lists the top ten collections relative to the size of the collection for the selected Heap Usage sample. The size of a collection is based upon the total number of objects in the collection.</p> <p>The table contains the following information:</p> <p>Size. The total number of objects in the collection at the end of the sample period.</p> <p>Class. The class name for the collection.</p> <p>To see details for the collection, hold the mouse pointer over the row in the table for the collection. The row is highlighted in pink and the details are displayed in the Samples and Collections Details pane.</p>

Part VI

Integrations

39

Diagnostics Integration with Other HP Software Products

This chapter describes integrations between Diagnostics and other HP Software products.

This chapter includes:

Concepts

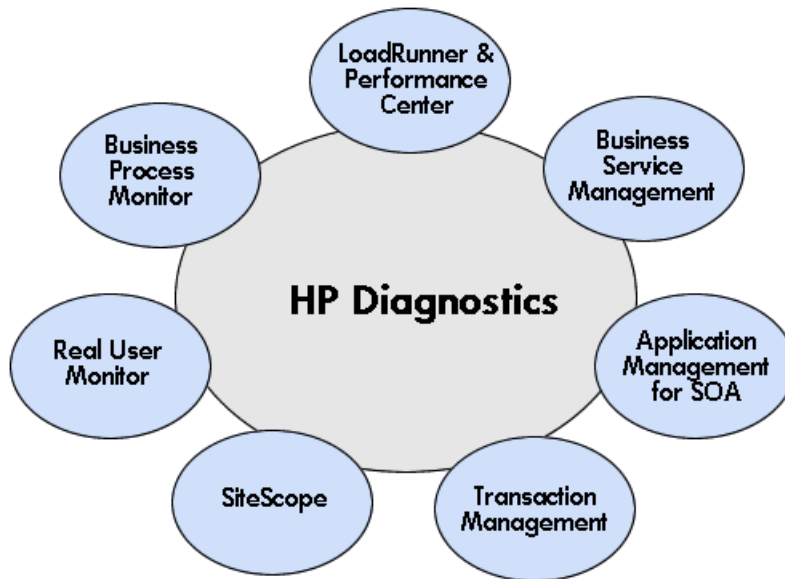
- ▶ Diagnostics Integrates with other HP Software Products on page 1085
- ▶ Diagnostics and Business Service Management on page 1086
- ▶ Overview of Business Service Management Concepts for Diagnostics Users on page 1088
- ▶ Overview of Diagnostics Data Provided to Business Service Management on page 1090
- ▶ CI Population and Models on page 1091
- ▶ Run-time Service Model on page 1093
- ▶ Event Based and Metric Based Coloring/HIs on page 1097
- ▶ Data Samples Sent to Business Service Management on page 1097
- ▶ Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics on page 1098
- ▶ Application Infrastructure HI Status Set Based on Events from Diagnostics on page 1100
- ▶ Open the Diagnostics UI and Online Help from Business Service Management on page 1106

- ▶ The Diagnostics View in Business Service Management Service Health on page 1108
- ▶ Business Transactions View in Business Service Management Service Health on page 1110
- ▶ General Integration between Diagnostics and Business Service Management on page 1112
- ▶ Integration between Diagnostics and End User Management on page 1114
- ▶ Integration between Diagnostics and Transaction Management on page 1115
- ▶ Integration between Diagnostics and Application Management for SOA on page 1117
- ▶ Diagnostics Configuration that can Affect Integration with Business Service Management on page 1118
- ▶ Integration Between Diagnostics and SiteScope on page 1121
- ▶ Integration Between Diagnostics and LoadRunner on page 1122
- ▶ Analyze Offline Data Provided by Diagnostics on page 1122
- ▶ Integration Between Diagnostics and Performance Center on page 1124

Concepts

Diagnostics Integrates with other HP Software Products

HP has invested significantly in integrating Diagnostics with other HP Software applications and components.



Diagnostics integrates with the following HP Software products and applications:

- ▶ HP Business Service Management - Access Diagnostics and drill down to Diagnostics data from reports and Service Health.
 - ▶ Application Management for SOA Application - Drill down to Diagnostics views from SOA reports. Diagnostics also provides data for the SOA view for web services monitored by Diagnostics.
 - ▶ Real User Monitor - Drill down to Diagnostics server request details.

- ▶ Business Process Monitor - Diagnostics monitors transactions generated by Business Process Monitor scripts and autogenerates applications for these transactions. Diagnostics provides data to give the backend view for the Business Process Monitor transaction. Drill down to Diagnostics Synthetic Transactions view.
- ▶ Transaction Management Application - Users can configure business transactions for monitoring in Transaction Management by selecting top level server requests from Diagnostics. Also users can drill down to Diagnostics transaction views.
- ▶ HP TransactionVision - Diagnostics and TransactionVision both use the Java Agent and .NET Agent for data collection.
- ▶ HP SiteScope - SiteScope data displayed in Diagnostics.
- ▶ HP LoadRunner/HP Performance Center - Diagnostics provides transaction, server request, layer and probe data metrics in the offline analysis report.

If you own any of these other HP software products already, then your investment in Diagnostics can increase the value you receive in using them.

And if you don't own any of these other HP software products today, then if you invest in Diagnostics, realize that if in future you are faced with a problem in any of these areas, HP has a solution that ties easily with Diagnostics and increases the value of your investment in it.

Diagnostics and Business Service Management

Diagnostics can be integrated with Business Service Management to provide information to help you understand and improve the performance of your J2EE and .NET applications.

Diagnostics 9.00 can be integrated with Business Service Management 8.0 or 9.0 but not with Business Service Management versions earlier than 8.0.

Note: This documentation is relevant for integration with Business Service Management 9.0

Caution: Anytime a Business Service Management system is upgraded or re-installed, a manual hard sync is needed (or a wait period of 24 hours) before CIs from Diagnostics are forwarded to Business Service Management. To do a hard sync, go to the Diagnostics Server Administration page, select **Synchronize** and then select **Hard** for **All Customers**.

From within Business Service Management, you can actively track the performance status of the applications being monitored by Diagnostics. The Diagnostics integration with Business Service Management allows you to:

- ▶ access the Diagnostics UI from within Business Service Management
- ▶ drill down to Diagnostics views from specific Business Service Management configuration items such as infrastructure elements and transactions (synthetic and any other)
- ▶ generate high level reports in Business Service Management about the performance of business transactions that are monitored by Diagnostics and view business transaction data from Diagnostics in the Transaction Management application, including topologies
- ▶ see Diagnostics information for monitored web services and operations in Application Management for SOA reports
- ▶ view the application infrastructure information from Diagnostics in the context of a business application or business transaction
- ▶ see status of specific Application Server Configuration Items that Diagnostics monitors
- ▶ configure business transactions for monitoring based on top level server requests in the Transaction Management administration application

See the *HP Diagnostics Installation and Configuration Guide* for information on how to configure the integration between Diagnostics and Business Service Management.

Overview of Business Service Management Concepts for Diagnostics Users

Business Service Management helps you model your business processes and services by providing a framework for mapping the complex and dynamic dependencies between applications and their supporting infrastructure. Business Service Management is integrated with an HP Universal Configuration Management Database, referred to in Business Service Management as the **Run-time Service Model (RTSM)**. The Run-time Service Model is a repository storing information on services, their components (or configuration items), details about configuration item attributes, the history of each configuration item and details of the relationships between configuration items. A **Configuration Item (CI)** is a component that makes up a part of your IT environment.

In your IT environment you have many services that depend on many applications and databases running on application infrastructure that uses a variety of server and network hardware. All this is interdependent. The interdependencies are represented as **relationships** or connections between CIs. The CIs, relationships and dependencies form a model of all the components of your company's IT infrastructure (**IT Universe**). The Run-time Service Model is a repository for information about your IT Universe.

The IT Universe model can be very large with millions of CIs. In Business Service Management you select and manage the CIs using **views**. Each view provides a subset of the overall components in the IT Universe model. The CIs contained in the view are displayed in a hierarchical tree format according to the relationships defined between the CIs.

Business Service Management applications and related HP software applications such as Diagnostics, **populate the model** by adding CIs into the Run-time Service Model for the infrastructure information they have.

Business Service Management uses **data collectors** (such as Diagnostics agents) to collect data from various points throughout the IT infrastructure as well as from external locations and feed it into the central repository (Run-time Service Model).

Some key CIs have **Key Performance Indicators (KPIs)** such as Volume or Availability which indicate the status of a CI. **Health Indicators** for a CI are more granular status indicators which are rolled up to KPIs. The status and value of these indicators is based, in part, on information from the data collectors and shows how well the CI is performing.

The status for a Health Indicator or KPI is displayed in Business Service Management in views such as Service Health (aka Dashboard). The **indicator color** shows you if the Health Indicator or KPI measurement is acceptable (green), critically failing (red), or is at some business risk level between the two.

Some Health Indicators take their status directly from an event (**event-based Health Indicators**). Status (coloring) for event based Health Indicators is sent to Business Service Management from Diagnostics when there is a threshold violation on relevant metrics. Diagnostics provides event data based on thresholds (on CIs) set in Diagnostics. Events are not stored in Business Service Management.

Other Health Indicators and KPIs use business rules to calculate status based on the data samples sent from Diagnostics' data collection/agents (**metric-based Health Indicators**).

For business transaction and web service CIs populated by Diagnostics - KPI and Health Indicator status is metric-based, Diagnostics sends data samples to Business Service Management and rules in Business Service Management are used to evaluate the data and set the indicator's status.

Overview of Diagnostics Data Provided to Business Service Management

Diagnostics agents collect data from applications, application servers, host computers and transactions. This data is used to populate CIs in Business Service Management and provide information on the relationships between CIs in order to form impact relationships and topologies in Business Service Management.

Note: The following is relevant for integration with Business Service Management 9.0

- ▶ In Business Service Management 9.00, the Probe CI is replaced by much more granular CIs and relationships representing specific application infrastructure.
- ▶ Diagnostics sends samples to Business Service Management which are persisted in the BSM database.
- ▶ Metric data from samples is used in Business Service Management to determine status of KPIs and Health Indicators. Threshold violations (events) from Diagnostics are also used in Business Service Management to determine status of KPIs and Health Indicators.
- ▶ In addition to populating CIs for application infrastructure data, Diagnostics also provides top level server requests to Business Service Management that you can select in the Transaction Management application to configure business transactions for monitoring.
- ▶ Once you connect the starting (top level) server requests to a business transaction in Transaction Management all the infrastructure components that the transaction passes/traverses can be seen in Business Service Management in the context of the transaction. In addition, Diagnostics sends the aggregate business transaction duration to Business Service Management which can be seen in the Transaction Management application.

CI Population and Models

Diagnostics populates an extensive set of application infrastructure, web service and business transaction CIs in the Business Service Management Run-time Service Model (RTSM) and provides information on relationships between CIs in common data models.

The following is a table showing the CI Types, model relationships and CIs Diagnostics populates.

CI Type	Display Name	Relationship to	Destination CI
host_node	Computer	composition composition composition composition composition composition composition	database interface ip_address application_server webspheremq infrastructure_element ip_service_endpoint
unix (host_node)	Unix		
nt (host_node)	Windows	composition composition	iis process
interface	Interface		
ip_address	IpAddress		
ip_service_endpoint	IpServiceEndpoint		
application_server	ApplicationServer	deployed usage	ip_service_endpoint
websphereas (application_server)	Websphere AS		
weblogicas (application_server)	Weblogic AS		
Jbossas (application_server)	JBoss AS		

CI Type	Display Name	Relationship to	Destination CI
oracleias (application_server)	Oracle iAS		
sap_r3_server (application_server)	SAP ABAP Application Server		
sap_j2ee_app_server (application_server)	SAP J2EE Application Server		
j2eeserver (application_server)	J2EE Server		
J2eeapplication	J2EE Application	dependency	J2eeapplication
j2eedomain	J2EE Domain	composition membership	J2eeapplication application_server
webspheremq	IBMWebSphereMQ	composition	mqueuemanager
mqueuemanager	WMQ Queue Manager	composition	J2EE mqueue
mqueue	MQ Queue		
oracle (database)	Oracle		
sqlserver (database)	SQL Server	composition	sqldatabase
sqldatabase	MSSQL Database		
webservice	Web Service	composition dependency dependency	webservice_operation webservice infrastructure_element
webservice_operation	Web Service Operation		
iis	IIS Web Server	composition composition	iiswebsite iisapppool
netappdomain	.NETAppDomain		
iiswebsite	IIS Web Site	usage composition	iisapppool iiswebdir
iisapppool	IIS Application Pool	dependency	process

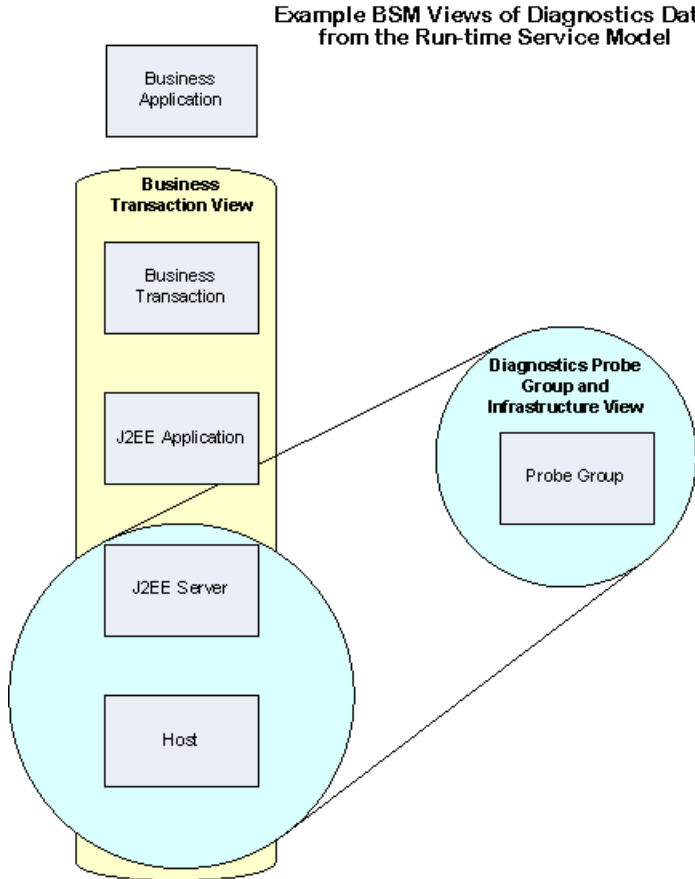
CI Type	Display Name	Relationship to	Destination CI
iiswebdir	IIS Web Dir	deployed dependency	iisapppool netappdomain
iisvirtualdir (iiswebdir)	IIS Virtual Dir		
process	Process	composition	netappdomain
infrastructure_element	InfrastructureElement	deployed	diagnostics_probegroup
diagnostics_probegroup	Diagnostics Probe Group	composition	infrastructure_element
diagnosticsprobe	HP Diagnostics Probe		
business_application	BusinessApplication	containment	transaction
business_transaction	BusinessTransaction	usage	infrastructure_element

Run-time Service Model

Business Service Management uses a common service model which means that data discovered or collected from various sources is identified and stored in the Run-time Service Model based on a common set of attributes.

Diagnostics populates many different CIs and relationship attributes in the Run-time Service Model as described in "CI Population and Models" on page 1091. You can view this data provided by Diagnostics in a number of views in Business Service Management.

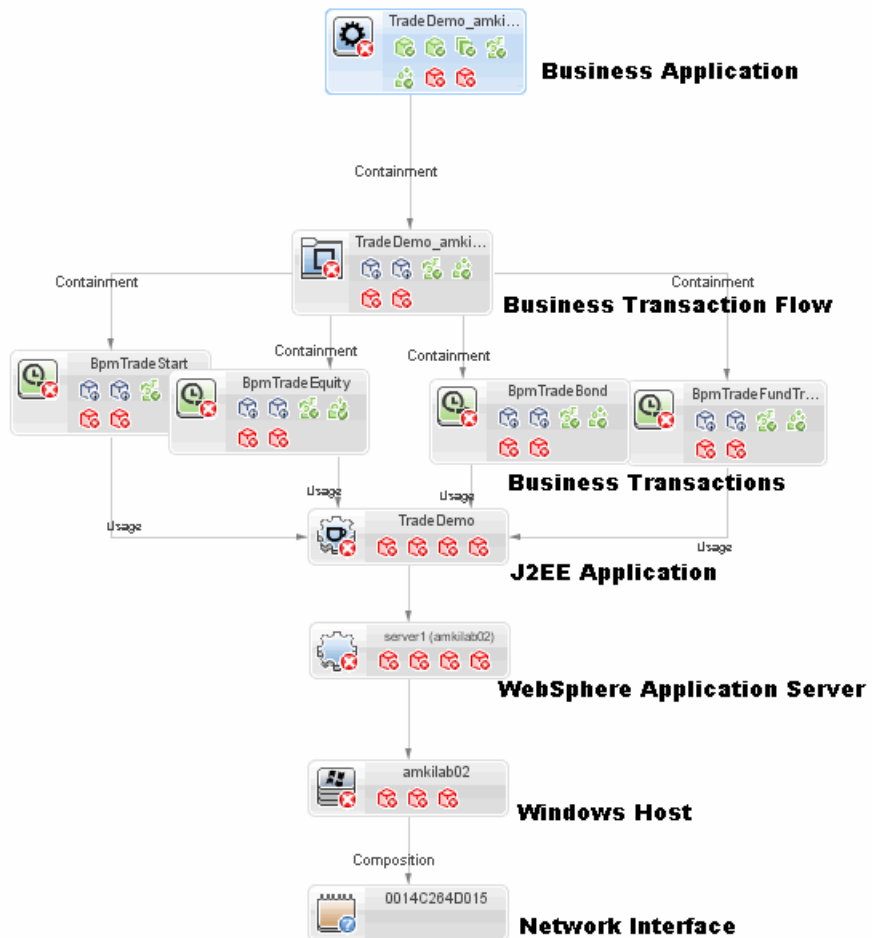
For example you can see infrastructure data provided by Diagnostics in Service Health in the **Diagnostics Probe Group and Infrastructure** view. And you can see transaction, application and infrastructure data provided by Diagnostics in Service Health in the **Business Transactions** view.



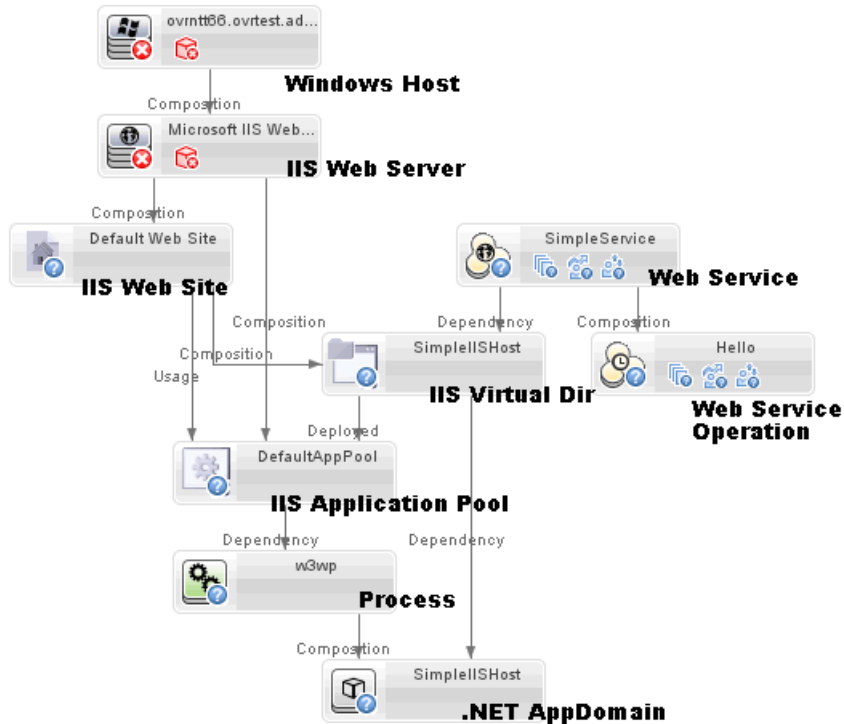
The Diagnostics Probe Group and Infrastructure model view displays the specific probe (as the more granular CI, for example WebSphere server, SQL DB), the probe group it is a member of and its host. The Business Transaction model view displays the business transaction, application, application server and host.

The following topologies give some examples of how you can view these CIs and relationships populated by Diagnostics in Business Service Management.

In the example below from Business Service Management **Service Health > Topology View > Business Transactions** view you can see the following CI types and relationships populated by Diagnostics: business application, business transaction, J2EE application, WebSphere application server and Windows host.



In the example below from Business Service Management **Service Health > Topology View > DotNetIIS Topology** view you can see the following CIs and relationships populated by Diagnostics: Windows host, IIS Web Server, IIS Web Site, Web service, Web service operation, IIS Virtual Dir, IIS application pool, process, .NET appdomain.



Event Based and Metric Based Coloring/HIs

Status (coloring) for event based Health Indicators is sent to Business Service Management from Diagnostics when there is a threshold violation on relevant probe and host metrics. Diagnostics provides event data based on threshold violations (on CIs) set in Diagnostics. Events are not stored in Business Service Management. See "Application Infrastructure HI Status Set Based on Events from Diagnostics" on page 1100.

Status for metric based KPIs and Health Indicators is sent to Business Service Management from Diagnostics in data samples. HI status for business transaction and web service CIs populated by Diagnostics is metric-based. Diagnostics sends data samples to Business Service Management and rules in Business Service Management are used to evaluate the data and set the indicator's status. See "Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics" on page 1098.

Data Samples Sent to Business Service Management

Diagnostics sends **Data Samples** which are persisted in the Business Service Management Profile database. Data samples are sent for Web Services (SOA samples) and for Transactions. Business Service Management applies rules to the data in these samples in order to color KPIs and HIs for related CIs. See "Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics" on page 1098.

- ws_perf_aggr_t (SOA sample)
- ws_event_aggr_t (SOA sample)
- appmon_vu_t (Transaction (BPM) sample)
- dg_trans_t (Business Transaction (Diagnostics) sample)

Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics

Health Indicator status for Business Transactions, Synthetic Transactions and Web Services CI Types is set based, in part, on data samples sent from Diagnostics to Business Service Management (see "Data Samples Sent to Business Service Management" on page 1097).

Samples are used for coloring status when there are other consumers of the data such as Service Health, reports, etc.

Data from the samples is evaluated based on Business Service Management rules and the results set KPI and Health Indicator coloring.

When the status is sample based, you can change default objectives for business transaction, synthetic transaction and Web service Health Indicators in Business Service Management **Admin > Service Health**. See the *Business Service Management Documentation Library* for information on using Service Health Admin.

The table shows the CI Type, Health Indicators and KPIs that Diagnostics contributes to in Business Service Management Service Health for the Business Transaction and Web Service CI types.

CI Type	Health Indicator	Contributing to KPI
Business Transaction	Backend Transaction Average Duration	Yes (Application Performance)
Business Transaction	Backend Transaction Count	Yes (Volume)
Business Transaction	Backend Transaction Minimum Duration	No
Business Transaction	Backend Transaction Maximum Duration	No
Business Transaction	Backend Transaction Application Exceptions	No
Business Transaction	Backend Transaction SOAP Faults	No
Business Transaction	Synthetic User Backend Average Response Time	No

CI Type	Health Indicator	Contributing to KPI
Business Transaction Synthetic	Synthetic User Backend Application Exceptions	No
Business Transactions Synthetic	Synthetic User Backend Max Server Request Duration	No
Web Service Operation	Operation Average Response Time	Yes (Performance)
Web Service Operation	Operation Availability	Yes (Availability)
Web Service Operation	Operation Throughput	Yes (Throughput)

And the following Health Indicators are shared with Transaction Management and Transaction Management has precedence in coloring if the same business transaction is monitored by both Diagnostics and Transaction Management: Backend Transaction Average Response Duration, Backend Transaction Minimum Response Duration, Backend Transaction Maximum Response Duration, Backend Transaction Count Rule.

The table shows the CI Type, Health Indicators and KPIs that Diagnostics contributes to in Business Service Management SLM for the Business Transaction and Web Service CI types.

CI Type	Health Indicator	Contributing to KPI
Business Transaction	Backend Transaction Average Duration	Yes (Throughput)
Business Transaction	Backend Transaction Count	Yes (Performance)
Business Transaction	Backend Transaction Minimum Duration	No
Business Transaction	Backend Transaction Maximum Duration	No

Application Infrastructure HI Status Set Based on Events from Diagnostics

KPI and Health Indicator status for CI Types listed in the table below is set based on events. This is lighter weight than sending data samples as the event data from Diagnostics is not stored in Business Service Management. Events are triggered based on threshold violations in Diagnostics.

For an event to be triggered, you must set a threshold on the metric in Diagnostics and then Diagnostics evaluates the probe data to identify threshold violations.

When the value for a metric in a **metric category** violates a set threshold, the category is colored red in the Diagnostics Details pane and the entity's status is also colored red. The metrics and metric categories can also be seen in the **metrics.config** file for the Diagnostics Java agent or .NET agent.

Diagnostics sends data on these threshold violation events to Business Service Management to be used in coloring the corresponding Health Indicators.

So for example, in the Details pane below you see the GC metric category is colored red because of a threshold violation. When event information is sent to Business Service Management you would see the GC Health Indicator for this CI in Business Service Management colored to show a similar status.

The screenshot shows a 'Details' window with a search bar and a list of performance metrics. The 'GC' section is highlighted in red. The metrics are as follows:

Metric	Value	Icons
Execute Queues		
Idle Threads (Avg)	0	Warning, Error, Info icons
Pending Requests (Avg)	0	Warning, Error, Info icons
Requests / sec (Avg)	0	Warning, Error, Info icons
Total Threads (Avg)	3	Warning, Error, Info icons
GC		
Collections/sec (Avg)	0	Warning, Error, Info icons
Time Spent in Collections % Over Threshold	16.2%	Warning, Error, Info icons
Time Spent in Collections (Avg)	6%	Warning, Error, Info icons
JDBC		
Active Connections (Avg)	3	Warning, Error, Info icons
Available Connections (Avg)	5	Warning, Error, Info icons

The event data is sent to Business Service Management using the OM agent and IAPA components installed with the Diagnostics commanding server. See the *HP Diagnostics Installation and Configuration Guide* Diagnostics Server installation chapter for information on these components.

Metric category names in Diagnostics mostly correspond to Health Indicators of the same name in Business Service Management. The Health Indicator name that Diagnostics uses (default metric category name) can be overridden in the `server.properties` file.

The HI rule used for coloring all these HIs shown in the table below is the **Diagnostics for J2EE/.Net General** rule.

CI Type	Health Indicator	Contributing to KPI
.NET AppDomain	Latency Heap Probe	Yes (Software Performance)
WebSphere AS	Dynamic Caching EJB GC Statistics J2C Connections JDBC Connection Pools JMS MDB Object Pool ORB Servlet Session Manager Thread Pools Transaction Manager Web Applications Web Services Web Services Gateway WPS Metrics Probe Heap GC Latency VM Thread	Yes (Software Performance)
WebSphere MQ	Totals (Queues) Probe Total (Channels)	Yes (Software Performance)

CI Type	Health Indicator	Contributing to KPI
WebLogic AS	EJB Execute Queues JDBC JMS JRockit JTA Servlets Servlet Session Manager Web Services Web Sessions Probe Heap GC Latency VM Threads	Yes (Software Performance)
JBOSS AS	JBoss Server JCA Session Manager Thread Pool Transaction Manager Probe Heap GC Latency VM Threads	Yes (Software Performance)
SQL Server	SqlServer Probe	Yes (Software Performance)
SQL Database	SqlServer Probe	Yes (Software Performance)

CI Type	Health Indicator	Contributing to KPI
Oracle iAS	EJB HTTP Server JSPs Servlets Task Manager Web Context Web Module Probe Heap GC Latency VM Threads	Yes (Software Performance)
Oracle	Oracle Probe	Yes (Software Performance)
SAP J2EE Application Server	Application Threads HTTP JNDI P4 Pool Manager Portal Runtime Portal Cache Portal Pool Security System Threads Timeout Transactions Probe	Yes (Software Performance)
SAP R3 Server	Probe R3	Yes (Software Performance)
Host Node	Disk Network CPU Memory Threads	Yes (System Performance)

CI Type	Health Indicator	Contributing to KPI
Diagnostics Probe (for 8.0 or earlier probes where the application server type cannot be identified)	Probe Heap Dynamic Caching EJB GC Statistics J2C Connections JDBC Connection Pools JMS MDB Object Pool ORB Servlet Session Manager Thread Pools Transaction Manager Web Applications Web Services Web Services Gateway WPS Metrics GC Latency VM Threads JBoss Server JCA Session Manager Thread Pool Transaction Manager HTTP Server JSPs Servlets Task Manager Web Context Web Module Global Request Processor JDBC Oracle SqlServer R3 Application Threads	Yes (Software Performance)

CI Type	Health Indicator	Contributing to KPI
Diagnostics Probe (continued)	HTTP JNDI P4 Pool Manager Portal Runtime Portal Cache Portal Pool Security System Threads Timeout Transactions .NET	

Open the Diagnostics UI and Online Help from Business Service Management

You can open the Diagnostics UI and the Diagnostics online help directly from Business Service Management.

This section includes:

- "Open the Diagnostics UI from Business Service Management" on page 1106
- "Open the Diagnostics Online Help from Business Service Management" on page 1107

Open the Diagnostics UI from Business Service Management

From Business Service Management you can open the Diagnostics UI in one of the following ways:

- Select **Applications > Diagnostics**.

- On the **Site Map**, click the **Diagnostics** link.
- Select to drill down to Diagnostics from various Business Service Management context menus and buttons to open Diagnostics UI in the context of the data displayed in Business Service Management.

Once the Diagnostics UI is displayed in the Business Service Management window you can select Help from the Diagnostics menu to get information on using Diagnostics. You can also select Help on View to get detailed information on the specific Diagnostics view displayed (context sensitive help).

Open the Diagnostics Online Help from Business Service Management

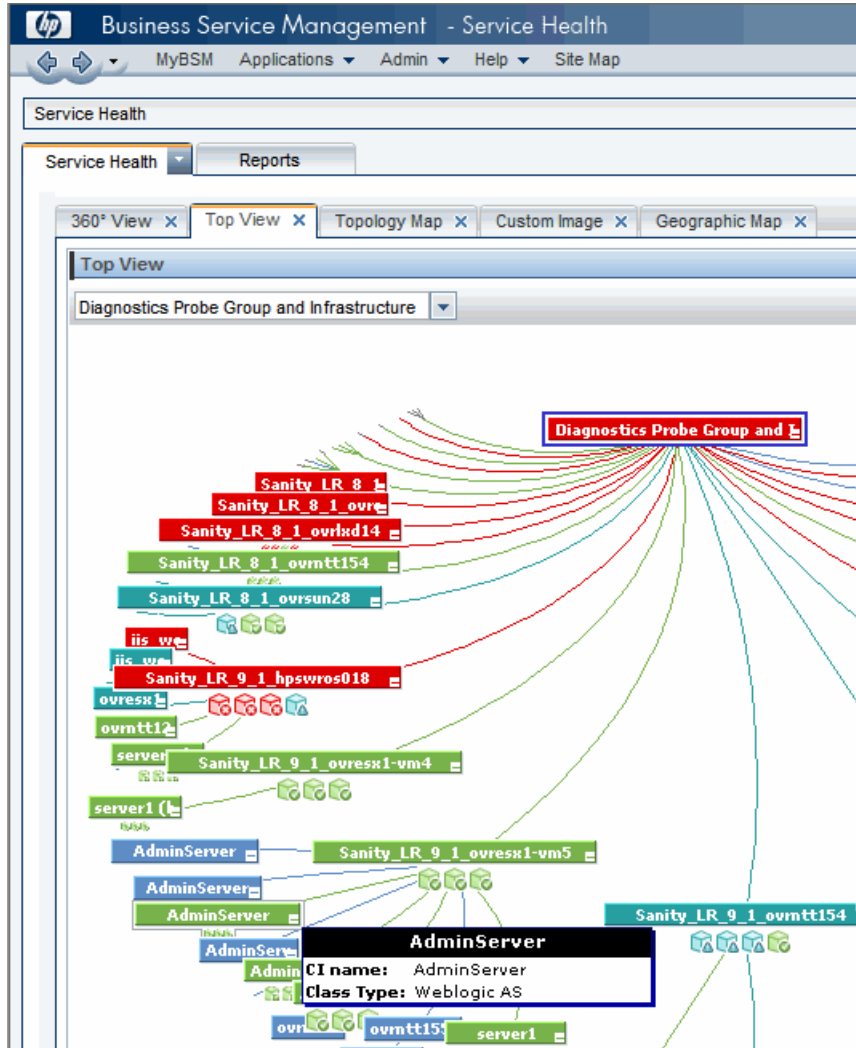
From Business Service Management you can open the Diagnostics online help as follows:

- From the Business Service Management menu Select **Help > Diagnostics Help**.
- A standalone version of the Diagnostics online help opens in a separate window.

Of course, if you have opened the Diagnostics UI from Business Service Management you can access the Diagnostics help from the Diagnostics menu.

The Diagnostics View in Business Service Management Service Health

In Business Service Management Service Health you can see the **Diagnostics Probe Group and Infrastructure** view showing data from Diagnostics. The example below of the Top View shows the probe groups, and associated application infrastructure being monitored (such as WebLogic AS) and the NT or UNIX hosts.



From Business Service Management's Service Health, you can also actively track the performance status of the applications being monitored by HP Diagnostics.

You track performance status in Service Health using Key Performance Indicators (KPIs) and Health Indicators. The KPIs and Health Indicators provide real-time assessment of the present status of you application, enable you to track critical performance variables over time, and help assess the impact of problems in the application.

The screenshot displays the HP Business Service Management Service Health interface. The top navigation bar includes 'hp Business Service Management - Service Health', 'Full Screen View', and 'User: administrator'. Below the navigation, there are tabs for 'Service Health' and 'Reports'. The main content area shows a 'hierarchy' view with a table of servers. The table has columns for 'Name', 'Software Performance', and 'System Performance'. The 'MedRecServer (ovresx4-vm7.ovrtest.adapps.hp.com)' is highlighted, showing a red 'X' icon for Software Performance and green checkmarks for System Performance. Below the hierarchy, there are tabs for 'Business Impact', 'KPIs', 'Health Indicators', 'Alerts', and 'Changes And Incidents'. The 'Health Indicators' tab is active, showing a table of health indicators for the selected server. The table has columns for 'KPI', 'Health Indicator', 'State', 'Value', 'Last Status Change', and 'Trend'. The 'Software Performance' KPI is highlighted, showing three health indicators: 'Probe' (Normal), 'Latency' (Critical), and 'GC' (Normal).

Name	Software Performance	System Performance
MedRecServer (ovresx4-vm7.ovrtest.adapps.hp.com)	❌	✅
ovresx4-vm7.ovrtest.adapps.hp.com	-	✅

KPI	Health Indicator	State	Value	Last Status Change	Trend
❌ Software Performance	Probe	✅ Normal		4/16/10 3:23 AM	↑
	Latency	❌ Critical		4/20/10 12:30 PM	↓
	GC	✅ Normal		4/15/10 9:09 PM	↑

The Software Performance KPI and Health Indicators are colored

(color-coded icon indicating status) based, in part, on information from Diagnostics. See "Application Infrastructure HI Status Set Based on Events from Diagnostics" on page 1100 for details.

You can see the Software Performance KPI which is associated with specific Health Indicators for each Probe Metric Category.

Note: You can drill down to the Diagnostics screens that display Diagnostics data. For more information, see "General Integration between Diagnostics and Business Service Management" on page 1112.

Business Transactions View in Business Service Management Service Health

In addition to the Diagnostics Probe Group and Infrastructure view, you can also select the Business Transactions view to monitor business transaction performance from Service Health.

Once you have enabled Diagnostics breakdown, you can monitor the performance status of transactions from Business Process Monitor and see backend status based on Diagnostics data. See "Business Transaction and Web Service HI Status Set Based on Data Samples from Diagnostics" on page 1098 for business transaction HI descriptions. This functionality is available starting with Business Process Monitor 9.x.

The example shows a BPM transaction with HI status for the backend activity provided by Diagnostics.

hp Business Service Management - Service Health Full Screen View User: administrator Log

MyBSM Applications Admin Help Site Map

Service Health Reports

360° View x Custom Image x Geographic Map x Top View Select Page

Hierarchy [Select a Filter]

Name	Application Performance	Application Availability	Volume	Software Performance
BpmTradeBond	✔ ⬆	✔ ⬆	-N/A-	
TradeDemo	-	-	-	
server1 (amkilab02.ovrtest.adap	-	-	-	
Sanity_TV75_ovruxt32	-	-	-	
amkilab02.ovrtest.adapps.hp.c	-	-	-	

Last Updated: Tuesday, April 20, 2010 2:12:05 PM

Business Impact KPIs Health Indicators Alerts Changes And Incidents

BpmTradeBond

Health Indicators Contributing to KPIs

KPI	Health Indicator	State	Value	Last Status Change	Trend
-N/A- Volume	Backend Transaction Count	-		-	↑
	Backend Transaction Average Duration	-		-	↑
✔ Application Performance	Synthetic User Transaction Performance	✔ OK	612	4/20/10 1:47 PM	↑
✔ Application Availability	Synthetic User Transaction Availability	✔ OK	100	4/20/10 3:27 AM	↑

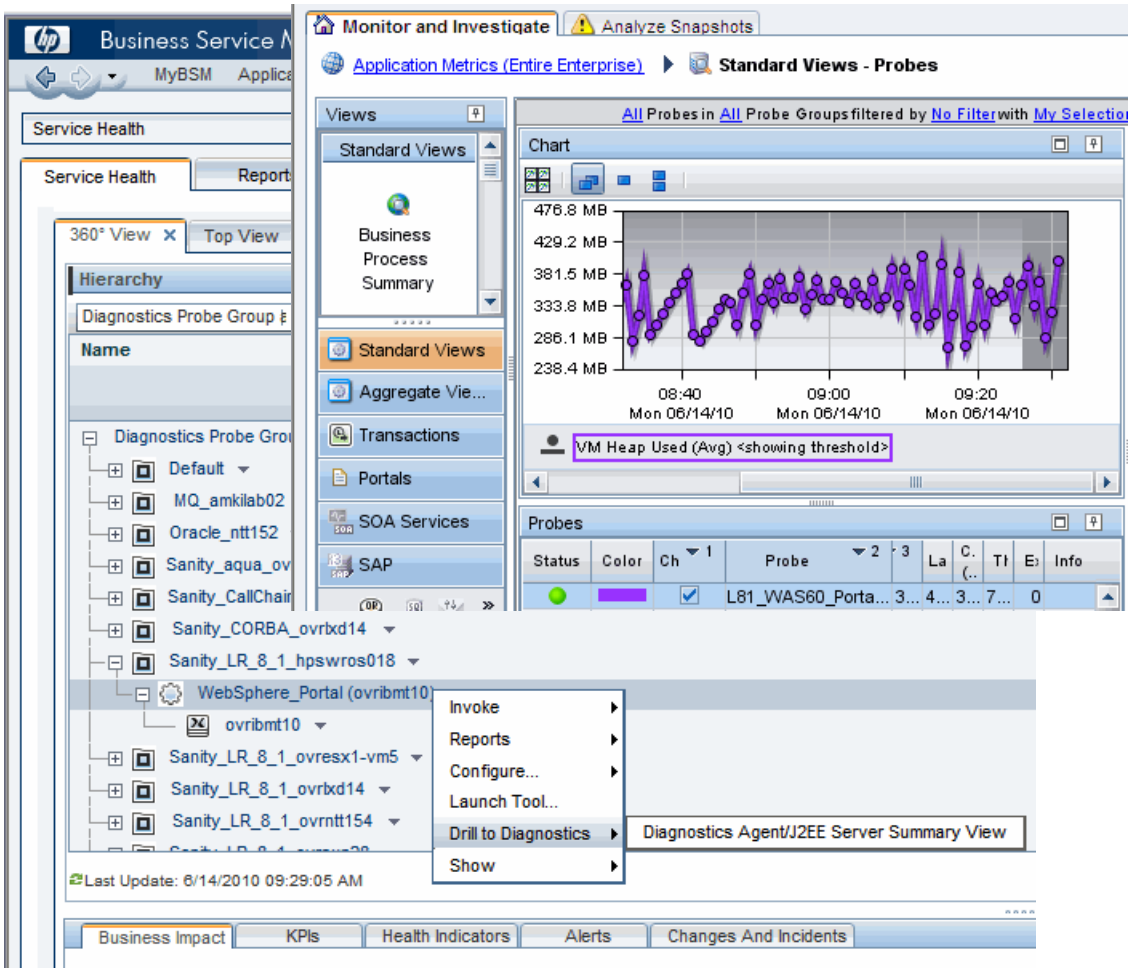
Health Indicators Not Contributing to KPIs

Health Indicator	State	Value	Last Status Change	Trend
Backend Transaction Application Exceptions	-		-	↑
Backend Transaction Maximum Duration	-		-	↑
Backend Transaction Minimum Duration	-		-	↑
Backend Transaction SOAP Faults	-		-	↑
Synthetic User Backend Application Exceptions	-	0	4/8/10 10:24 AM	↑
Synthetic User Backend Average Response Time	-	0.2 sec	4/8/10 10:24 AM	↑
Synthetic User Backend Max Server Request Duration	-	0.594 sec	4/8/10 10:24 AM	↑

General Integration between Diagnostics and Business Service Management

In Business Service Management, you can drill down from selected CIs (Configuration Items) to the Diagnostics UI views displaying data about that item. In general, drill downs are available from the application infrastructure elements in Business Service Management to Diagnostics views such as the Probes view, Hosts view, Server Requests view or specialized views such as Oracle Database-Oracle Probes view for the Oracle Database CI.

The following example shows a drill down from a WebSphere application server CI to the Diagnostics Probes view displaying data from the probe monitoring that application server.



Another example shown below illustrates a drill down from a business transaction CI to Diagnostics transactions views.

The screenshot displays the HP Business Service Manager interface. The main window is titled 'Monitor and Investigate' and shows 'Application Metrics (Entire Enterprise)'. The left sidebar contains a 'Hierarchy' tree with 'Business Transactions' selected. The main area features a 'Chart' showing 'Latency (Avg) <showing threshold>' over time, with a red horizontal threshold line at 1.0 s. Below the chart is a table with columns: Status, Color, Ch, Transaction, BP Profil, Thro, and CPU (Avg). A context menu is open over the table, listing options: Invoke, Reports, Configure..., Launch Tool..., Drill to Diagnostics, and Show. The 'Drill to Diagnostics' option is highlighted, with a sub-menu showing 'Synthetic Users Transactions View' and 'Synthetic Users Transactions Layers View'.

Integration between Diagnostics and End User Management

Drill downs to Diagnostics from End User Management are described below.

Business Process Monitor-related views. From selected business transactions in Business Process Monitor related views, you can drill down to the Diagnostics Synthetic Transactions view, which displays the backend performance metrics for the synthetic transactions that are being executed by your applications. You can also drill down directly into the view displaying the layers for the selected transactions.

For details about the Diagnostics Synthetic Transactions view, see "Synthetic Transactions View Description" on page 607.

Real User Monitor-related views. From certain reports in Real User Monitor (RUM) related views, you can drill down to Diagnostics Server Requests, which displays the performance metrics for the monitored server requests in your application. The drill down to Diagnostics displays a snapshot with details for the corresponding server requests. In the snapshot, you can select any of the instance tree icons to drill down further to a call profile.

Note: If no server requests are found when drilling down to Diagnostics, there will be no data displayed in the snapshot and an error message in a yellow box will be displayed indicating the server request was not found.

For more information about the Diagnostics Server Requests, see "Server Requests View" on page 379 and for more information about Diagnostics snapshots see Chapter 6, "Performing Snapshot Analysis".

Integration between Diagnostics and Transaction Management

In the Transaction Management application in Business Service Management when you configure a business transaction for monitoring, you do this by selecting server requests to monitor. The server requests you can select, come from Diagnostics. And Diagnostics provides the monitoring functionality.

Additional integration is provided through drill downs to Diagnostics available from the following Transaction Management reports:

- ▶ Event Analysis (to Diagnostics Host view)
- ▶ Transaction Detail (to Diagnostics Server Requests snapshot analysis view)
- ▶ Transaction Summary (to Diagnostics Business Transaction Paths view)
- ▶ Aggregate Topology (to Diagnostics Business Transaction Paths view)

The drill downs to Diagnostics from Transaction Management are described below.

- ▶ The drill down to a Diagnostics Hosts view provides system metrics like CPU utilization and memory utilization that can help you determine if the problem is a system issue. From this Hosts view in Diagnostics you can see the performance of the host system charted over various time periods. Also you can check to see if the system has recently experienced a performance degradation or if the system has consistently under performed.
- ▶ The drill down to a Diagnostics snapshot of server request details can help you study the performance of these server requests over time and look at detailed instance trees (call profiles, methods, exceptions and faults) associated with the server requests to identify where the issue lies. The drill to Diagnostics icon will be displayed whenever there is a server request associated with the business transaction or the transaction step.
- ▶ From the Server Request view in Diagnostics you can navigate to the corresponding probe entity and look at the application server metrics (for example, JMX metrics) to see if the issue lies with the application server.
- ▶ The drill down to a Diagnostics Business Transaction Paths view provides a superimposed area chart of average latency for each path of the selected business transaction. A Business Transaction Path represents the aggregate latency from the starting Server Request (Path Start) to an end point of the transaction (Path End).

Note: If no server requests are found when drilling down to Diagnostics, there will be no data displayed in the snapshot and an error message in a yellow box will be displayed indicating the server request was not found.

If the server request is not found when drilling down to Diagnostics it is most likely due to latency trimming on the Diagnostics side where server requests that take less than 51ms are not collected. Latency trimming is configurable on the probe. For the Java Agent this is changed by setting **minimum.fragment.latency** to 0ms in the **dispatcher.properties**. For the .NET Probe this is changed by setting `<trim><latency min="0" /></trim>` in the **probe_config.xml** file.

Another reason you might not get a snapshot displayed is if the transaction you drilled down from in Transaction Management is a forwarded event (HTTP redirection with HTTP status codes in the 300s). This is because Diagnostics does not track forwarded requests.

For details about the Diagnostics Business Transaction Path view, see "Business Transactions Paths View Description" on page 597.

For details about the Diagnostics Hosts view, see "Hosts View Description" on page 373.

For more information about the Diagnostics Server Requests, see "Server Requests View" on page 379 and for more information about Diagnostics snapshots see Chapter 6, "Performing Snapshot Analysis".

Integration between Diagnostics and Application Management for SOA

Drill downs to Diagnostics from Application Management for SOA are described below.

Application Management for SOA-related views. From selected CIs in the Application Management for SOA-related views, you can drill down to the Diagnostics Web Service Topology view (SOA Services: Service Topology), which displays a topology view of monitored Web services and service connections or drill down to Diagnostics Operations view (SOA Services: Operations).

The Diagnostics Service Topology, shows each instance of the Service on a probe. If the Service you drill down from is hosted on multiple probes, Service Topology will display the topology for the first service that matches the drill down parameters. You can distinguish between service instances using the Probe column in the Services Topology entity table.

Drill down to Diagnostics is available from the following Business Service Management views:

- Applications > Application Management for SOA > Health Report
- Applications > Service Health > SOA Web Services view

For more information about the Diagnostics SOA Services view group, see "SOA Services Views" on page 651.

Diagnostics Configuration that can Affect Integration with Business Service Management

Diagnostics provides information to Business Service Management and this information is included in several Business Service Management reports. Some of the information from Diagnostics is configurable as described below.

- Anytime a Business Service Management system is upgraded or re-installed, a manual hard sync is needed (or a wait period of 24 hours) before CIs from Diagnostics are forwarded to Business Service Management. To do a hard sync, go to the Diagnostics Server Administration page, select **synchronize** and then select **Hard** for All Customers.
- In Diagnostics, you can configure consumers in a flexible way according to:
 - IP address or IP range.
 - HTTP Header.
 - SOAP request Headers, Body or Payload. SOAP payload is the entire SOAP envelope.

- ▶ JMS Queue name (or topic name) and JMS message header or JMS message property for SOAP over JMS web services.

IP address is used as the default consumer ID for SOAP over HTTP/S web services and inbound queue name (or topic name) is used as the default consumer ID for SOAP over JMS web services.

In Business Service Management for SOA reports such as the **Consumer Summary Report**, consumer information can be any of these as configured in Diagnostics. You can also configure the number of consumers to be monitored by probe. See "About Consumer IDs" on page 663 and the configuring consumer IDs section of the *HP Diagnostics Installation and Configuration Guide* for more information.

- ▶ The business transaction and server request topologies depend on the agents supporting cross-VM for the technologies involved. For example, if two server requests are communicating via HTTP, then the server request topology will be built correctly since Diagnostics supports HTTP cross-VM. Out of the box, Diagnostics supports cross-VM for HTTP, JMS and .NET Remoting.

RMI and Corba are also supported but they have to be enabled in the agents.

If using a different (maybe legacy) cross-VM technology, cross-VM support can also be added to Diagnostics using instrumentation points and code snippet changes (as of 9.0, it is possible to add cross-VM support to Diagnostics using only points and code snippets - however, this is only applicable to the java agent). When cross-VM support is added, then business transaction and server request topologies will work automatically.

- ▶ In Diagnostics, you can also configure the application server instance field in the Probe view's details pane to be something other than the probe name (application server instance name is used as the Endpoint in Business Service Management). In Business Service Management for SOA there is a **Server/Endpoint Summary Report** that displays application server instance as configured in Diagnostics. Note that blank web service namespaces and end_points are filled with "Unknown". See "How to Specify a Name for the Application Server Instance" on page 667 for information.

- ▶ In Diagnostics, you can specify a set of SOAP faults that should not be counted towards unavailability in Business Service Management. Out-of-the-box, only server-side faults (Receiver/Server) are counted in determining availability in Business Service Management. Client-side faults are excluded from availability calculations but are shown as SOAP Faults in the Business Service Management for SOA Health report.

By default, the following SOAP faults are excluded in the availability calculation used in Business Service Management: VersionMismatch, MustUnderstand, DataEncodingUnknown, Sender/Client.

In Diagnostics, you can configure the faults that should be excluded when calculating availability. You set the **soap.faults.excluded** property in the **server.properties** file and **dispatcher.properties** file for Java Probes. The .NET probe's aggregation is done on the server in the **server.properties** file so no additional changes need to be made for the .NET probes.

This must be configured the same for the whole deployment (all Diagnostics probes and servers). See the *HP Diagnostics Installation and Configuration Guide* for more information on the property files.

- ▶ If the server request is not found when drilling down to Diagnostics it is most likely due to latency trimming on the Diagnostics side where server requests that take less than 51ms are not collected. Latency trimming is configurable on the probe. For the Java Probe this is changed by setting **minimum.fragment.latency** to 0ms in the **dispatcher.properties**. For the .NET Probe this is changed by setting `<trim><latency min="0" /></trim>` in the **probe_config.xml** file. See the *HP Diagnostics Installation and Configuration Guide*.

There also may be no data when drilling down to Diagnostics if the application server handling a particular page is not monitored by a Diagnostics probe.

- ▶ There are a number of options when configuring the Diagnostics integration with RUM. A brief overview of the configuration is provided below, see the *HP Diagnostics Installation and Configuration Guide* for more information.
 - ▶ You can disable the RUM integration in case it is not used. For the Java probe, in **capture.properties** set the value of **rum.coloring.enabled** to false. For the .NET probe, in **probe_config.xml** configure the element to `<rum enabled="false" />`.

- ▶ Parameter aggregation is enabled by default in Diagnostics (**auto_detect.points** file [**HttpCorrelation**] element). You can define keys if you want to see data specific for some parameters. If you turn off parameter aggregation then all URLs in RUM that contain a parameter will drill down to the same server request in Diagnostics (for example, if RUM has /url?p=a and /url?p=b the corresponding server request in Diagnostics will be /url).
- ▶ You can configure Diagnostics to monitor post parameters in the **inst.properties** file by setting **ignore.post.parameters=false**.

Integration Between Diagnostics and SiteScope

If you set up the integration between SiteScope and Diagnostics (configured in SiteScope), then SiteScope sends data to Diagnostics. This SiteScope data is displayed in the Diagnostics External Monitors view group. See "External Monitors Views" on page 947 for information on this view and on configuration.

SiteScope can forward data to HP Diagnostics that allows the user to see a more complete view of the application servers that are monitored by Diagnostics and/or provide insight into the infrastructure components that these application servers are deployed into. For example, integrating data from the SNMP monitor can help determine problems with the infrastructure that the application server runs on.

Integration Between Diagnostics and LoadRunner

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

During a LoadRunner load test scenario, you can view Diagnostics data for the whole scenario or you can drill down to HP Diagnostics data from a particular transaction. Refer to the *HP LoadRunner Controller User Guide* for details about how to configure LoadRunner to access Diagnostics data during a load test scenario.

LoadRunner displays monitoring versions of the Synthetic Transactions view, Server Requests view, Load view and Probe view for the current run. Once you drill down to Diagnostics, you can navigate to other views in HP Diagnostics. The Business Transaction view is not available when drilling down to Diagnostics from LoadRunner.

Analyze Offline Data Provided by Diagnostics

LoadRunner Analysis provides offline graphs and reports with in-depth performance analysis information. Using these graphs and reports, you can pinpoint and identify the bottlenecks in your application and determine what changes need to be made to your system to improve its performance.

When you configure LoadRunner to integrate with Diagnostics, Diagnostics supplies data to LoadRunner in an .eve file generated for each run. In LoadRunner Analysis, you can add the **J2EE/.NET Server Diagnostics** graphs to display Diagnostics performance metrics that were captured during the load test scenario.

These graphs show you the performance of server requests and methods in the application you are monitoring. These include requests generated by virtual user transactions and by real users. And there are graphs showing exceptions and timeouts on the server.

The J2EE/.NET - Probe Metrics graph displays performance metrics collected by Diagnostics probes. Metrics can include JVM related data such as Heap usage and Garbage Collection, application server specific data, JDBC (Java Database Connectivity) metrics, and more. The following probe metric data is provided, by default, for offline analysis: HeapUsed, GC Collections/sec, GC time Spent in Collections. Use the **etc/offline.xml** files to specify the Diagnostics probe metrics you want to be included in the offline analysis (.eve) file.

See the chapter on "Setting Up LoadRunner and Diagnostics Integration" in the *HP Diagnostics Installation and Configuration Guide* for details on selecting probe metrics to include in the offline analysis (.eve) file.

For detailed information about using the HP Diagnostics graphs in LoadRunner Analysis, refer to the *HP LoadRunner Analysis User Guide*.

Note: The following features and functionality in LoadRunner Analysis are different from the Diagnostics online views:

- ▶ Data in the Profiler that is not sent into the Diagnostics server will not be included in LoadRunner Analysis after the Performance Center and LoadRunner runs. This includes LWMD, Heap Breakdown and the Summary view.
- ▶ Oracle 10g data will not be displayed in LoadRunner Analysis.
- ▶ System metric data will not be displayed in LoadRunner Analysis.
- ▶ Instance trees will not be available in LoadRunner Analysis, only aggregate trees are available after the needed drill down.

You can improve the transfer time and load time of the offline analysis files that include Diagnostics data by lowering the resolution of the .eve files.

Use the **bucket.lr.offline.duration** property and the **bucket.lr.offline.sr.duration** properties in the **server.properties** file on the Diagnostics server to increase the aggregation period.

Integration Between Diagnostics and Performance Center

Before you can use HP Diagnostics with Performance Center, you need to ensure that you have specified the Diagnostics Server details in Performance Center, according to the guidelines set out in the *HP Performance Center Administrator Guide*.

During a load test, you can drill down to HP Diagnostics data for the whole load test or for a particular transaction. HP Diagnostics opens, displaying the Synthetic Transactions view, which contains performance metrics and drill down options for the relevant script generated transaction. The Business Transaction view is not available when drilling down to Diagnostics from Performance Center.

For details about interpreting data in the Diagnostics Synthetic Transactions view, see "Synthetic Transactions View Description" on page 607.

After you have run your load test, you can use LoadRunner Analysis to analyze offline Diagnostics data generated during the load test.

Note: For more information about configuring Diagnostics to work with a firewall see the *HP Diagnostics Installation and Configuration Guide*. For more information about configuring Performance Center to work with Diagnostics in a firewall environment see the *HP Performance Center Documentation Library*.

For information on viewing Diagnostics data in Performance Center see the *HP Performance Center Documentation Library*.

Index

Symbols

- .NET Diagnostics profiler 1049
 - accessing 1055
 - enable and disable 1056
 - how the agent provides data 1051
 - inactivity timeout 1054
 - overview 1050
 - UI navigation and controls 1052
- .NET layers 505

Numerics

- 360 view in BSM 1110

A

- ABAP server requests
 - view description 756
- ABAP summary
 - view description 745
- active users
 - list of 117
- add all connected probes to application 56
- add or remove buttons 86
- add probe contents to application 57
- add to application 121
- aggregate requests
 - overview 530
 - view description 538
- aggregate trees 272
- aggregate views 529
 - overview 530
- alert
 - entity status change 180
 - metric status change 180
- alert events view 199
- alert events view description 197
- alert logs 184
- alert notification rules 181
- alert notifications
 - email 193
 - turn on or off 182
- alert properties 182
 - accessing 186
- alert properties page 181
- alert rule
 - active 110
 - create for entity 190
 - create on metric 190
 - disabled 110
 - entity status 183
 - metric status 184
- alert rules 183
- alert rules view 196
- alert rules view description 194
- alert triggers 193
- alerting enabled 187
- alerts
 - working with 179
- alerts displayed in the message box 81
- Alias 155
- alias
 - use to change Web service name 669
- all methods tab
 - Java Diagnostics profiler 1012
- all SQL tab
 - Java Diagnostics profiler 1015
- allocation capture
 - enabling for Java 993
- allocation lifecycle analysis tab
 - Java Diagnostics profiler 1032
- alert on metric 157
- analysis mode 76
- analyze snapshots 203

Index

- Analyze Snapshots view
 - accessing 212
- analyze snapshots view description 210
- application
 - create a new group 121
- application explorer
 - drill down and analysis 319
 - metrics available by entity 324
 - overview 316
 - view description 326
- application explorer topology 318, 329
- application group
 - about 46
 - creating new 53
- application identifier 218
- application infrastructure HI status 1100
- application metrics
 - drill down and analysis 339
 - overview 338
 - view description 341
- application metrics view 337
- application server
 - specify a name 667
- application summary 282
- applications
 - adding an entity to 58
 - App column 52
 - create or edit 54
 - delete 58
 - editing properties 57
 - overview 46
 - permissions 63
 - remove or exclude from 60
 - removing an entity from 60, 122
 - select 61
 - selecting pre-defined 66
- applications window 45, 65
 - description 64
 - returning to 49
- apply recommended threshold 161
- AquaLogic Service Bus
 - monitoring 669
- area chart 99
- argument capture 418
- asynchronous server requests
 - monitoring 673

- asynchronous thread
 - stack trace sampling 411
- asynchronous thread sampling 976
- auto hide button 94
- availability in BSMC
 - SOAP faults excluded 1120

B

- bac.webservice.create.samples 664
- backend transaction data 577
- BEA AquaLogic Service Bus
 - monitoring 669
- BEA WebLogic summary
 - view description 842
- BEA WebLogic views 839
 - overview 840
- BEA WLI business process tracing data 690
- biztxn.discovery.frequency 588
- bizTxnDiscoveryRules.properties 589
- BPM application or profile 46
- BPM related views
 - BSM and Diagnostics integration 1115
- breadcrumb 82
- breakdown by portal components
 - view description 401
- breakdown by server requests 623
 - view description 645
- BSM
 - business transactions views 1110
 - configurable Diagnostics information 1118
 - data from Diagnostics 1090
 - data samples from Diagnostics 1097
 - Diagnostics probe group and infrastructure view 1108
 - general drill downs to Diagnostics 1112
 - open Diagnostics UI from 1106
 - SOAP faults excluded from availability 1120
- BSM concepts for Diagnostics users 1088
- BSM integration with Diagnostics
 - overview 1086
- BTP 579

- examples 579
 - business process summary
 - overview 615
 - view description 300
 - business process summary with portlets
 - view description 634
 - business transaction
 - configure discovery 587
 - create or edit 583
 - definition 576
 - business transaction HI status 1098
 - business transaction monitoring 578
 - business transaction path
 - latency calculation 582
 - business transaction paths 579
 - business transaction views in BSM 1110
 - business transactions
 - view description 591
 - business transactions paths
 - view description 597
- C**
- call profile
 - analysis 422
 - call tree table 408
 - drill down 414
 - graph 407
 - how to use in analyzing performance 969
 - interesting instances 258
 - overview 406
 - SOA services 656
 - SOAP faults 419
 - view description 427
 - call profile graph 972
 - call profile view 405
 - call profiles
 - portlets 618
 - call stack
 - monitoring 966
 - call stack trace sampling 976
 - call tree tab
 - .NET Diagnostics profiler 1067
 - call tree table 974
 - categories
 - details pane 152
 - Category Name 155
 - chart
 - multiple graph layout options 101
 - chart column 102
 - charts 97
 - CI 1088
 - CI population and models 1091
 - CI types with HIs and KPIs
 - table of 1098, 1100
 - CICS summary
 - view description 917
 - CICS views 915
 - overview 916
 - close this view group 222
 - code resources
 - view description 470
 - code resources view 459
 - collapse all categories 153
 - collection leak pinpointing (CLP)
 - comparison to LWMD 978
 - collection leaks
 - view description 453
 - collection leaks tab
 - Java Diagnostics profiler 1016
 - collections
 - view description 465
 - collections and resources
 - drill down and analysis 463
 - overview 460
 - collections tab
 - .NET Diagnostics profiler 1075
 - Java Diagnostics profiler 1020
 - collections view 459
 - color 109
 - status indicator 110
 - coloring 1089
 - event based and metric based 1097
 - comments 123
 - common data model 1093
 - common tasks and navigations 95
 - common tasks pane 117
 - composite application 46
 - composite application explorer 316
 - configuration tab
 - Java Diagnostics profiler 1042

Index

- connection arrows in topology 248
- connections in topology view 352
- consumer connections in topology 719
- consumer connections in topology view 353
- consumer icon in topology 248
- consumer ID
 - capture increases data sent to BAC 664, 727
 - configuring 663
- consumer IDs 663
- consumers
 - configuring 1118
- consumers in topology view 352
- Contents summary 67
- contents summary 47
- CPU 168
- cpu time
 - display for server requests 381
- CPU time metrics 168
- cpu.timestamp.collection.method 170
- cputime element 169
- cputime for portlets
 - configure 616
- create a new view 224
- create a view group 221
- create and edit alert rules 123
- create and edit comments 123
- create new snapshot and add 119
- cross VM for services 659
- cross VM instance tree 260
- cross VM instance trees 265
- cross VM tree 258
- CSV
 - save page as 82
- custom attributes in the details pane 154
- custom dashboard 131
- custom layers 496
- custom threshold 161
- custom time range 79
- custom view
 - create 223
 - delete 230
 - rename 229
 - save 227
 - sharing 217
 - upgrade 219

- custom view group
 - create 221
- custom views 216
 - adding summary views 226
 - applications 52
- customize columns 142

D

- dashboard layout
 - example 88
- data export from TSDB 131
- data samples 1097
- data_integration_uom.xml 949
- database names in JDBC connections 355
- delete probe 124
- delete view 230
- dependent services
 - overview 531
 - view description 544
- detail layout
 - example 87
- detail view, about 95
- detail views
 - common features 95
- details pane 96, 149
 - alert rules 157
 - custom attributes 154
 - example 151
 - find 154
 - metric categories 152
 - metric in snapshot 157
 - thresholds 158
- Diagnostics and LoadRunner integration 1122
- Diagnostics and SiteScope integration 1121
- Diagnostics data provided to BSM 1090
- Diagnostics integration with offline analysis 1122
- Diagnostics probe group and infrastructure view in BSM 1108
- Diagnostics profiler for .NET 1049
- Diagnostics profiler for Java 963
- Diagnostics UI
 - using 71
- diagram legend 251, 252

- diagram overview 251
- disable alert rule 193
- discovery policies 56
- discovery process 69
- docking framework 92
- docking windows 92
- drill down to other Diagnostics views 125
- drill downs from BSM 1112
- drop view here 226
- dynamic instrumentation 976
- dynamic threshold 160

E

- edit screens 56
- EJB 3.0 business method 386
- EJB pooled resource contention
 - view description 848
- email alert notification 188
- endpoint 667
- enterprise service bus
 - about 666
- Entire Enterprise 46
- entities in applications 49
- entity-metric pairs 207
- ESBs
 - about 666
- EUM
 - integration with Diagnostics 1114
- event based HI status 1089, 1100
- exception data
 - configuring collection of 269, 409
- exception data in call profile 266
- exception icon 260, 659
- exception metrics tab
 - application explorer 331
- exceptions
 - find 416
- exceptions tab
 - .NET Diagnostics profiler 1071
 - Java Diagnostics profiler 1024
- execute permissions 63
- export data from the database 131
- export view data 128
- external monitors views 947
 - overview 948

F

- filters
 - view 97
- filters in topology 351
- find 154
- find metrics 154
- fit data to size 80
- forwarded events not monitored 386

G

- garbage collection 969
- graph entity table 95
 - about 108
- graphs
 - about 97
- group probes 253

H

- heap breakdown
 - enabling for Java 996
- heap breakdown tab
 - Java Diagnostics profiler 1036
- heap walker analysis steps 984
- heap walker performance 989
- help 81
- help on view 81
- HI 1089
- HI status for application infrastructure CIs
 - 1100
- HI status for web service and business
 - transaction CIs 1098
- hidden columns 142
- hide metrics 153
- hierarchical topology layout 254
- host table 525
- hosts
 - drill down and analysis 372
 - overview 370
 - view description 373
- hosts in topology view 354
- hosts view 369
- hotspots tab
 - Java Diagnostics profiler 1002
- HP Software Support Web site 20, 21

Index

HP Software Web site 21

HTML

save page as 82

I

IBM WebSphere Enterprise Service Bus
monitoring 680

IBM WebSphere summary
view description 871

IBM WebSphere views 869
overview 870

inactivitytimeout 1054

inspect selected tracked objects 987

instance name

application server 667

instance tree

examples solving performance
problems 261

instance tree drill down to call profiles 260

instance tree marker icons 259, 383, 619,
658

instance trees 258

portlets 618

SOA services 656

instrumenting sampled methods 976

integration

Application Management for SOA
1117

Diagnostics and EUM 1114

LoadRunner 1122

LoadRunner offline analysis 1122

overview 1085

Performance Center 1124

SiteScope 1121

TM and Diagnostics 1115

integration with other HP software products
1083

J

J2C connection pool

view description 922

J2C connections and faults

view description 927

J2C load

view description 932

J2C usage time

view description 936

J2C wait time

view description 941

Java Diagnostics Profiler

garbage collection 969

metrics, refreshing 968

metrics, resetting 969

Java Diagnostics profiler 963

accessing 991

how the agent provides data 966

overview 965

UI navigation and controls 968

java Diagnostics profiler

set cpu timestamps 171

Java layers 503

java probe memory usage custom view

description 237

Java probes

view description 445

JDBC connection status

view description 853, 878

JDBC resource contention

view description 858, 883

jdbc.url.length 355

JMS calls in topology 349

JMS producer and consumer on cross VM
instance 657

K

Knowledge Base 20

KPI 1089

L

latency trimming 1120

REST services 665

layer details 513

layers

.NET 505

about instrumentation 496

defaults for Java and .NET 502

drill down and analysis 497

overview 496

- portal 505
 - view description 498
 - layers view 495
 - layers view for portlets 624
 - layout
 - multiple graph 101
 - layout types 254
 - life cycle methods 621
 - view description 642
 - lifecycle monitoring 980
 - load
 - calculating 508
 - drill down and analysis 510
 - overview 507, 508
 - view description 511
 - load view 507
 - LoadRunner
 - viewing Diagnostics data in 1122
 - LWMD 461
 - comparison to collection leak
 - pinpointing 978
 - enabling for Java 992
 - lwmd
 - enable for .NET 461
 - enable for Java 461
- M**
- maximize and restore 93
 - memory
 - monitoring 1052
 - memory analysis
 - enabling for Java 997
 - memory analysis steps 984
 - memory analysis tab
 - Java Diagnostics profiler 1038
 - memory monitoring 967
 - method latency
 - monitoring 966, 1051
 - methods
 - configuring trending 438, 536
 - methods tab
 - .NET Diagnostics profiler 1065
 - metric alert 157
 - metric based HI and KPI status 1089
 - metric charting 156
 - metric filter 153
 - metric history 162
 - metric in snapshot analysis 157
 - metric threshold 158
 - metrics
 - configuring 168
 - metrics in application explorer 324
 - metrics tab
 - Java Diagnostics profiler 1004
 - minimum.fragment.latency 665
 - model
 - example 1095, 1096
 - model population 1091
 - monitoring mode 76, 203
 - monitors
 - SiteScope integration 949
 - view description 954
 - monitors table 526
 - monitors view
 - how to display a graph and see status 952
 - MQ channels
 - view description 823
 - MQ connection stats
 - view description 888
 - MQ message driven bean stats
 - view description 893
 - MQ queue managers
 - view description 833
 - MQ queue stats
 - view description 898
 - MQ queues
 - view description 828
 - MQ views 821
 - overview 822
 - multiple entity select
 - add or remove from application 59
 - for alert rules 123, 191
 - for comments 124
 - multi-row select in tables 147
 - my views 216
- N**
- navigations 95
 - navigations pane 117

Index

- NET Diagnostics Profiler
 - refreshing metrics 1052
 - resetting metrics 1053
- NET Diagnostics profiler
 - taking snapshots 1053
- NetWeaver requests
 - view description 761
- NetWeaver summary
 - view description 749
- NetWeaver threads
 - view description 766
- non-Java connections in topology 353

O

- object allocation
 - how to analyze 995
- object lifecycle
 - allocation analysis comparison 983
- object lifecycle monitoring 980
 - enabling for Java 994
- object reference graph
 - example 987
- offline analysis
 - integration with Diagnostics 1122
- OM agent and IAPA components 1101
- online resources 20
- open a view group 222
- open in custom view 120, 219
- operations
 - view description 709
- operations by consumer ID
 - view description 726
- Oracle database summary
 - view description 780
- Oracle database views 777
 - overview 778
- Oracle probe summary
 - view description 795
- Oracle probes
 - view description 785
- outbound calls
 - drill down and analysis 478
 - overview 476
 - types of 476
 - view description 479

- outbound calls view 475
- outbound operations calls
 - view description 737
- outbound service calls
 - view description 733

P

- panes
 - described 95
- panning 249
- parameter aggregation enabled 1121
- parent entities 50
- path 55
- path topology
 - view description 602
- pause, pan, zoom controls 77
- payload on SOAP faults
 - capture 410
- Performance Center
 - viewing Diagnostics data in 1124
- Performance Center integration with Diagnostics
 - Diagnostics integration with Performance Center 1124
- performance metrics tab
 - application explorer 333
- permissions
 - application 63
- persisting custom layout changes 216
- pinning windows 94
- populate the model 1088
- portal components
 - overview 616
 - view description 636
- portal layers 505
- portal server summary
 - overview 614
 - view description 626
- portals view group
 - drill down and analysis 621
- portals views 613
 - overview 614
- post parameters 1121
- probe connections in topology view 353
- probe CPU utilization metrics 167

- probe data continues to be reported 434, 441, 446
- probe details custom view description 232
- probe group summary
 - view description 303
- probe group table 523
- probe grouping in topology diagram 356
- probe groups in topology view 354
- probe rename
 - data under old name 124
- probe summary
 - view description 309
- probe table 524
- probe topology 348
- probes
 - drill down and analysis 436
 - overview 434
 - view description 440
- probes in topology view 352
- probes view 433
- profiler and LR or PC
 - requires login 1055
- profiler for .NET 1049
- profiler for Java 963
- proxy services 666
- Pseudo 312, 380, 392, 399, 542
- purge probe
 - no data received for 6 months 124

Q

- query
 - view thresholds 162
- query load
 - UI limits on 116

R

- redirection requests not monitored 386
- refresh 81
- refresh metrics 968
- refresh now 1052
- relationship rule 50
- relationships 1088
- relevant tasks and navigations 119
- remote access disabled

- profiler 1055
- remove from chart 99
- rename view 229
- reply to queue name 657
- reset screen layout 82, 93
- resize windows 92
- resource utilization metrics tab
 - application explorer 335
- resources
 - view description 470
- resources view 459
- REST style services 665
- rest.properties file 665
- RootRename 312, 380, 392, 399, 542
- rule based transaction discovery 587
- RUM integration disabled 1120
- RUM related views
 - BSM and Diagnostics integration 1115
- run garbage collection 986
- Run-time Service Model 1088, 1093

S

- samples sent to BSM 1097
- sampling
 - thread call stack 411
 - thread call stack trace 976
- sampling example
 - in call profile 411
- SAP ABAP probes
 - view description 770
- SAP probe summary
 - view description 774
- SAP view group 564
- SAP views 743
 - overview 744
- save this view 82, 228
- save this view as an HTML web page 256
- save to CSV 129
- save to HTML 128
- scenario summary
 - overview 615
 - view description 297
- scenario summary with portlets
 - view description 630

Index

- search for entities in tables 146
- server request not found on drill down 1120
- server request topology
 - view description 396
- server request types 380
- server requests
 - delete 388
 - drill down and analysis 385
 - instance trees and call profile 382
 - overview 380
 - view description 389
- server requests not found on drill down 1117
- server requests tab
 - .NET Diagnostics profiler 1058
 - Java Diagnostics profiler 1026
- server requests view 379
- server summary
 - drill down and analysis 285
 - overview 282
 - view description 289
- server summary view 281
- server threads
 - view description 863, 903
- service calls
 - view description 549
- service connections in topology 719
- service model 1093
- service summary
 - view description 701
- service topology
 - view description 715
- services
 - cross VM instances 660
 - view description 704
- services by consumer ID
 - view description 721
- services monitored by Diagnostics 654
- servlet session management
 - view description 908
- set threshold dialog box 173
- shared views 216
- show metrics 153
- show more buttons 86
- SiteScope integration 1121
 - external monitors 949
- snapshot
 - add an entity 207
 - add entity 208
 - add metric 207
 - create new 205
 - delete 206
 - delete a metric 208
 - delete an entity 209
 - notes 214
 - remove from active 209
 - rename 206
- snapshot analysis
 - about 202
- snapshot analysis mode 203
- snapshot profiler session 1052, 1053
- SNMP alert notification 188
- SOA Policy Enforcer broker
 - monitoring 675
- SOA related views
 - BSM and Diagnostics integration 1117
- SOA services
 - about monitoring 653
 - overview 655
- SOA services view group 655
- SOA services views 651
- SOAP fault data in call profile 269
- SOAP fault details 420
- SOAP fault types 271
- SOAP faults
 - configuring capture 270
 - configuring collection of 409
- SOAP faults for services 661
- SOAP faults icon 260, 659
- SOAP message handler 270
- SOAP payload 270, 410
 - in call profile 420

sort

- table columns 144

specialized views 86

spring topology layout 254

SQL Server database summary

- view description 802

SQL Server database views

- overview 800

SQL server database views 799

SQL Server probe summary

- view description 817
 - SQL Server probes
 - view description 808
 - sql statements
 - overview 486, 532
 - view description 554
 - sql statements by probe
 - view description 489
 - sql statements view 485
 - SQL tab
 - .NET Diagnostics profiler 1062
 - sql trending
 - setting thresholds 534
 - sql trending threshold setting 487
 - stack trace details 419
 - stack trace on exceptions 430
 - stack trace sampling 411
 - standard detail views
 - server summary view 281
 - standard views 83, 86
 - start tracking new objects 984
 - status
 - view description 520
 - status in topology view 354
 - status indicators 110
 - description 517
 - status layout
 - example 89
 - status propagation 517
 - status tab
 - application explorer 330
 - status tab in application explorer 319, 331
 - status view 515
 - overview 516
 - stored custom view XML 218
 - summary tab
 - Java Diagnostics profiler 1000
 - synthetic transaction
 - definition 575
 - synthetic transactions
 - view description 607
- T**
- table
 - sort 144
 - table columns
 - customize 141
 - table header controls 114
 - target icon in topology 248
 - targets in topology view 354
 - temporary queues 351
 - thread sampling 411, 976
 - threads tab
 - Java Diagnostics profiler 1007
 - threshold
 - display 99
 - threshold for metric that triggered alert 199
 - threshold setting 158
 - threshold violation severity 110
 - threshold violations 165
 - threshold violations average and totals 166
 - thresholds
 - monitoring applications 163
 - server.log 163
 - setting sql statements 534
 - thresholds.configuration 160
 - thresholds.configuration file 160
 - TIBCO ActiveMatrix BusinessWorks
 - enable Apache-HttpClient-Outbound point 690
 - monitoring 689
 - TIBCO ActiveMatrix Service Bus
 - monitoring 684
 - time filter 78
 - time measurements in Diagnostics 35
 - timestamping 168
 - timestamps
 - cpu collections for portlets 618
 - title and filters
 - view 97
 - TM integration with Diagnostics 1115
 - tooltips 112
 - view details for data 106
 - tooltips on 81
 - top 5 100
 - top level server request 576
 - topn 117
 - topology
 - about 244
 - application explorer 318, 329
 - diagram overview 351

Index

- how to generate 350
- save as HTML page 256
- view description 359
- topology diagram 247
- topology layout
 - description 245
 - example 90
- topology toolbar 249
- topology view 347
- trace
 - instance trees 258
- transaction monitoring 578
- transaction types 575
- transactions views 573
 - overview 574
- tree topology layout 254
- trended methods
 - configure trending 437
 - overview 532
 - view description 559
- trended methods by probe
 - view description 450
- trending
 - configuring methods for 437, 536, 560
- trending threshold
 - sql statements 487
- Troubleshooting and Knowledge Base 20
- TSDB 131
- type filter 132

U

- ui.properties 116
- uniform topology layout 254
- unknown database
 - target name 355
- use default threshold settings 160
- use.cpu.timestamps 170
- user or role 56
- users
 - list of 117

V

- view bar 83

- show more buttons 86
- view context 75, 203
- view groups
 - hiding or opening 222
- view in area chart 99
- view link 255
- view panes
 - described 95
- view permissions 63
- view profiler for an entity 126
- view title and filters 97
- viewing time filter 78
- views
 - selecting 84
- violating metric 199
- virtual machine
 - host details 378
 - host topology 354
- VMware and CPU time metrics 169, 617

W

- wait time
 - view description 790, 813
- web service HI status 1098
- web service name
 - changing 668
- Web services
 - supported platforms 654
- web services
 - enterprise service bus environment 666
 - web services monitoring 653
- web services tab
 - Java Diagnostics profiler 1029
- WebLogic views 839
- WebSphere Enterprise Service Bus
 - monitoring 680
- WebSphere views 869
- Window 64
- workload metrics tab
 - application explorer 334

X

- x-axis 98

Y

y-axis 97
 alternate 98

Z

zoom 107
 reset 251
zoom contents to fit view 251
zoom in 138
zoom out 141
zooming 250

