



WebSphere Studio Application Monitor User's Guide

3.1



WebSphere Studio Application Monitor



WebSphere Studio Application Monitor User's Guide

3.1

Note:

Before using this information and the product it supports, read the information in “Notices” on page 181.

First Edition (November 2004)

This edition applies to WebSphere Studio Application Monitor (product number 5697-J18) and to all subsequent releases and modifications until otherwise indicated in new editions.

You can order publications through your IBM representative or the IBM branch office serving your locality.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Licensed Materials - Property of IBM.

WebSphere Studio Application Monitor (program number 5697-J18).

© Copyright, IBM Corp. 2004 All Rights Reserved.

US Government User Restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, and WebSphere are trademarks of IBM Corp. in the U.S., other countries, or both.

© Copyright International Business Machines Corporation 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Tables	xi
Preface	xiii
About this book	xiii
Who should read this book	xiii
Where to find more information	xiii
Documentation Conventions	xiii
Chapter 1. Getting Started	1
Product Overview	1
Administration	1
Account Management	1
Server Management	1
Monitoring on Demand™	2
Managing Server	2
Availability	2
Enterprise Overview	2
Group Overview	2
Server Overview	2
Web Server Overview	3
Portal Overview	3
Server Statistics Overview	3
Recent Activity Display	3
System Resources	3
System Resource Comparison	3
Workload Manager	3
Problem Determination	4
In-Flight Request Search	4
Server Activity Display (SAD)	4
Memory Diagnosis	4
JVM Thread Display	4
Software Consistency Check	4
Trap & Alert Management	5
Performance Management	5
Performance Analysis & Reporting	5
Daily Statistics	5
Help	5
Glossary	5
About	5
Chapter 2. Account Management	7
Purpose	7
Usage Overview	7
User Scenarios	7
Notes	7
User Profiles	7
Creating a UNIX Account	7
Creating a User Account	8
Modifying a User Account	8
Deleting a User Account	9
Role Configuration	9
Creating a Role	10
Duplicating a Role	10
Modifying a Role	10
Assigning a Role	11
Deleting a Role	11

Chapter 3. Server Management	13
Purpose	13
Usage Overview	13
User Scenarios	13
Notes	13
Server Groups	13
Creating a Group	14
Modifying a Group	15
Deleting a Group	15
Duplicating a Group	16
Data Collector Configuration	16
Configuring a Data Collector	17
Unconfiguring a Data Collector	17
Disabling a Data Collector	18
Enabling a Data Collector	18
Creating a Configuration	18
Applying a Configuration	19
Modifying a Configuration	20
Duplicating a Configuration	20
Deleting a Configuration	21
Web Server Administration	21
Adding a Web Server	21
Deleting a Web Server	21
Chapter 4. Managing Server	23
Purpose	23
Usage Overview	23
Notes	23
System Properties	23
Configuring the Data Collection Settings	23
Configuring the Enterprise Overview Display	24
Configuring the SNMP Network	25
Self-Diagnosis	26
Viewing the Self-Diagnosis for the Kernel	26
Viewing the Self-Diagnosis for the Archive Agent	27
Viewing the Self-Diagnosis for the Publish Server	27
Viewing the Self-Diagnosis for the Global Publish Server	27
Viewing the Self-Diagnosis for the Data Collector Controller	27
Viewing the Self-Diagnosis for the Message Dispatcher	28
Viewing the Self-Diagnosis for the Polling Agent	28
Chapter 5. Monitoring on Demand™	29
Purpose	29
Usage Overview	29
User Scenarios	29
Notes	29
Tracing Intent (z/OS Platform Only)	30
.....	30
Managing MOD	30
Selecting a Monitoring Level	31
Changing the System Default Setting	32
Creating a Schedule	33
Applying a Schedule	33
Overriding the Monitoring Level	34
Modifying a Schedule	34
Deleting a Schedule	35
Duplicating a Schedule	35
Chapter 6. Systems Overview	37
Purpose	37
Usage Overview	37
User Scenarios	37
Enterprise Overview	37
Group Overview	38
Server Overview	39

Web Server Overview	40
Web Server Details	41
Portal Overview	42
Portal Page Summary	43
Portlet Summary	43
WLM Associated Service Class Summary	44
WLM Associated Service Class Period Detail	45
WLM Enclave	46
Viewing Server Statistics Overview	47
Purpose	47
Configuring the Server Statistics Overview	50
Chapter 7. In-Flight Request Search	53
Purpose	53
Usage Overview	53
User Scenarios	53
Searching for an Application Request	53
Sorting Search Results	54
Chapter 8. Server Activity	55
Purpose	55
Usage Overview	55
User Scenarios	55
Server Activity Display	55
Active Requests	56
Recent Requests	57
Viewing Request Detail	58
Suspending a Thread	59
Activating a Thread	60
Canceling a Request	60
Changing a Thread's Priority	61
Viewing the Request Object and Session Object	61
Viewing a Stack Trace	62
Viewing a Method/Component Trace	63
About the Method/Component Trace	63
Using the Flow View	64
Searching a Method/Component Trace	67
Additional Method/Component Trace Features for Completed Requests	68
Nesting Summary	69
Drilldown View	69
Chapter 9. Recent Activity	71
Purpose	71
Usage Overview	71
User Scenarios	71
Notes	71
Creating a Recent Activity Report	71
Chapter 10. Memory Diagnosis	73
Purpose	73
Usage Overview	73
User Scenarios	73
Notes	73
Memory Analysis	73
Creating a Memory Analysis Report	73
Heap analysis	75
Setting up a Heap Analysis	75
Memory Leak	77
Creating a Memory Leak Confirmation report	77
Creating a Memory Leak Candidate Finder Report	78
Viewing a Memory Leak Candidate Finder Report	78
Chapter 11. JVM Thread Display	81
Purpose	81

Usage Overview	81
User Scenarios	81
Notes	81
Viewing the JVM Thread Display	81
Viewing an Active Thread	82
Change a Thread's Priority	83
Viewing a Stack Trace	83
Canceling a Thread	83
Thread Dump	84
Chapter 12. Software Consistency Check	87
Purpose	87
Usage Overview	87
User Scenarios	87
Notes	87
The Installed Binary Files	87
Setting up an Installed Binary Comparison	87
Viewing the Results of the Installed Binary Comparison	88
Running the Installed Binary Check	89
Viewing the Installed Binary Check Detail	89
The Runtime Environment	90
Running the Runtime Environment Comparison	90
Chapter 13. Trap & Alert Management	93
Purpose	93
Usage Overview	93
User Scenarios	93
Notes	93
Managing Traps and Alerts	94
Setting an Application Trap	96
Setting a Server Resource Trap	97
Setting Alert Actions and Data Actions	97
Activating a Trap	99
Deactivating a Trap	100
Modifying a Trap	101
Duplicating a Trap	101
Deleting a Trap	101
Viewing a Trap Action History	102
Chapter 14. System Resources	103
Purpose	103
Usage Overview	103
User Scenarios	103
Notes	104
Viewing the System Resources Overview - Non-z/OS	104
Viewing the System Resource Overview - z/OS	105
General	106
WebSphere	108
WebSphere - PMI	108
WebSphere - SMF	112
WebSphere on z/OS Only	115
Chapter 15. Daily Statistics	117
Purpose	117
Usage Overview	117
User Scenarios	117
Notes	117
Accessing the Daily Statistics	117
Viewing the Daily Statistics Overview	118
Chapter 16. System Resource Comparison	121
Purpose	121
Usage Overview	121
User Scenarios	121

Notes	121
Chapter 17. Performance Analysis & Reporting	123
Purpose	123
Usage Overview	123
User Scenarios	123
Create Reports	124
Creating a Request/Transaction Analysis Report	124
Creating a Method/Program Analysis Report	125
Creating a SQL Analysis Report	125
Creating an MQI Analysis Report	126
Creating a Top Report	127
Creating a System Resource Analysis Report	128
Creating a Server Availability Analysis Report	129
Creating a Capacity Analysis Report	129
Creating a Scheduled Report	130
View Saved Reports	131
Viewing the Reports	131
Running a Report	134
Modifying a Report	134
Deleting a Report	135
Emailing a Report/Link	135
Viewing a PDF File	136
Exporting to a File	136
Understanding the Date Range Settings	137
Chapter 18. Composite Requests	139
Purpose	139
Usage Overview	139
User Scenarios	139
Notes	139
Product Overview	139
WSAM and Composite Requests	139
Finding Composite Requests: WSAM	139
Viewing Composite Requests	140
The Scope of Composite Request	140
WSAM Architecture: The Context of the Managed Space	140
Defining the Composite Request Space	141
Multiple Hops	143
Configuring Data Collectors that use MQ	143
Finding Composite Requests	143
Identifying Composite Requests in WSAM	144
Using the Composite Request Indicator	144
Viewing Composite Requests	145
Composite Request Features	145
Viewing a Composite Method Trace	146
Viewing a Composite Stack Trace	147
Authorization and Composite Requests	148
Chapter 19. WebSphere Studio Workload Simulator Integration	151
Real-time Statistics Query	151
Auto-generated Benchmark Report	151
Chapter 20. Audit Trails	153
Purpose	153
Usage Overview	153
User Scenarios	153
Accessing the User Audit Trail	153
Chapter 21. Request Mapper	155
Purpose	155
Usage Overview	155
User Scenarios	155
Notes	155

Data Used by the Request Mapper	155
Request Name	155
Application Name	156
User IDs	156
Default Request Mapping Behavior	156
Writing and Deploying a Request Mapper	157
Package com.cyanea.mapper	157
Interface Mapped Request	158
Interface Request Mapper	158
Sample Request Mapper - mapRequest	159
Glossary	163
Notices	181
Trademarks	185
Index	187
Bibliography	191

Figures

1. Top Navigation	1	27. Method/Component Trace: Depth Drilldown Detail	
2. Create group	14	70	
3. Apply page	19	28. Recent Activity report	72
4. Data Collection Settings	24	29. Memory Analysis Report	74
5. Enterprise Overview Display	25	30. Heap Analysis results	76
6. SNMP Network Configuration	26	31. Memory Leak Confirmation report	77
7. Modify Server Settings	32	32. Memory Leak Candidate Finder Report	79
8. Enterprise Overview	38	33. JVM Thread Display	82
9. Group Overview page	39	34. Folder contents in the Matched folders	88
10. Server Overview page	40	35. Installed Binary Check	89
11. Web Server Overview page	41	36. Runtime Environment Check	91
12. Web Server Details page	41	37. Trap & Alert Management	95
13. Portal Overview	42	38. Set Trap Alerts	98
14. Portal Page Summary	43	39. Activate page	100
15. Portlet Summary	44	40. System Resources Overview	104
16. WLM Associated Service Class Summary page ..	45	41. System Resources Overview - SMF Data	105
17. WLM Associated Service Class Period Detail page ..	46	42. System Resources Overview - PMI Data	106
18. Server Statistics Overview	49	43. Daily Statistics Selection	118
19. Server Statistics Configuration	51	44. Daily Statistics	118
20. Server Activtiy Display (Active Requests)	57	45. System Resource Comparison	122
21. Server Activtiy Display (Recent Requests)	58	46. Top Report	128
22. Request Detail	59	47. Trend report	132
23. Request Object and Session Object	62	48. Decomposition report	133
24. Stack Trace	63	49. Date Range Settings	137
25. Method/Component Trace	65	50. The Composite Request Indicator	144
26. Method/Component Trace: Search	68	51. Composite Method Trace	146

Tables

1. Documentation Conventions	xiv	14. Session Manager data	112
2. CICS Transaction data	106	15. Thread Pool data	112
3. Queue Manager data	106	16. Database Connection Pool data	113
4. Queue data	107	17. EJB data	113
5. SQL data	107	18. Server data	114
6. WebSphere Breakdown	108	19. Servlet and Session Manager data	114
7. Database Connection Pool data	109	20. Web Applications data	115
8. EJB data	109	21. Server Regions' data	115
9. JCA Connection Pool data	110	22. Sample from User Audit Trail Log	154
10. JTA Transaction data	110	23. Interface Summary	157
11. JVM/System data	111	24. Method Summary	158
12. ORB Detail/Interceptor data	111	25. Method Summary	158
13. Web Application data	111	26. Trademarks	185

Preface

About this book

This book is the User's Guide for WebSphere® Studio Application Monitor (Application Monitor). It contains instructions and user information for the Application Monitor.

Who should read this book

Anyone who wants to learn more about how to use the Application Monitor.

Where to find more information

The following list shows the books in the Application Monitor library:

- *WebSphere Studio Application Monitor User's Guide* contains instructions and user information for the Application Monitor.
- *WebSphere Studio Application Monitor Operator's Guide* contains information about the operation of Application Monitor and the common services address space.
- *WebSphere Studio Application Monitor Installation and Customization Guide* contains instructions on installing user's exits and customizing the Application Monitor.
- *WebSphere Studio Application Monitor for CICS® Data Collector Product Guide* contains information about the installation, configuration and use of the Application Monitor CICS Data Collector.
- *WebSphere Studio Application Monitor for IMS Data Collector Product Guide* contains information about the installation, configuration and use of the Application Monitor IMS Data Collector.
- *WebSphere Studio Application Monitor Messages and Codes* contains information about messages and codes generated by the Application Monitor.
- *WebSphere Studio Application Monitor Program Directory* contains complete installation instructions for the Application Monitor Engine.
- *WebSphere Studio Application Monitor Program Directory for the CICS Data Collector* contains complete installation instructions for the Application Monitor CICS Data Collector Engine.
- *WebSphere Studio Application Monitor Program Directory for the IMS Data Collector* contains complete installation instructions for the Application Monitor IMS Data Collector Engine.
- *WebSphere Studio Application Monitor* has an online help system that describes all of the commands and dialogs available from its graphical user interface.

Documentation Conventions

The following table describes the conventions used throughout this book.

Table 1. Documentation Conventions

Convention	Item
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<installation directory>	Indicates that you should insert the directory appropriate for your environment.
monospace text	Indicates code samples.

Chapter 1. Getting Started

Welcome to WSAM — the system that monitors the status of transactions in your J2EE application server farm. WSAM also provides a complete history of performance and availability and a real-time Visualization Engine. You can use it to find the root cause of problems, troubleshoot them quickly, and enable capacity planning and sizing within a business context.

Product Overview

The Monitoring Console makes the Data Center Operator's job easier by allowing them to easily answer questions about systems and by troubleshooting applications. The following list describes WSAM's features based on its top-level navigation structure.



Figure 1. Top Navigation

Administration

The top-level navigation for Administration includes Account Management, Server Management, Monitoring on Demand™, and Managing Server.

Account Management

The Account Management section contains the User Profiles and Role Configuration sections.

Manage your user accounts in User Profiles. Add and delete user accounts as necessary. Role Configuration displays the system default roles and any custom roles, created by the administrator, specific to the needs of their data center environment. Manage the custom roles by maintaining and updating user account access.

Server Management

Server Management contains the Server Groups, Data Collector Configuration and Web Server Administration sections.

In Server Groups, manage the groups by creating, duplicating, and deleting groups as needed. Maintain existing groups by editing them when necessary.

In Data Collector Configuration, configure and unconfigure Data Collectors, maintain your Data Collectors' status and create configurations to apply to your Data Collectors. In addition, manage your configurations by creating, applying, modifying, duplicating, and deleting to keep them up-to-date.

The Web Server Administration section provides a method for adding and deleting Web servers from the Web Server Overview page.

Monitoring on Demand™

Monitoring on Demand™ provides three different types of monitoring levels for each group or server including: L1 (Production mode), L2 (Problem Determination mode), and L3 (Tracing mode).

Create a schedule to apply to a server or group of servers.

Managing Server

The Managing Server contains the System Properties and Self-Diagnosis sections. In System Properties, maintain the system settings for WSAM. Also control the settings for the following properties: Data Collection Settings, Enterprise Overview Display and SNMP Network.

The Self-Diagnosis allows you to view all WSAM components and their states and attributes. The Managing Server consists of the following components: Kernel, Data Collector Controller, Publish Server, Global Publish Server, Polling Agents, Archive Agent and Message Dispatcher. WSAM is designed to work as a loosely-coupled system, so the components can be up or down without affecting the integrity of the whole system.

Availability

The top-level navigation for Availability includes the Systems Overview and the Server Statistics Overview.

The System Overview section includes Enterprise Overview, Group Overview, Server Overview, Web Server Overview, Server Statistics Overview and Portal Overview.

Enterprise Overview

The Enterprise Overview page displays information for all groups of servers. It provides the highest level view of health status for your Data Center. Additional data displayed on the page includes completed requests for the group. Links are available for each of the groups to further investigate the availability, to compare resource use, and to search the group information for a request.

Group Overview

The Group Overview page provides a high-level overview of activity for each server in the group. Specifically, the overview includes the response time and throughput for the last hour as well as the current monitoring level for each server. You can analyze this data in order to ascertain whether the servers in the group are functioning properly.

Server Overview

The Server Overview page displays comprehensive server information, activity, statistics, and resource data for the selected server. View the summary data to understand the status of your applications and application server behavior. This page provides vital information for determining the health of a server.

Web Server Overview

The Web Server Overview page shows whether your Web servers are properly functioning. While performing problem determination functions, it is useful to know the status of your Web servers. You can efficiently eliminate your Web servers as the source of the problems by checking the Web Server Overview page.

Portal Overview

The Portal Overview page shows you the portals in your system and how they are operating. You can monitor the status of your portals from the slowest portals to the portals with the highest throughput for the last hour. In addition, view the metrics for the portals including Average Response Time and Count for authentication and authorization, as well as credential and content access metrics.

Server Statistics Overview

The Server Statistics Overview page provides application server-level statistics for quick assessment of server activity and related platform data. This page assists you in drawing an educated guess of the true availability of the applications being served in the individual application servers. (This is also called activity- based availability as opposed to IP availability.)

Recent Activity Display

Recent Activity helps you investigate potential memory problems relating to garbage collection and the JVM heap size. At times garbage collection may not clean up properly or the heap may have too little memory allocated. Use Recent Activity to create a server activity analysis report.

System Resources

The System Resources page displays summary information for all a resources on the selected application server. WSAM captures the data for this page every 5 minutes for display. From the System Resources Overview, you can use the left navigation to switch the view to data on EJBs, JCA Connection Pools, SQL, MQI, Database Connection Pools, JVM/System, ORB, JTA Transactions, Session Manager, Servlet/Session Manager, Thread Pools, Transactions, Connectors, Execute Queue, Server, JMS, JCA and Web Applications.

System Resource Comparison

Compare a selected resource on all servers in a group by using the System Resource Comparison. Decisions regarding taking servers off-line that are under utilized, or adding servers to a group that are at maximum capacity, can be made using this comparison.

Workload Manager

The Workload Manager feature offers a way to view selected data from the Workload Manager (WLM) for z/OS® and OS/390®, for the address space associated with a particular server, as well as its associated service class data, service class period and enclave data. This feature is only available for z/OS servers.

The Workload Manager feature is not available directly from the top-level navigation. It is available through the Tools button on the Server Overview (within the Systems Overview feature) and the Server Statistics Overview, for z/OS servers.

Problem Determination

The top-level navigation for Problem Determination includes the In-Flight Request Search, Server Activity Display, Recent Activity Display, Memory Diagnosis, JVM Thread Display, Software Consistency Check and Trap & Alert Management.

In-Flight Request Search

The In-Flight Request Search page lets you search for a request on your application servers. To search for a request, enter in the request using alpha numeric characters or a URL string or leave it blank to search for everything. You may also view the stack trace, component trace, or method trace for a particular request. View, email or export the PDF file of the trace to other WSAM users.

Server Activity Display (SAD)

The Server Activity Display page provides thread data for an application server at a specific point in time, as well as the 100 most recently completed requests. You may filter the threads by the type or thread status. This limits the list to the type of threads you want to view. After pinpointing a hung thread, click the Thread ID link to review its request detail. Click links to view the stack trace, component trace or method trace. View, email or export the PDF file of the trace to other WSAM users.

Memory Diagnosis

The Memory Diagnosis section helps you discover memory related problems. Memory Analysis lets you create server activity analysis reports regarding memory. Heap Analysis captures the runtime heap of an application server and breaks it down by the class names of the objects residing in the heap at the time of the snapshot while providing the number of instances and the size of the information. Lastly, Memory Leak helps confirm the existence of a memory leak and identifies the most likely memory leak candidates.

JVM Thread Display

The JVM Thread Display shows all the threads running on the JVM, as organized within their thread groups. In addition, the JVM Thread Display provides a Thread Dump so you can view detailed information about resource consumption in a JVM. In addition, you can click on a thread to view the details for the thread, or to view a stack trace, change the thread priority, or cancel a thread.

Software Consistency Check

View runtime environment and installed binaries information for your entire server farm through the Software Consistency Check. Perform a check on a selected server or compare one properly functioning server to up to 10 other servers in the farm. Use these functions to locate files that have not been updated or do not match.

Trap & Alert Management

Set software traps and alerts to monitor a group of servers or a selected server. By setting traps and alerts, notifications are sent immediately when the system meets the conditions of the trap. Actions include sending an email or an SNMP message, collecting Stack Trace, Component Trace, Method Trace, or Thread Dump. View the Trap History on the Trap Action History page of a trap that met the set conditions.

Performance Management

The top-level navigation for Performance Management includes the System Resources, System Resource Comparison, Performance Analysis & Reporting and Daily Statistics sections.

Performance Analysis & Reporting

Analyze application and application server data using Performance Analysis & Reporting. Create reports for a group of servers or a selected server. Analyze data for requests, methods, SQL calls, server availability and system resources. Decompose reports by server, request type or application. You may email or view a PDF file of a report, or export a file to a comma-delimited file format.

Daily Statistics

Daily Statistics provide daily information snapshots for z/OS WebSphere servers only. WSAM gathers the day's SMF data for all servers running z/OS WebSphere instances every night at midnight.

Help

Find answers to your questions using WSAM online Help. You can use the Contents tab to browse through the available Help topics; Index tab for an alphabetical listing of all our help text; and Search tab to find the answer to a specific question.

Glossary

Our easy-to-use glossary offers definitions for headings in results tables, processes, and unfamiliar terms used in the WSAM application. Search alphabetically to find definitions.

About

About provides the current version number for WSAM and trademark information, regarding pending and approved trademarks for WSAM and International Business Machines Corporation.

Chapter 2. Account Management

Purpose

Account Management enables you to control users' access to features and servers. Use roles to restrict access to features, and use server groups to grant access to servers.

Usage Overview

This feature helps you:

- Grant access to WSAM by creating new user accounts.
- Control access to servers by associating server groups with user accounts.
- Restrict access to features by assigning an appropriate role to each user account.

User Scenarios

Scenario 1: Granting members of Team XYZ access to WSAM

Team XYZ has asked for access to WSAM, but only needs access to features that use historical data. Since the existing roles provide access to features that use both real time and historical data, create a new role for them called team_XYZ. When you define this role, provide access to features that use only historical data, for example PAR. Assign role team_XYZ to each user account belonging to members of team XYZ.

Scenario 2: Creating an account for a new employee

Employee John Smith is an operator that just joined your company. John will need to use WSAM to monitor QA systems. As the WSAM administrator, you create John's account with access granted to QA server groups but not Production server groups. Furthermore, you restrict John's access to features by assigning the Operator role to his account.

Notes

Note: The user name can be different from the UNIX user name, but it must be at least 6 alpha characters and no more than 50. Multiple WSAM user accounts may use the same UNIX account. Also, multiple concurrent logins under the same WSAM user account are allowed.

Note: WSAM authenticates these accounts against the authentication mechanism used by the Managing Server's operating system. This means password maintenance is performed outside of WSAM.

User Profiles

The following instructions indicate how to manage user accounts within WSAM.

Creating a UNIX[®] Account

In order for the system to function properly, you must create a local user account on the Linux server that runs WSAM. Set up the UNIX user account prior to creating

your accounts, since it uses UNIX user names and passwords for authentication. The administrator creates a UNIX account using the following default method, unless your administration programs overwrote the default settings with an alternative method.

You can have multiple WSAM accounts sharing the same UNIX user ID and password. The system does not prohibit such usage.

To create a UNIX account:

1. Login as the UNIX Root user.
 2. Type in the UNIX command,
[useradd newusername]
 3. Press **Enter**.
 4. Type in the UNIX command,
[passwd newpassword]
 5. Press **Enter**.
- UNIX creates a new user account.

Creating a User Account

Add new user accounts to WSAM on the Create User Account page. Limit the rights of your user accounts to the groups of servers you select. All user accounts must have an existing UNIX user name in order to authenticate.

To create a user account:

1. From the top navigation, click **Administration > Account Management > User Profiles**.
The User Profiles page opens.
2. On the left navigation, click **Create User Account**.
The Create User Account page opens.
3. Enter the First Name.
4. Enter the Last Name.
5. Enter the User Name.
6. Enter the UNIX User Name.
7. Select the role you want to assign to the user account from the drop-down menu.
8. Select Active or Suspend for the Account Status.
9. Enter the user's Email Address.
10. Enter Remarks in the available fields.
11. Click to select the Group name in the All Groups box.
12. Click **Add** to grant the user account rights to the selected groups.
13. To save the user account setup, click **Save**.

Note: A user account is not ready for use if its status is not marked **Active**.

Modifying a User Account

Modify existing user accounts in WSAM on the Modify User Account page. Limit the rights of your user accounts to the groups you select.

To modify a user account:

1. From the top navigation, click **Administration > Account Management > User Profiles**.

The User Profiles page opens.

2. Click the user name to select the user account you want to modify.

The Modify User Account page opens.

3. Select the field you want to edit, and enter the new information.

4. After entering your changes, click **Save**.

Note: You may want to suspend the user accounts when the operators are on leave. When they return, select Active to turn their user accounts back on.

Deleting a User Account

Keep your system up-to-date by deleting old and unused WSAM user accounts. You can delete existing user accounts on the User Profiles page.

To delete a user account:

1. From the top navigation, click **Administration > Account Management > User Profiles**.

The User Profiles page opens.

2. Click **X** or **Delete** on the last column of the user account that you want to delete from WSAM.

A confirmation box displays.

3. Click **OK** in the confirmation box to delete the user account, or click **Cancel** to return to the User Profiles page.

4. If you select **OK**, the system deletes the user account and the User Profiles page no longer displays the deleted account.

5. To sort by heading, click the heading you want to sort. Only underlined headings can be sorted. When the page refreshes, the results display sorted by the selected heading.

Role Configuration

In order to have thorough control over the user accounts accessibility to the product functions, each user account will be assigned a role that grants access to the specific product functions. A role maps to individual product functions based on the following four sections of the system: Administration, Availability, Problem Determination, and Performance Management. There are three system default roles created in the Role Configuration page, namely Administrator, Operator and User. These roles cannot be deleted. The administrator role has the permission to create custom roles to suit the needs of their specific environment.

After setting up the custom roles, the administrator assigns a role to each user account. For example, the administrator creates a custom role for the Trading application and then selects the operations that data center operators need to monitor the trading functions.

Creating a Role

The Create Role page provides the functionality to create a custom role for your environment. Design the custom role to restrict and grant privileges specific to the needs of your environment.

To create a role:

1. From the top navigation, click **Administration > Account Management > Role Configuration**.
The Role Configuration page opens.
2. On the left navigation, click **Create Role**.
The Create Role page opens.
3. Type in the name of the new role.
4. Click **OK**.
The new role displays on the Role Configuration page.
5. Click to select the features user accounts will access in WSAM.
6. Click **Save**.

Duplicating a Role

To easily customize a new role, you may duplicate a role that uses a similar set of permissions rather than checking or unchecking the boxes one by one repeatedly.

To duplicate a role:

1. From the top navigation, click **Administration > Account Management > Role Configuration**.
The Role Configuration page opens.
2. On the left navigation, click **Duplicate Role**.
The Duplicate Role page opens.
3. Select a role name for the duplicated role from the Role Name drop-down menu.
4. Enter a new name for the duplicated role.
5. Click **Save**.
The new duplicated role displays on the Role Configuration page.
6. Click to select the features user accounts will access in WSAM.
7. Click **Save**.

Note: The duplicated role does not have any users since its user-to-role relationship is not duplicated.

Modifying a Role

The Role Configuration page provides the functionality to modify your custom roles. Update and delete custom roles based on the needs of your environment.

To modify a role:

1. From the top navigation, click **Administration > Account Management > Role Configuration**.
The Role Configuration page opens.
2. Change the custom role privileges users will access in WSAM.

3. Click **Save**.

Note: The reset function returns the modified roles to their original state.

Assigning a Role

After creating a new role on the Role Configuration page, assign the role to user accounts on the Modify User Account page. You may also modify user accounts to assign appropriate privileges to them.

To assign a role:

1. From the top navigation, click **Administration > Account Management > User Profiles**.
The User Profiles page opens.
2. Click the user name that you want to assign a role.
The Modify User Account page opens.
3. On the Modify User Account page, from the Role drop-down menu, select the role to assign to the user account.
4. Click **Save**.

Note: You must have a role to use the application.

Deleting a Role

The Role Configuration page provides the functionality to delete your custom roles. Manage your custom roles based on the needs of your environment. In addition, you cannot delete a role while the system associates a user account with it.

To delete a role not assigned to a user account:

1. From the top navigation, click **Administration > Account Management > Role Configuration**.
The Role Configuration page opens.
2. Click the **X** next to the role you want to delete.
A confirmation box displays.
3. Click **OK** in the confirmation box to delete the user account, or click **Cancel** to return to the User Profiles page.

To delete a role still assigned to a user account:

1. From the top navigation, click **Administration > Account Management > Role Configuration**.
The Role Configuration page opens.
2. Click the **X** next to the role you want to delete.
A confirmation box displays.
3. Click **OK** at the confirmation box.
A list of the user accounts assigned to the role appears. Since the system assigned the role to a user account, you have to change the role of the user account on the Update Role page.
4. Click on the link to select the user account.
The Modify User Account page opens.
5. Click to select a role for the user account from the Role drop-down list.

6. Click **Save**.

The system displays the Role Configuration page without the deleted role.

Chapter 3. Server Management

Purpose

Use the Server Group Management page to add and delete server groups. Associate groups with individual accounts. Restrict users' access to data and operations to a specific group of servers.

Usage Overview

This feature helps you:

- Control access to servers by associating server groups with user accounts.
- Divide your servers into server groups according to lines of business, authority structure or based on your needs.

User Scenarios

Scenario 1: Separating server groups according to applications

As the WSAM administrator, you want to distinguish the group of servers that process trading requests from the group of servers that process quote requests. You create two server groups: Trading and Quotes. In the Trading server group, you include only those servers that deal with trading, and in the Quotes server group you include only those servers that deal with quotes. Grant users access to the appropriate server group(s).

Scenario 2: Grouping servers by authority structure

As the WSAM administrator you want to separate the servers in your environment by the authority structure present in the company. The current Support team is separated into smaller groups that control individual groups of servers. You create server groups that contain these servers such as Support A controls servers 1 through 29, Support B controls servers 30 through 59 and Support C controls servers 60 through 90.

Notes

Note: Only configured servers appear in the list of servers available to group, but servers do not have to be up and running to appear.

Note: While creating groups, use only alphanumeric characters in the name (except +, ', ", \, ~, *, # or SPACE). Group names are case-sensitive.

Note: In a z/OS environment, server instances are grouped, not the server regions. Server regions that belong to a server instance are automatically grouped under that server instance, and they are distinguished from a server instance by having the address space ID appended to the end of their server name.

Server Groups

The following instructions indicate how to manage the groups in WSAM.

Creating a Group

Combine servers into groups to streamline daily server maintenance. The Create group page provides the functionality to create groups of servers and grant users access to those groups.

To create a group:

1. From the top navigation, click **Administration > Server Management > Server Groups**.

The Server Group Management page opens.

2. On the left navigation, click **Create group**.

The Create group page opens.

The screenshot shows the 'Create Group' form with the following sections and fields:

- GROUP INFORMATION** (Required field):
 - *Group Name: [Text box]
 - Description: [Text box]
- ENTERPRISE OVERVIEW DISPLAY**:
 - BASELINE INDICATORS** - Enter a percentage up to 10 times (999%) for each Indicator's response.
 - Indicator 1 (Slow Response) >= [25] %
 - Indicator 2 (Very Slow Response) >= [50] %
 - BASELINE DEFINITIONS** - Select one baseline definition and fill out the appropriate information.
 - Rolling Date (1-31 days): [7] days
 - Fixed Date: Start Date [Jan] [01] [2002], End Date [Jan] [01] [2002]
 - Fixed Response Time (0-10,000 ms): [1000] (ms)
- GROUP MEMBERS**:
 - All Servers: [List of servers: ADCDPL.M1L2.P390.CICS2QA3, ADCDPL.M1L2.P390.CICS2QA4, ADCDPL.M1L2.P390.CICS2QA9, ADCDPL.M2L2.M2L2.CIC23QA1, ADCDPL.M2L2.M2L2.CIC23QA2, ADCDPL.M2L2.M2L2.CIC23QA3]
 - Servers in Group: [Empty list box]
 - Buttons: Add >, < Remove
- USER ACCESS**:
 - All Users: [List of users: Arun Srinivas, Calvin So, eddie park, meena nigam, Sarah Butts, ankem sriram]
 - Granted Access: [List containing: Cyanea Administrator]
 - Buttons: Add >, < Remove
- Buttons: Cancel, Save

Figure 2. Create group

3. Enter a unique Group Name in the text box. (Required field).
4. Enter a Description in the text box.
5. Enter the Baseline Indicator 1 in the box. (Percent value between 1% -999%.)
6. Enter the Baseline Indicator 2 in the box. (Percent value between 1% -999%.)
7. Select a baseline definition and fill out the information.

Note: Steps 5 through 7 are all default settings based on the settings on the System Properties page under Configuring the Enterprise Overview Display. For detailed information, see “System Properties” on page 23.

8. Click to select server name(s) in the All Servers box.
Note: To select multiple servers in a row, hold down the shift key during your selection. To add multiple servers non-continuously, Ctrl + click the servers for selection.
9. Click **Add** to place the servers for the Group.
The server name appears in the Servers In Group box.
10. In the Servers In Group box, select the server(s) you want to remove and click **Remove** to delete the server(s) from the group.
The server name(s) disappear from the Servers in Group box.
11. Select the user and click **Add** to grant users access to the group.
The user name(s) appear in the Granted Access box.
12. Click **Remove** to remove the user's access to the group.
The user name disappears from the Granted Access box.
13. Click **Save** to save the group's settings.

Modifying a Group

Maintain your groups with the most updated definition. The Modify Group page lets you modify your groups and grant users access to those groups.

To modify a group:

1. From the top navigation, click **Administration > Server Management > Server Groups**.
The Server Group Management page opens.
2. Click the Group Name of the group you want to modify.
The Modify Group page opens populated with the selected group's information.
3. Select the field you want to edit and enter the new information.
4. Click **Save** to save the group's settings.
Note: Changes made to the server-to-group assignments and user-to-group grants occur immediately. Also, if an administrator removes a server from a group, anyone logged in will notice the change.

Deleting a Group

Delete outdated groups from the system. You can delete existing groups on the Server Group Management page.

To delete a group:

1. From the top navigation, click **Administration > Server Management > Server Groups**.
The Server Group Management page opens.
2. Click **X** or **Delete** next to the group name you want to delete from WSAM.
3. Click **OK** in the confirmation box to delete the group, or click **Cancel** to return to the Server Group Management page.

4. If you select **OK**, the system deletes the group and the Server Group Management page no longer displays the deleted group.

Note: Once a group is deleted, the records in the WSAM database that belong to the group via the server relationship will no longer be accessed through the group. However, they can still be accessed either via the server name or another group which contains the servers.

Note: When you try to delete a group from the WSAM database, you will first be shown a list of all reports that involve that group, which you must delete before the group can be deleted. Click on the link of each report in the list and confirm that you want to delete it. When you delete all reports that involve the group, the group will be deleted.

Duplicating a Group

Save time by duplicating groups. Duplicating a group allows you to quickly create a new group based on the settings of an existing group.

To duplicate a group:

1. From the top navigation, click **Administration > Server Management > Server Groups**.

The Server Group Management page opens.

2. On the left navigation, click **Duplicate Group**.

The Duplicate Group page opens.

3. From the Group Name drop-down menu, select the group name you want to duplicate.

4. Enter a new name for the duplicated group.

5. Click **Save** to duplicate the group.

Note: The Duplicate Group link will not appear when there is no group in the system. The duplicated group does not have any users since its user-to-group relationship is not duplicated.

Data Collector Configuration

The Data Collector section provides lists of configured and unconfigured Data Collectors. A Data Collector is software that runs within the same JVM as the application server and captures information regarding the applications running inside the application server. At times, it may be necessary to unconfigure Data Collectors on the application server or configure new Data Collectors.

There are three different levels of monitoring available for the Data Collector:

- L1 (Production mode) provides Availability Management, System Resources and basic request data.
- L2 (Problem Determination mode) monitors production level plus advanced request data, including CPU information, additional monitoring fields and functions.
- L3 (Tracing mode) monitors everything in L2 plus method and SQL-call level operations. Trap and Alert functions that are based on L3 events require that you set Data Collectors on L3.

When a Data Collector is on an MOD schedule, the Data Collector obtains monitoring level changes from the applied schedule.

In a z/OS environment, a WebSphere server instance is represented by a Data Collector definition. It serves as a template for all the Data Collectors in the server regions belonging to the same server instance. This means that while you may be configuring the Data Collector for a server instance, the configuration is actually used by all the Data Collectors in the server regions when monitoring the applications.

Configuring a Data Collector

Use the Data Collector Overview page to configure the Data Collectors. The table at the top of the page shows the configured Data Collectors, while the table at the bottom of the page displays the unconfigured Data Collectors. When you configure a Data Collector, the system removes it from the unconfigured Data Collectors list and displays it with the configured Data Collectors. You can also apply a configuration to a Data Collector from the Apply page, see “Applying a Configuration” on page 19.

Note: The system assigns a name to the Data Collector. On the non z/OS platform, the Data Collector's name is a combination of the Admin Server name and the Application Server name. For the z/OS platform, the name is a combination of the Sysplex name and the Application Server name. The name cannot be changed.

To configure a Data Collector:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Go to the Unconfigured Data Collectors on the bottom of the page.
3. Select a configuration from the Apply a Configuration drop-down menu.
4. Click **Select All** or click in the individual check box of the unconfigured Data Collector you want to configure.
5. Click **Apply**.

The Data Collector displays in the list of configured Data Collectors at the top of the page.

Unconfiguring a Data Collector

Use the Data Collector Overview page to unconfigure the Data Collectors. In general, there is only one scenario that requires you to unconfigure a configured Data Collector: If you decide to retire or re-deploy an application server, you should unconfigure the Data Collector and the system will remove its configuration record from the WSAM database. When you unconfigure a Data Collector, the system removes it from the configured Data Collectors list and displays it with the unconfigured Data Collectors.

Note: Once a Data Collector is retired, its data will be purged.

To unconfigure a Data Collector:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Go to the Configured Data Collectors at the top of the page.
3. Click the unconfigure button for the Data Collector you want to unconfigure.

The Data Collector displays in the list of unconfigured Data Collectors at the bottom of the page.

Disabling a Data Collector

If you want to stop the Data Collector from sending and receiving data, you can disable the Data Collector. This is similar to a pause as opposed to unconfiguring the Data Collector (which would cause data to be lost.)

To disable a Data Collector:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Click **Disable** next to the Data Collector you want to disable.

The system disables the Data Collector and the button face changes to **Enable**.

Enabling a Data Collector

Enable your Data Collectors on the Data Collector Overview page. Manage monitoring on your system by enabling and disabling Data Collectors as needed.

To enable a Data Collector:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Click **Enable** next to the Data Collector you want to enable.

The system enables the Data Collector and the button face changes to **Disable**.

Note: If you stopped the Data Collector from sending and receiving data by disabling it, you can enable the Data Collector again when you are ready. Since a disabled Data Collector doesn't lose settings, you can simply turn it back on without any reconfiguration.

Creating a Configuration

Use this page to create a configuration. Group those classes you do not want to monitor in the Exclude (Classname) list. Any classes that are not in the list will be monitored.

If your Exclude (Classname) list of classes is too broad and you want to monitor a subset of the lower level classes, put them in the Exclude Override (Classname) list.

For example, the Exclude (Classname) list may include `com.sun.*`, while the Exclude Override (Classname) list includes `com.sun.java`. This means that WSAM will not monitor any `com.sun` classes except the Java™ classes.

If you are monitoring Composite Requests for applications that use MQ as a mechanism to bridge J2EE and CICS or IMS, then you must configure each participating Data Collector to monitor MQ.

Name the configuration for your Data Collectors. Create multiple configurations that monitor different classes.

To create a configuration:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Click **Create a Configuration** on the left navigation.

The Create page opens.

3. Enter the names of classes you want to ignore into the Exclude (Classname) list.
4. Enter the names of classes you want to monitor into the Exclude Override (Classname) list.
5. If you want the Data Collectors that use this configuration to monitor Composite Requests that use MQ:
 - a. Check the **Enable MQ** checkbox.
 - b. Fill in the **Exclude (Queue)** and **Exclude Override (Queue)** lists to specify which queues you want to monitor.
6. Enter a name for the configuration. (Required field)
7. Click **Save** to create the configuration or **Save & Apply** to create the configuration and apply it to a Data Collector.

Note: You can configure or change these options at any time.

Applying a Configuration

Use the Apply page to apply the configuration to a Data Collector. After you create a configuration, you must apply it to a Data Collector in order to start monitoring.

To apply a configuration:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Click **Configuration Library** on the left navigation.

The Data Collector Configuration List page opens.

3. Click the Apply icon next to the configuration you want to apply.

The Apply page opens.

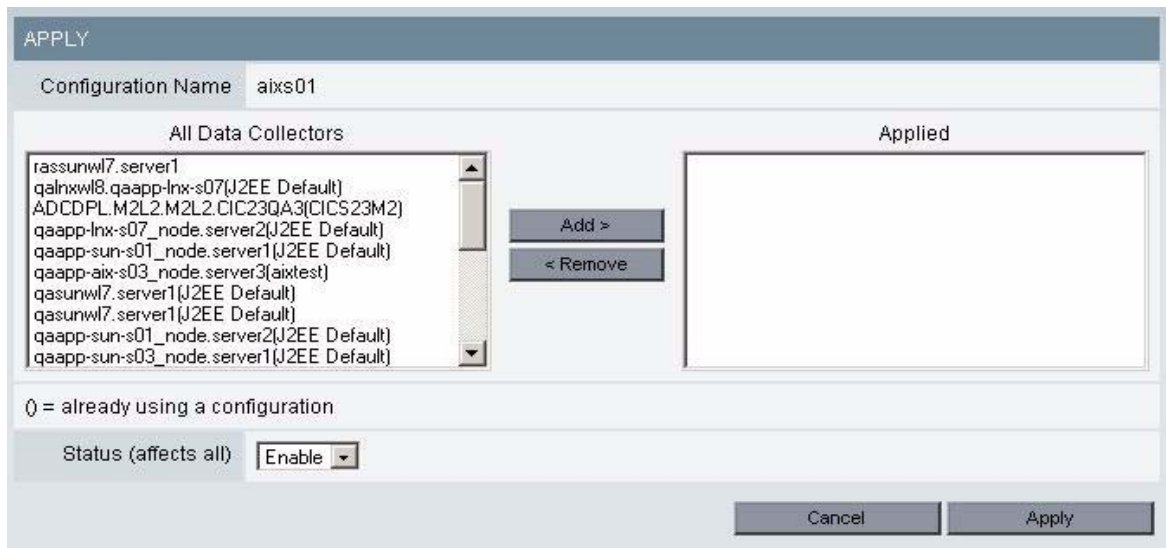


Figure 3. Apply page

4. Click to select Data Collector name(s) from the All Data Collectors box.

Note: To select multiple Data Collectors in a row, hold down the shift key during your selection. To add multiple Data Collectors non-continuously, Ctrl + click the servers for selection.

5. Click **Add** to apply the configuration to the Data Collector(s).
The Data Collectors' names appear in the Applied box.
6. Select **Enable** or **Disable** for the status of the configuration.
7. Click **Apply**.
The Data Collector now appears in the Configured Data Collector section of the Data Collector Overview page.

Modifying a Configuration

You can modify an existing configuration for your Data Collectors by updating the list of classes you monitor. Remove and add classes to the Exclude (Classname) list and Exclude Override (Classname) list to change what you monitor.

To modify a configuration:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.
The Data Collector Overview page opens.
2. Click **Configuration List** on the left navigation.
The Data Collector Configuration List page opens.
3. Click the Modify icon next to the configuration you want to modify.
The Modify page opens.
4. Enter the names of classes you want to ignore into the Exclude (Classname) list.
5. Enter the names of classes you want to monitor into the Exclude Override (Classname) List.
6. Click **Save** to save your modifications to the configuration.
The Data Collector Configuration List displays with the updated information.

Duplicating a Configuration

You can duplicate an existing configuration from your Data Collectors. Duplicate a configuration based on the selections made in an existing configuration.

To duplicate a configuration:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.
The Data Collector Overview page opens.
2. Click **Configuration Library** on the left navigation.
The Data Collector Configuration List page opens.
3. Click the duplicate icon next to the configuration you want to duplicate.
The Duplicate page opens.
4. Select an existing configuration from the drop-down menu.
5. Enter a new name for the configuration.
6. Click **Save**.

The new configuration displays in the Configuration List.

Deleting a Configuration

You can delete outdated configurations from the list to keep your list current. When you delete a configuration, the Data Collectors using that configuration will be unconfigured. See “Creating a Configuration” on page 18 and “Applying a Configuration” on page 19 to configure those Data Collectors.

To delete a configuration:

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

The Data Collector Overview page opens.

2. Click **Configuration List** on the left navigation.

The Data Collector Configuration List page opens.

3. Click the Delete icon next to the configuration you want to delete.

A confirmation box appears to warn you that deleting this configuration will unconfigure all the associated servers.

4. Click **OK** to delete the configuration.

The Configuration List displays without the deleted configuration.

Note: Remember to apply a new configuration to the servers you unconfigured while deleting the configuration.

Web Server Administration

The Web Server Administration section provides a method for adding and deleting Web servers from the Web Server Overview page.

Adding a Web Server

If you want to monitor a Web server’s activity, add the Web server to the Web Server Overview page. The Web Server Overview page shows at a glance indicators about requests, bytes per request and server availability.

Note: You can only monitor Apache Web servers.

To add a Web server to the Overview page:

1. From the top navigation, click **Administration > Server Management > Web Server Administration**.

The Web Server Administration page opens.

2. Enter the Host Name/IP Address and the port number for the Web server you want to add.

3. Click **Add**.

The Web server displays on the Web Server Management screen and is added to the Web Server Overview page.

Deleting a Web Server

Periodically, you may decide you no longer want to monitor a Web server. When this occurs, delete the Web server from the Web Server Overview page.

To delete a Web server from the Overview page:

1. From the top navigation, click **Administration > Server Management > Web Server Administration**.

The Web Server Administration page opens.

2. In the Web Server Management table, click the **X** next to the server you want to delete.

The Web server no longer displays in the Web Server Management table and on the Web Server Overview page.

3. Click **OK** in the confirmation window.

Chapter 4. Managing Server

Purpose

The Managing Server section is separated into two categories: System Properties and Self-Diagnosis. System Properties enables you to tune WSAM, while Self-Diagnosis provides you a method for debugging the Managing Server when problems arise.

Usage Overview

This feature helps you:

- Define the global and/or default data collection settings.
- Setup the default baseline settings for the Enterprise Overview Display.
- Set the SNMP Network configuration.
- Service WSAM by viewing the individual components running on the system.

Notes

Note: A Data Collector will stay at the default monitoring level unless a schedule overrides it or you override it.

System Properties

The System Properties are separated into three sections: the Data Collection Settings, the Enterprise Overview Display, and the SNMP Network Settings. Control the setup of WSAM using these properties.

Configuring the Data Collection Settings

Use the Data Collection Settings to set and modify the system settings for the Managing Server to regulate the frequency of data collection, the percentage of data stored, the level of monitoring, and the number of event records stored in traces.

To configure the Data Collection Settings:

1. From the top navigation, click **Administration > Managing Server > System Properties**.

The System Properties page opens.

DATA COLLECTION SETTINGS	
System Resources Polling Frequency	<input type="text" value="60"/> second(s)
Request Sampling Rate	<input type="text" value="3.0"/> L1% <input type="text" value="13.0"/> L2% <input type="text" value="23.0"/> L3%
Default Monitoring Level	<input type="text" value="(L2) Problem Determination Mode"/>
Maximum Method Records	<input type="text" value="10000"/>
Maximum IMS Message Data Length	<input type="text" value="256"/>
<input type="button" value="Reset"/> <input type="button" value="Save"/>	

Figure 4. Data Collection Settings

2. Enter the appropriate value for the following properties:

System Resources Polling Frequency – Set how often the system resources information is requested from your application servers. The default setting is 60 seconds.

Request Sampling Rate – The percentage of requests stored in the database for reporting and analysis. The default request sampling rate is 2%. There can be a distinct sampling rate for each monitoring level. Data collectors use the sampling rate associated with their current monitoring level.

Default Monitoring Level – The default monitoring level applies to all Data Collectors when they are first brought under the management of the Application Monitor.

Maximum Method Records – The maximum number of method entry/exit records. For each Method Trace, the records will be overwritten when they reach this value, saving the most recent records. The default value is 10,000.

3. Click **Save**.

Configuring the Enterprise Overview Display

Use the Enterprise Overview Display settings to set the Baseline Indicator and the Baseline Definitions. The Baseline Indicator consists of two settings: the percentages above the baseline that you determine to be slow, and very slow. Baseline Definitions define the performance you want to use as a standard of comparison.

To configure the Enterprise Overview Display:

1. From the top navigation, click **Administration > Managing Server > System Properties**.

The System Properties page opens.

2. On the left navigation, click **Enterprise Overview Display**.

The Enterprise Overview Display page opens.

ENTERPRISE OVERVIEW DISPLAY	
BASELINE INDICATORS - Enter a percentage up to 10 times (999%) for each Indicator's response.	
Indicator 1 (Slow Response)	>= <input type="text" value="25"/> %
Indicator 2 (Very Slow Response)	>= <input type="text" value="50"/> %
BASELINE DEFINITIONS - Select one baseline definition and enter the appropriate information.	
<input type="radio"/> Rolling Date	<input type="text" value="7"/> days
<input type="radio"/> Fixed Date (1-31 days)	Start Date <input type="text" value="Jan"/> <input type="text" value="01"/> <input type="text" value="2002"/> End Date <input type="text" value="Jan"/> <input type="text" value="01"/> <input type="text" value="2002"/>
<input checked="" type="radio"/> Fixed Response Time (0-10,000 ms)	<input type="text" value="1000"/> (ms)
<input type="button" value="Reset"/> <input type="button" value="Save"/>	

Figure 5. Enterprise Overview Display

3. Enter the appropriate value for the following properties:

Baseline Indicators – The percentage above the baseline that you determine to be slow or very slow. “Slow Response” means the present response time is between 26% and 50% of the baseline as in the example above; “Very Slow Response” means the present response time exceeds 50% of the baseline. You can set the indicators to the level you desire.

Note: When the response time exceeds Indicator 1, an orange indicator will display on the Application Overview page; a red indicator means the response time has exceeded Indicator 2.

Baseline Definitions – The baseline the application must fall below for an average response time for all servers in the group.

Rolling Date – The number you place in this field will represent the days over which the average response will be calculated for the baseline in 5 minute increments over a 24 hour cycle. The response time on the Enterprise Overview page will be compared to the appropriate 5 minute interval of this baseline.

Fixed Date – The average response time per 5 minute increments from between the start date and end date will become the baseline against which your current response times on the Enterprise Overview page will be compared.

Fixed Response Time – The response time entered in this field will become the response time against which your current response times on the Enterprise Overview page will be compared.

4. Click **Save**.

Configuring the SNMP Network

Use the SNMP Network settings to indicate the configuration for the SNMP server. A test message will be sent to the SNMP Network Manager to test for connectivity.

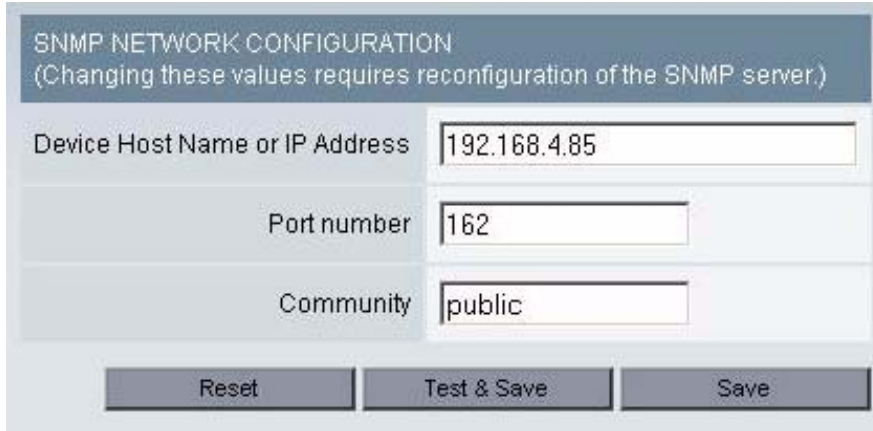
To configure the SNMP Network:

1. From the top navigation, click **Administration > Managing Server > System Properties**.

The System Properties page opens.

2. On the left navigation, click **SNMP**.

The SNMP Network Configuration page opens.



SNMP NETWORK CONFIGURATION (Changing these values requires reconfiguration of the SNMP server.)	
Device Host Name or IP Address	<input type="text" value="192.168.4.85"/>
Port number	<input type="text" value="162"/>
Community	<input type="text" value="public"/>
<input type="button" value="Reset"/> <input type="button" value="Test & Save"/> <input type="button" value="Save"/>	

Figure 6. SNMP Network Configuration

3. Enter the appropriate value for the following properties:

Device Host Name or IP Address – The name or address of your SNMP Network Manager, to which SNMP messages will be sent.

Port Number – The port number of your SNMP Network Manager.

Community – A string that is part of the SNMP protocol.

4. Click **Test** to send a test message to the SNMP Network Manager.
5. Click **Save** to save your settings.

Self-Diagnosis

This section is designed for the Support staff to service WSAM. The Self-Diagnosis provides a view of all the components currently running, their states and attributes. WSAM consists of the following components: Kernel, Data Collector Controller, Publish Server, Global Publish Server, Message Dispatcher, Polling Agent and Archive Agent. Since WSAM is designed to run in a loosely-coupled, dynamic environment, individual components can be up or down without affecting the integrity of the whole system.

Viewing the Self-Diagnosis for the Kernel

The Kernel is a directory service dedicated to WSAM that monitors the components that join and leave the network. The Self-Diagnosis provides a view of all the components on the Kernel currently running and their attributes. Under normal condition, there are two copies of the Kernel running simultaneously.

To view the Self-Diagnosis for the Kernel:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens displaying the first Kernel's data.

2. Use the left navigation to view the Self-Diagnosis for the other Kernel.

Viewing the Self-Diagnosis for the Archive Agent

The Archive Agent collects data from the Publish Server and archives it into the database for reporting. The Self-Diagnosis provides a view of the Archive Agent's attributes, as well as all the components with which the Archive Agent has relationships.

To view the Self-Diagnosis for the Archive Agent:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens.

2. From the left navigation, click the Archive Agent link.
3. Click to select the Archive Agent you want to view.

The data for the selected Archive Agent displays.

Viewing the Self-Diagnosis for the Publish Server

The Publish Server receives data from the Data Collector and aggregates it based on different needs. The Self-Diagnosis provides a view of the Publish Server's attributes, as well as all the components with which the Publish Server has relationships.

To view the Self-Diagnosis for the Publish Server:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens.

2. From the left navigation, click the Publish Server link.
3. Click to select the Publish Server you want to view.

The data for the selected Publish Server displays.

Viewing the Self-Diagnosis for the Global Publish Server

The Global Publish Server keeps track of Composite Requests, as they move from one server to another. The Self-Diagnosis provides a view of the Global Publish Server's attributes, as well as all the components with which the Global Publish Server has relationships.

To view the Self-Diagnosis for the Global Publish Server:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens.

2. From the left navigation, click the Global Publish Server link.
3. Click to select the Global Publish Server you want to view.

The data for the selected Global Publish Server displays.

Viewing the Self-Diagnosis for the Data Collector Controller

The Data Collector Controller is the part of a Data Collector that regulates the behavior of a Data Collector, including the monitoring level, filter list, and enable or disable status. The Self-Diagnosis provides a view of the Data Collector's attributes, as well as all the components with which the Data Collector has relationships.

To view the Self-Diagnosis for the Data Collector Controller:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens.

2. From the left navigation, click the Data Collector Controller link.
3. Click to select the Data Collector Controller you want to view.

The data for the selected Data Collector Controller displays.

Viewing the Self-Diagnosis for the Message Dispatcher

The Message Dispatcher sends out emails of performance reports and trap actions from the Performance Analysis & Reporting and the Trap & Alert Management features. The Self-Diagnosis shows all the attributes of the Message Dispatcher, such as total number of emails sent.

To view the Self-Diagnosis for the Message Dispatcher:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens.

2. From the left navigation, click the Message Dispatcher link.
3. Click to select the Message Dispatcher you want to view.

The data for the selected Message Dispatcher displays.

Viewing the Self-Diagnosis for the Polling Agent

The Polling Agent maintains the list of Web servers monitored by WSAM. The Self-Diagnosis provides a view of the Polling Agent's attributes, as well as all the components with which the Polling Agent has a relationship.

To view the Self-Diagnosis for the Polling Agent:

1. From the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

The Self-Diagnosis page opens.

2. From the left navigation, click the Polling Agents' link.
3. Click to select the Polling Agent you want to view.

The data for the selected Polling Agent displays.

Chapter 5. Monitoring on Demand™

Purpose

Monitoring on Demand™ enables you to create a schedule and respond to problems by defining the level of monitoring.

Usage Overview

This feature helps you:

- Set the level of monitoring best suited for your servers at a given time.
- Create schedules to automatically change the monitoring level for your servers.

User Scenarios

Scenario 1: Setting a schedule for detailed monitoring at night

Your manager wants you to monitor your servers at Level 3 during off hours because that's when the load is the lightest. As the WSAM administrator, you set a schedule to monitor the servers during business hours at Level 1 and at night at Level 3.

Scenario 2: Overriding the monitoring level during an emergency

An emergency arises that requires Level 3 monitoring to locate a problem. As the WSAM administrator, you override the current schedule and set the monitoring level to Level 3. After fixing the problem, you can reset the monitoring level or wait until the next schedule change.

Notes

Note: The following describe the different monitoring levels available:

L1 (Production Mode) – this monitoring level provides Availability Management, System Resources and basic request-level data. This monitoring level least affects the CPU overhead per transaction and is appropriate for servers that are not malfunctioning.

L2 (Problem Determination Mode) – this monitoring level provides production level monitoring plus advanced request data, including external component and CPU information, as well as additional monitoring fields and functions. Under Problem Determination mode you can view Component Traces. These are traces that show J2EE request-related events that are made to external services. This level should be used when you suspect a problem or need to capture data about external events but do not need all the method-level data.

L3 (Tracing Mode) – this is the most powerful monitoring level, therefore only this level utilizes all reporting elements available. For example, in L3 the Server Activity Display shows additional data for the following columns: Accumulated CPU, Last Known Class Name, Last Known Method, and Last Known action. In addition, on the Request Detail page, the Method Trace with SQL statements are also available. L3 has inherently higher overhead than the other monitoring levels. Therefore, this level should be used for servers that have been selected for diagnostics and detailed workload characterization.

Tracing Intent (z/OS Platform Only)

Tracing is the execution path of applications at the Java class/method level. This requires JVMPI, a feature of the JVM, to be enabled when L3 monitoring is active. The method tracing is relatively efficient compared to the amount of data that is being collected (method and SQL data), enabling analysis to be performed using metrics such as elapsed time, CPU time, SQL types and table names.

For tracing to work on z/OS, the application Java classes must be running in interpretive mode, otherwise classes compiled by the JIT compiler will not appear in WSAM. Running in interpretive mode will impact performance, but since this is only applicable when you are tracing, the extra overhead will not have much impact in practice.

When a Data Collector runs at L3 at startup time, the JIT compiler will skip application Java classes. This will enable full trace data to be obtained.

However, when a server region is started up at L1, you must indicate whether the region may be later set to L3 (via the MOD scheduler or override function.) By indicating to WSAM that there is such intent, WSAM will instruct the JVM to skip compiling the relevant application Java classes in the EAR and WAR files during the server start up. This will ensure that you will obtain trace data whenever the server is switched to L3.

This intent switch is implemented through the use of the variable named **wsam_intent_trace**. Turn it on by setting it to yes. Any other value is treated as no, including not specifying the variable at all.

- Specify `JITC_COMPILEOPT=SKIP{classnames}{methodnames}` in the current.env file to add classes and methods to the list of classes to trace.
- Set the variable `WSAM_APPSERVER` to the name of the WebSphere application server (NOT the server instance). Default is BBOASR2.
- If `wsam_intent_trace=no` (or not specified) and the initial monitoring level is 1, then no trace information is generated.
- If `wsam_intent_trace=yes`, the initial level is irrelevant, and all classes are traced.

The list of classes to trace is built as follows:

- The initial list of classes is specified in the <apps> directory of the specific J2EE application server.
- Classes in the “classes_not_to_trace” environment variable (defined in the datacollector.env file) are removed from the list of classes to trace.
- Classes in the “classes_to_trace” environment variable (defined in the datacollector.env file) are added to the list of classes to trace.

Put this name/value pair in the current.env file of the server instance before starting up the server regions.

Managing MOD

The MOD Console page displays the following information for a Group of Servers: Platform, Schedule Name, Current Monitoring Level and Current Request Sampling Rate. The MOD Console page also provides access to the Modify Server Settings page, Schedule Detail page, Schedule Management page and the Duplicate Schedule page.

To view the MOD Console page:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens displaying all the servers.
2. To filter the information by group, click to select the group that you want to view from the drop-down menu.
3. Click the heading by which you want to sort. You can only sort by headings that are underlined. When the page refreshes, the results display sorted by the selected heading. To reverse the sort, click the same heading a second time.
4. To modify a schedule, click the Schedule Name link.
5. Click the arrow to modify a server's setting or override a monitoring level.

To go to the Schedule Management page:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens.
2. Click **Schedule Management** on the left navigation.
The Schedule Management page opens displaying the schedules and their status.

Selecting a Monitoring Level

Set the monitoring level based on the anticipated load on the server and the information you want to capture in the database.

To select a monitoring level for a server:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens.
2. Click **Schedule Change/Override** for the server which you need to select a new monitoring level.
The Modify Server Settings page opens.

SELECTED GROUP/SERVERS				
GROUP/SERVER	Platform	Schedule Name	Current Level	Current Sampling
qaapp-aix-s01_node.server1	AIX	testcase_week	L3	2%

SETTING OPTIONS	
Schedule Selection	No Schedule
Override Monitoring Level	No Override
Sampling Rate	<input type="text" value="2"/> (L1%) <input type="text" value="100"/> (L2%) <input type="text" value="100"/> (L3%) <input type="checkbox"/> System Default
<input type="button" value="Cancel"/> <input type="button" value="OK"/>	

Figure 7. Modify Server Settings

Note: For the z/OS platform, the server displayed is the name of a server instance. When a server instance is selected, all the server regions belonging to the server instance will be listed. You can override the monitoring level of a particular server region.

3. Select No Schedule from the Schedule Selection drop-down menu.
4. Select a monitoring level from the Override Monitoring Level drop-down menu.
5. Fill in the Sampling Rate boxes or select to use System Default.
6. Click **OK**.

Note: The System Default values for Sampling Rate are defined on the Data Collection Settings page of System Properties. The system defaults are used for any server when there is no applicable schedule.

Changing the System Default Setting

WSAM comes with default Monitoring Levels and Sampling Rates that are used by servers connected to WSAM and that are not on a monitoring schedule. For example, when a server is first configured, it will not have a schedule and therefore will use the defaults. The defaults are in effect between schedules, as well.

The Sampling Rate can be set per monitoring level. For example, if the Default Sampling Rates for levels 1 and 3 are 5% and 2%, respectively, when you change the default monitoring level from 1 to 3, WSAM changes from collecting 5% of the data to 2%.

Note: The default Sampling Rate for all levels is 2%.

To change the System Default setting:

1. From the top navigation, click **Administration > Managing Device > System Properties**.

The System Properties page opens.

2. Enter a percentage in the boxes for the Request Sampling Rate.
3. Select a new default monitoring level from the Default Monitoring Level drop-down menu.
4. Click **Save**.

Note: If you override the monitoring level of a specific server and do not specify a Request Sampling Rate, WSAM will use the system default Request Sampling Rate associated with the monitoring level at which the server is set.

Note: Changing the default Request Sampling Rate only affects servers added to the system subsequent to the rate change. It does not change the Request Sampling Rate of servers already deployed.

Creating a Schedule

At times, you may need to monitor a server in more detail. You can create a schedule that changes the monitoring level based on a specified date and time. Using the schedule, modulate the monitoring level at different times based on the anticipated load on the server.

Note: You may want minimum 1 or max 5 percent of your servers running at L3, either as dedicated servers, or only during non-peak hours. This arrangement will give you good quality data for workload tracing and application sizing.

To create a schedule:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens.
2. Click **Create Schedule** on the left navigation.
The Schedule Detail page opens where you can create a new schedule.
3. Enter a Schedule Name for the new schedule.
4. Select the Day of the Month or the Day of the Week when you want your schedule event to take effect; for example, you may want the event to start on the 5th of every month or on every Monday.
5. Select the Hour and Minute when the schedule event starts.
6. Select the Monitoring Level that best suits your needs: L1, L2, or L3.
7. Click **Add** to insert the event into the schedule. Each schedule can include multiple monitoring level changes; to save each change, click **Add**.
8. To save the schedule, click **OK**.

The Schedule Management page opens with the new schedule displayed.

Note: In the event of a schedule conflict, the most recently entered event will take precedence.

Applying a Schedule

After creating a schedule, you may apply it to a server that needs monitoring. Each server can be on only one schedule at a time.

To apply a schedule:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™(MOD) Console page opens.

2. Click **Schedule Change/Override** for the server to which you want to apply a schedule.
The Modify Server Settings page opens.
3. Click to select a schedule from the Schedule Selection drop-down menu for the server.
4. Enter a percentage in the boxes for Sampling Rate or choose the default settings by checking the System Default box.
5. Click **OK**.
The MOD Console page displays the server with the schedule applied in the table.

Overriding the Monitoring Level

In case you need to collect more or less detailed data in a particular period of time, you may override the current monitoring level until the next scheduled monitoring level occurs.

The following is an example schedule for a server:

Day of the Week	Start Time	Monitoring Level
Monday	00:00	L1
Monday	08:00	L2
Monday	16:00	L3

Assume the time is now 07:00 and the system is currently running at L1. If you want to collect more detailed data information, you can override the scheduled monitoring level from L1 to L3. The monitoring level will change to L2 when the next scheduled monitoring level begins at 08:00.

To override the monitoring level:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™(MOD) Console page opens.
2. Click **Schedule Change/Override** for the server whose monitoring level you want to override.
The Modify Server Settings page opens.
3. Click to select a monitoring level from the Override Monitoring Level drop-down menu.
4. Click **OK**.

Modifying a Schedule

If you find that an existing schedule is not providing the correct level of monitoring, modify the schedule to reflect your needs.

To modify a schedule:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens.
2. Click the Schedule Name for the schedule you want to modify on the console page or the Schedule Management page.

- The View MOD Schedule page opens.
3. Click **Modify Schedule**.
The Schedule Detail page opens.
 4. Enter the information to modify the schedule.
 5. Click **Add** to insert an event into the schedule. Each schedule can include multiple monitoring level changes; to save each change, click **Add**.
Note: Changes take effect immediately.
 6. To save the schedule, click **OK**.

Deleting a Schedule

Keep your schedules updated by deleting schedules from the system that are no longer in use.

To delete a schedule:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens.
2. Click **Schedule Management** on the left navigation.
The Schedule Management page opens.
3. Click **X** or **Delete** next to the schedule you want to remove.
4. At the confirmation box, click **OK** to delete the schedule.

The Schedule Management page displays without the deleted schedule.

Note: If a schedule is currently being used by a server, you have to apply another schedule to that server or it will automatically apply the System Default after you delete the schedule.

Duplicating a Schedule

Save time by duplicating schedules. Duplicating a schedule allows you to quickly create a new schedule based on the settings of an existing schedule.

To duplicate a schedule:

1. From the top navigation, click **Administration > Monitoring on Demand™**.
The Monitoring on Demand™ (MOD) Console page opens.
2. Click **Duplicate Schedule** on the left navigation.
The Duplicate Schedule page opens.
3. From the Schedule drop-down menu, select the schedule you want to duplicate.
4. Enter a new name for the duplicated schedule.
5. Click **Save**.

The Schedule Management page opens displaying the duplicated schedule.

Chapter 6. Systems Overview

Purpose

Systems Overview allows you to evaluate the availability of your entire system by looking at recent performance trends.

Usage Overview

This feature helps you:

- Monitor your enterprise's availability.
 - View dashboards of Enterprise, Server Group, Server, Web Server and Portal metrics.
 - Quickly isolate deviations from baseline response time thresholds.
 - Monitor both server availability and application availability.
- Isolate problematic servers.
 - Drill down to the problematic server group or server.
 - Identify problematic resources on individual servers.
- Easily jump to other relevant product features to continue isolating problems.

User Scenarios

Scenario 1: Verifying customer response time complaints

Customer service has been receiving complaints that your company's Websites have been responding slowly. As one of the administrators of the servers, the inquiry has come to your attention. Upon opening the Enterprise Overview page, you immediately see that three of your production servers are no longer available. You also verify that the response time has degraded.

Scenario 2: Diagnosing an application problem

Customers have been complaining that they cannot place orders. As one of your company's administrators, you go to the Enterprise Overview page and see that all the servers are up. You find the group that appears to have the highest response time and drill down to the Server Overview page where you see that a database connection pool is saturated.

Enterprise Overview

The Enterprise Overview displays the availability of all servers in all server groups. The page shows how many servers in a group are available out of the total number of servers in the group, the percentage of servers in the group that are available, and the total number of requests completed on the servers in the group for each 5 minute period in the last hour.

Note: Requests include the first JSP, Servlet or EJB request coming into the application server, which represents a top-level user transaction.

The Enterprise Overview page also provides a comparison of the current response time to a baseline response time. The baseline is on the server group level and can be modified on the Modify Group page, located under **Administration > Server Management > Server Groups**. When the current response time exceeds the baseline response time for any 5 minute period, the Enterprise Overview page provides an indicator under the current response time for that period. Mouse over the indicator or bar to view the actual data.

To open the Enterprise Overview page:

1. From the top navigation, click **Availability > Systems Overview > Enterprise**.

Note: If you set the Enterprise Overview as your default page, the Enterprise Overview will open when you click the WSAM logo.

The Enterprise Overview page opens.

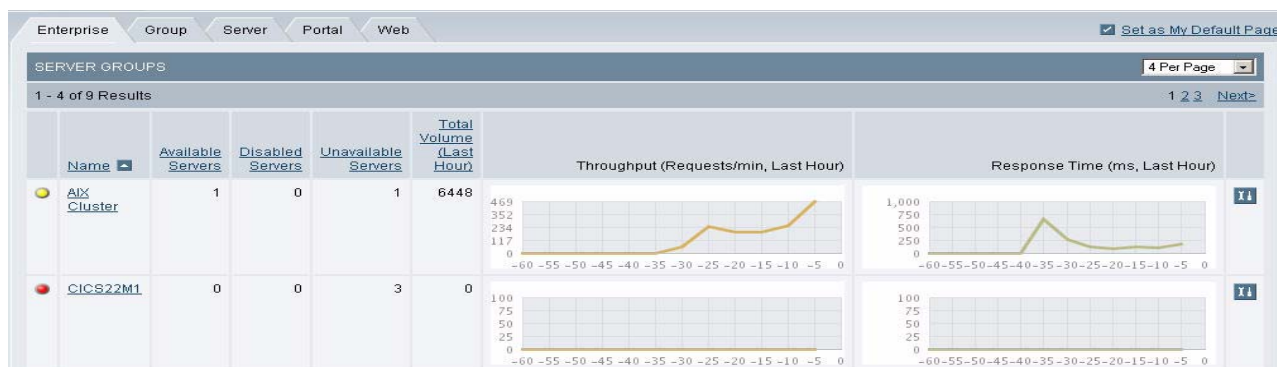


Figure 8. Enterprise Overview

Note: The information that displays on the Enterprise Overview page depends on the user’s account setup.

Note: Only the servers the user has access to view will display.

Note: The percentage of servers up and running in the group is rounded up.

Note: The Volume Throughput shows the number of completed Web requests processed in five minute increments for the past hour.

Note: Each group displays in its own box labeled with the name of the group.

Group Overview

The Group Overview page provides a high-level overview of activity for each server in the group. Specifically, the Overview includes the response time and throughput for the last hour as well as the current monitoring level for each server. This allows you to efficiently discern whether all the servers in the group are properly functioning.

To open the Group Overview page:

1. From the top navigation, click **Availability > Systems Overview > Group**.

The Group Overview selection page opens.

2. Select a Group from the drop-down menu.

The Group Overview page opens, displaying data for the selected group.

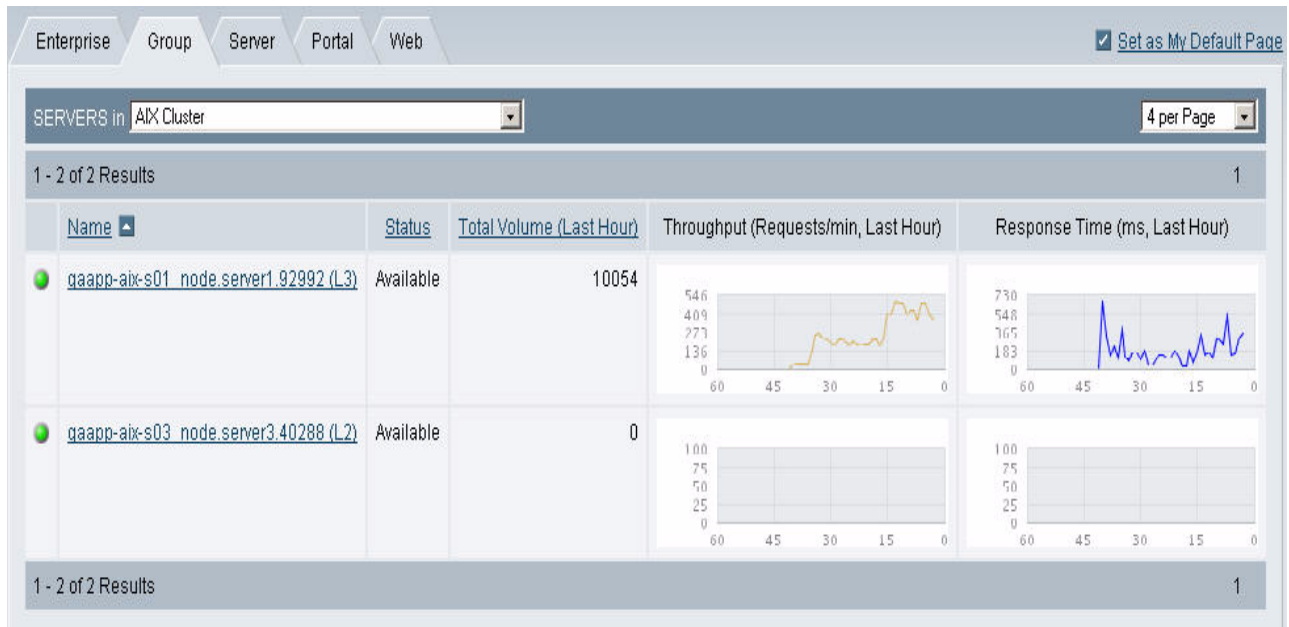


Figure 9. Group Overview page

Server Overview

The Server Overview page displays comprehensive server information, activity, statistics, and resource data for the selected server. View the summary server data to understand the status of your applications and application server behavior. This page provides vital information for determining the health of your server.

To open the Server Overview page:

- From the top navigation, click **Availability > Systems Overview > Server**.
The Server Overview selection page opens.
- Select a Group and a Server from the drop-down menus.
The Server Overview page opens, displaying data for the selected server.

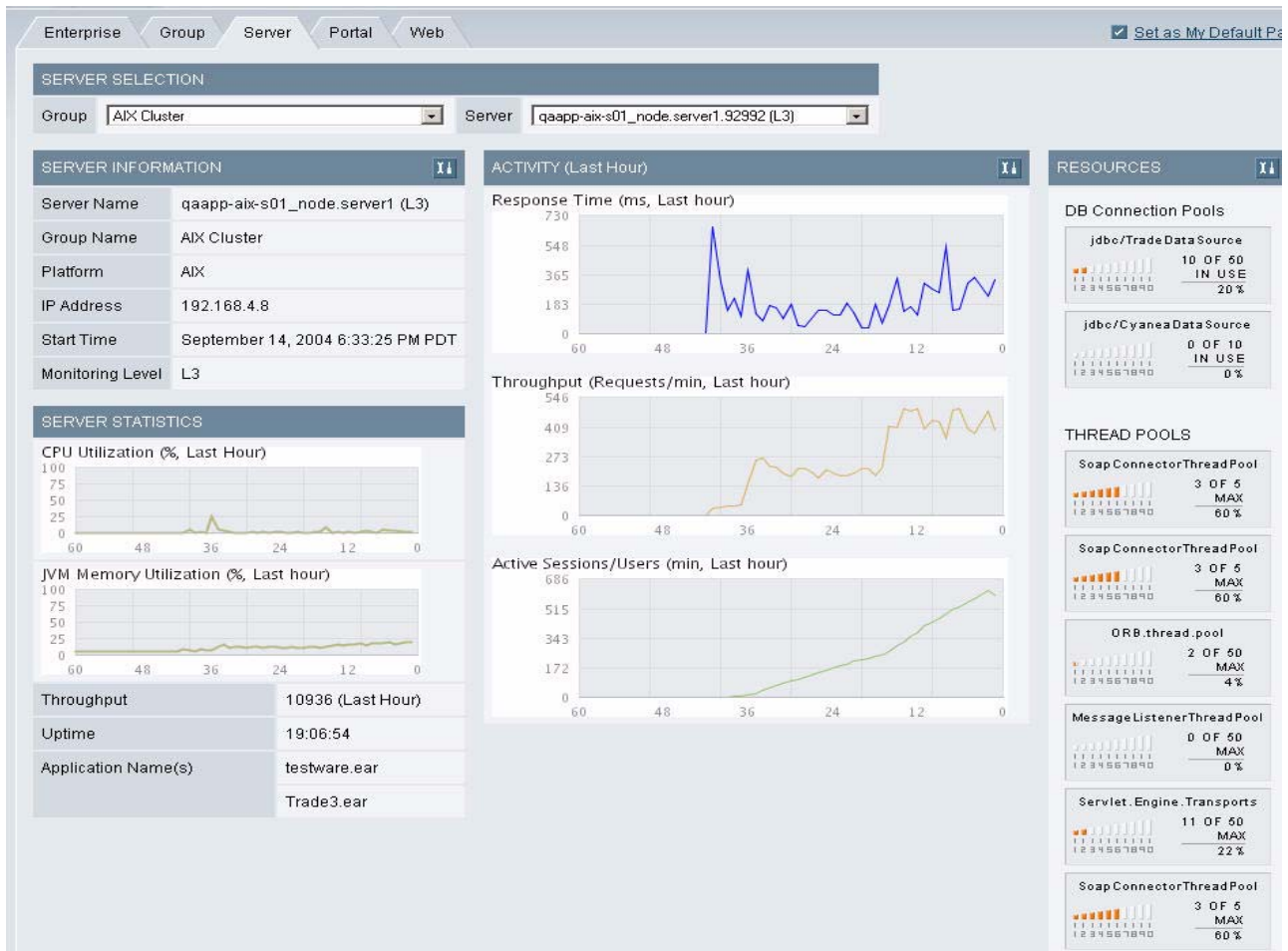


Figure 10. Server Overview page

Web Server Overview

The Web Server Overview page offers a quick method for viewing whether your Web servers are functioning properly. While performing problem determination functions, it is useful to know the status of your Web servers. You can efficiently eliminate your Web servers as the source of the problems by checking the Web Server Overview page and drilling down to a particular Web server's details. (See "Web Server Details" on page 41.)

To open the Web Server Overview page:

1. From the top navigation, click **Availability > Systems Overview > Web Server**.
2. The Web Server Overview page displays the Web Servers and summary data.

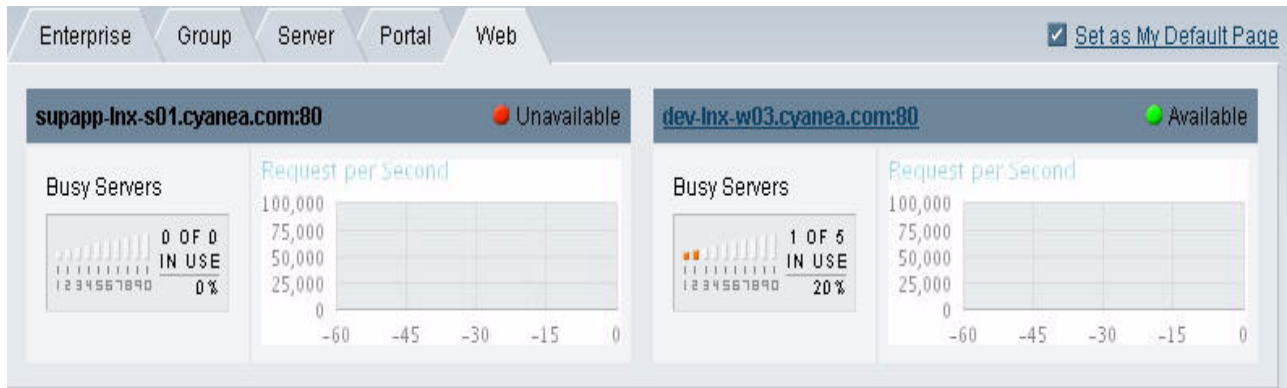


Figure 11. Web Server Overview page

Web Server Details

The Web Server Details page displays the activity data for the selected Web server being monitored. The Web servers that are available will have an associated detail page, while unavailable Web servers will not.

To open the Web Server Details page:

1. From the top navigation, click **Availability > Systems Overview > Web Server**.
The Web Server Overview page opens.
2. Click the link of the Web server's name whose details you want to view.
The Web Server Details page opens.

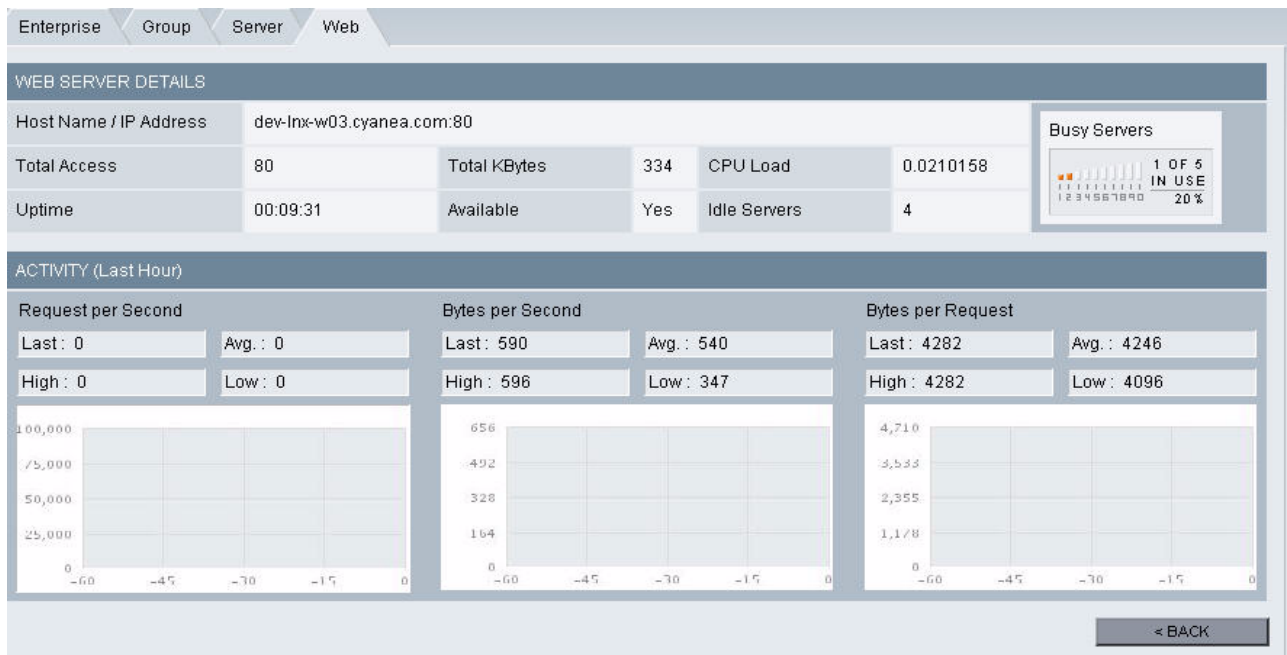


Figure 12. Web Server Details page

Portal Overview

The Portal Overview page helps you assess the portals in your system and how they are operating. You can monitor the status of your portals from the slowest portals to the portals with the highest throughput for the last hour. In addition, view the metrics for the portals including Average Response Time and Count for authentication and authorization, as well as credential and content access metrics.

To open the Portal Overview page:

1. From the top navigation, click **Availability > Systems Overview > Portal**.
2. Select a server from the drop-down menu.

The Portal Overview page displays the Portals and summary data.

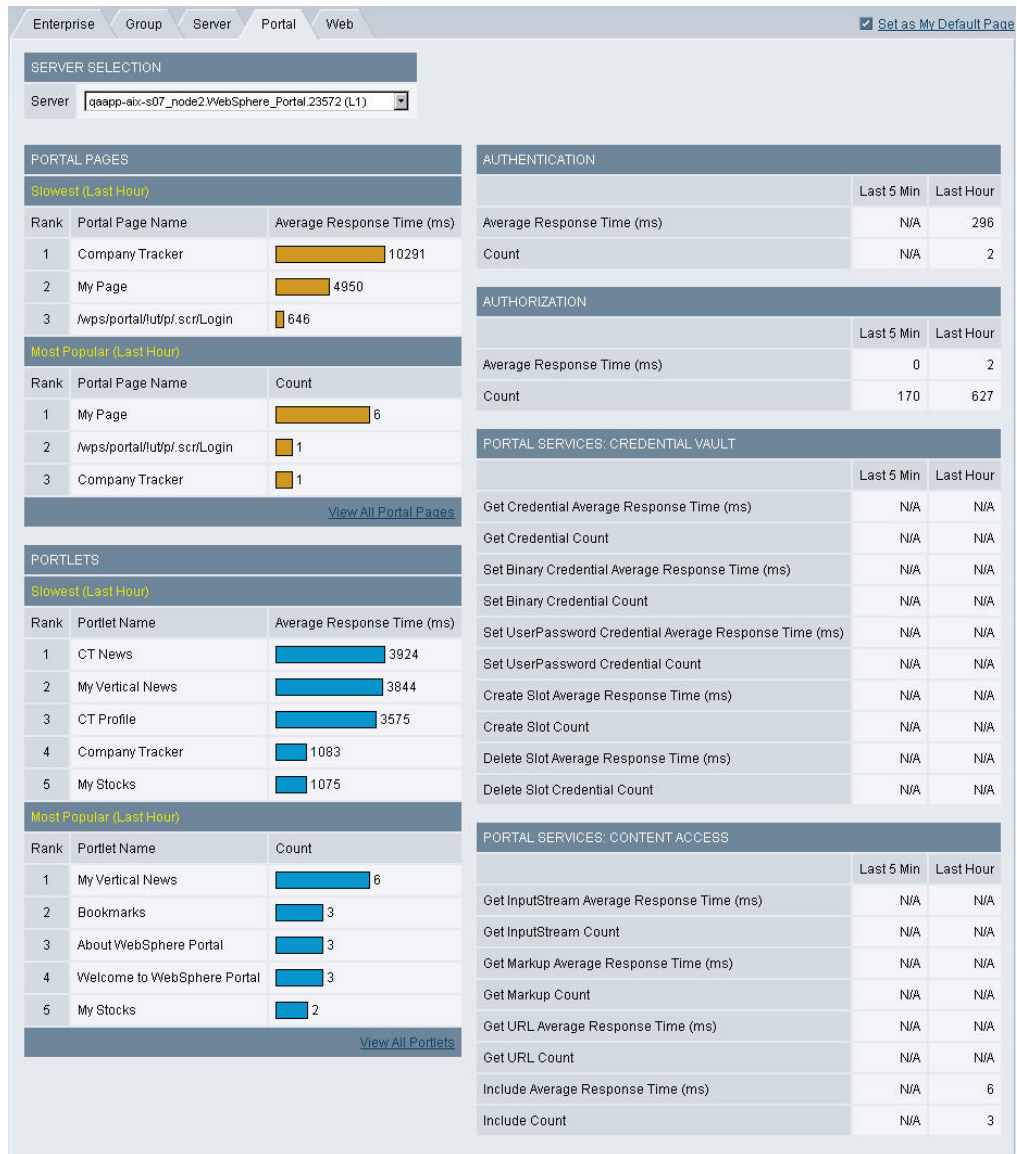


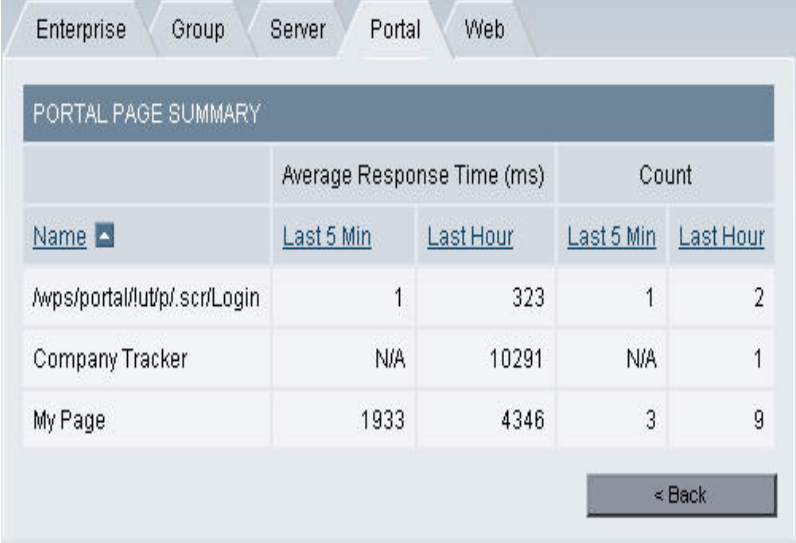
Figure 13. Portal Overview

Portal Page Summary

The Portal Page Summary displays data for the portals available on the system. The metrics include the Average Response Time and Count for the last five minutes and the last hour.

To open the Portal Page Summary:

1. From the top navigation, click **Availability > Systems Overview > Portal**.
The Portal Overview page displays the Portals and summary data.
2. Click the View All Portal Pages link.
The Portal Page Summary displays the portals and summary data.



The screenshot shows a web interface with a navigation bar containing 'Enterprise', 'Group', 'Server', 'Portal', and 'Web'. Below the navigation bar is a section titled 'PORTAL PAGE SUMMARY'. This section contains a table with columns for 'Name', 'Average Response Time (ms)', and 'Count'. The 'Average Response Time' and 'Count' columns are further divided into 'Last 5 Min' and 'Last Hour' sub-columns. The table lists three portals: 'Awsportal/ut/p/scr/Login', 'Company Tracker', and 'My Page'. A '< Back' button is located at the bottom right of the table.

Name ▾	Average Response Time (ms)		Count	
	Last 5 Min	Last Hour	Last 5 Min	Last Hour
Awsportal/ut/p/scr/Login	1	323	1	2
Company Tracker	N/A	10291	N/A	1
My Page	1933	4346	3	9

Figure 14. Portal Page Summary

Portlet Summary

The Portlet Summary page displays data for the portlets available on the system. The metrics include the Average Response Time and Count for the last five minutes and the last hour.

To open the Portlet Summary page:

1. From the top navigation, click **Availability > Systems Overview > Portal**.
The Portal Overview page displays the Portals and summary data.
2. Click the View All Portlets link.
The Portlet Summary page displays the portlets and summary data.

PORTLET SUMMARY				
Name	Average Response Time (ms)		Count	
	Last 5 Min	Last Hour	Last 5 Min	Last Hour
About WebSphere Portal	N/A	37	N/A	5
Bookmarks	N/A	174	N/A	5
Company Tracker	N/A	1083	N/A	1
CT Chart	N/A	692	N/A	1
CT News	N/A	3924	N/A	1
CT Profile	N/A	3575	N/A	1
CT Stock	N/A	802	N/A	1
My Stocks	N/A	1003	N/A	4
My Vertical News	N/A	3306	N/A	9
My Weather	N/A	338	N/A	4
Welcome to WebSphere Portal	N/A	14	N/A	5

Figure 15. Portlet Summary

WLM Associated Service Class Summary

The WLM Associated Service Class Summary page offers a way to view selected data from the Workload Manager (WLM) for z/OS and OS/390, for the address space associated with a particular server, as well as its associated service class data and service class period data.

Note: This feature is only available for z/OS servers.

The address space properties include:

- Server name
- Associated Report Class
- Resource Group
- Workload
- Server Space
- Associated Service Class Goals

The associated service class properties include:

- Service Class Name
- Description
- Associated Workload
- Associated Resource Group
- Number of Service Class Periods
- WLM Mode
- Last Initialized Time
- Data Collected Time
- Policy Name
- Policy Activated Time
- Policy Activator User ID
- Policy Activated System

- Current Delay Sample Intervals
- Number of times the WLM Sample Code ran

Each associated service class period data includes:

- Period Name
- Goal Type
- Response Time Units
- Goal % Value
- Importance Level
- Response Time Velocity
- Period Duration

Drill down on the period name to see the WLM Associated Service Class Period Detail.

To open the WLM Associated Service Class Summary page:

1. From the top navigation, click **Availability > Systems Overview > Server**. The Server Overview selection page opens.
2. Select a Group and a Server from the drop-down menus. The Server Overview page opens displaying data for the selected server.
3. Click the **WLM Associated Service Class Summary** from the tools button. The WLM Associated Service Class Summary page opens.

Service Class		Enclave					
ADDRESS SPACE PROPERTIES							
Server Name (Region)	ADCDPL.M2L2.M2L2.servqa.bdf (L3)	Associated Report Class		Associated Resource Group			
Associated Workload	STC	Server Space	Yes	Associated Service Class Goals	Applied		
ASSOCIATED SERVICE CLASS PROPERTIES							
Name	STCMED	Description	ExVel(60) Imp(2)	Associated Workload	STC		
Associated Resource Group		# of Service Class Periods	2	WLM Mode	Goal Mode		
Last Initialized Time	Sep 15, 2004 5:31:24 AM	Data Collected Time	Sep 28, 2004 10:37:41 AM	Policy Name	BASE		
Policy Activated Time	Apr 2, 2004 1:56:46 PM	Policy Activator User ID	CSFFF	Policy Activated System	M2L2		
Current Delay Sample Intervals (ms)	250	Total # WLM Sample Code	4,565,527				
ASSOCIATED SERVICE CLASS PERIOD							
Period	Goal Type	Response Time Units (msec., sec., min., hr.)	Goal % Value	Importance Level (1-5)	Response Time/Velocity (Goal Value)	Period Duration	
Period 1	Velocity Goal	Unknown	0	2	60	200	
Period 2	Percentile Goal	ms	50	2	2000	0	

Figure 16. WLM Associated Service Class Summary page

WLM Associated Service Class Period Detail

The WLM Associated Service Class Period Detail page offers a way to view selected data from the Workload Manager (WLM) for z/OS and OS/390, for a selected service class period. This includes the response time distribution detail, and possibly delay detail information, about each subsystem work manager.

The service class period properties include:

- Period Name
- Response Time
- Goal % Value
- Importance Level
- Response Time Velocity
- Period Duration
- Data Collected Time

The response time distribution detail includes the Total Transactions and the Goal and Transaction information for all 14 goal "buckets."

The delay data includes, if available, for each subsystem work manager, the number of programs in each of the Active, Ready, Waiting, Local, Sysplex and Network states, for both the begin-to-end phase and execution phase.

To open the WLM Associated Service Class Period Detail page:

1. From the top navigation, click **Availability > Systems Overview > Server**.
The Server Overview selection page opens.
2. Select a Group and a Server from the drop-down menus.
The Server Overview page opens displaying data for the selected server.
3. Click the **WLM Associated Service Class Summary** from the tools button.
The WLM Associated Service Class Summary page opens.
4. Click the name of one of the associated service class periods.
The WLM Associated Service Class Period Detail page opens.


Service Class		Enclave					
ASSOCIATED SERVICE CLASS PERIOD PROPERTIES							
Name	STCMED	Response Time Units (msec., sec., min., hr.)	ms	Goal % Value	50	Importance Level (1-5)	2
Period	2	Response Time/Velocity (Goal Value)	2000	Period Duration	0	Data Collected Time	Sep 28, 2004 10:42:40 AM
RESPONSE TIME DISTRIBUTION DETAIL							
Total Transactions		230					
	Buckets	Transactions	Buckets	Transactions	Buckets	Transactions	
	< 50%	0	90 - 100%	0	130 - 140%	0	
	50 - 60%	0	100 - 110%	1	140 - 150%	0	
	60 - 70%	0	110 - 120%	1	150 - 200%	0	
	70 - 80%	1	120 - 130%	1	200 - 400%	1	
	80 - 90%	0			> 400%	 225	
DELAY DETAIL -- CB							
Begin-to-End Phase	Active	Ready	Waiting	Local	Sysplex	Network	
	0	0	0	0	0	0	
Execution Phase	Active	Ready	Waiting	Local	Sysplex	Network	
	0	0	0	0	0	0	
< Back							

Figure 17. WLM Associated Service Class Period Detail page

WLM Enclave

The WLM Enclave page offers a way to view selected data from the Workload Manager (WLM) for z/OS and OS/390, for an enclave.

The enclave properties include:

- Job Name

- ASID
- System Name
- Subsystem Type
- Subsystem Name
- Time of Snapshot

The enclave detail includes Enclave Token, Active and Cumulative CPU Time. The tokens that appear depend on the version of z/OS:

- For z/OS 1.2, all tokens in the Enclave are shown. (There is no filtering on the basis of server instance.)
- For z/OS 1.3 and above, only the tokens in the Enclave initiated by the server instance are shown.

To open the WLM Enclave page:

1. From the top navigation, click **Availability > Systems Overview > Server**.
The Server Overview selection page opens.
2. Select a Group and a Server from the drop-down menus.
The Server Overview page opens displaying data for the selected server.
3. Click the **WLM Associated Service Class Summary** from the tools button.
The WLM Associated Service Class Summary page opens.
4. Click the Enclave tab.
The WLM Enclave page opens.

Viewing Server Statistics Overview

Purpose

The Server Statistics Overview helps you compare activity and related platform data across servers so that you can recognize problems.

Usage Overview

This feature helps you:

- Assess activity on servers and platforms.
- Set visual alerts to appear on the screen when resources pass what you determine an acceptable threshold.
- Easily jump to other relevant product features to continue isolating problems.

User Scenarios

Scenario 1: Investigating an unresponsive system

Your first line of support receives calls that some parts of the system are not responding. The support team goes to the Server Statistics Overview page and immediately sees that one server displays the red icon representing the “unavailable” status. The support team determines the unavailable server needs to be restarted, which will return the system to full functionality.

Scenario 2: Monitoring proactively

As the administrator of production systems, you have set appropriate thresholds for the fields displayed on the Server Statistics Overview page. During your regular monitoring you see that the Paging Rate threshold is being crossed. You know that the increase in paging rate probably means an increase in overhead. You can now increase memory, add servers, or take some similar course of action to keep production running smoothly.

To open the Server Statistics Overview page:

1. From the top navigation, click **Availability > Server Statistics Overview** or by selecting from the Tools button on the Server Overview page.

The Server Statistics Overview page opens.

Note: The Server Statistics Overview page shows information that helps track server performance. This page is similar to a work bench because you choose the servers to monitor. Using the left navigation, select the servers from your server farm that you want to monitor, and change them depending on your needs.

Note: In general, with a healthy system, Volume Delta is never erratic, while the Total Volume gradually increases. The JVM CPU% shows a snapshot of the CPU used by the JVM and the Total CPU% shows the total CPU used by the platform.

Note: When a server displays red, the Data Collector no longer sends a heart beat. This could mean the hardware is not functioning; the server is not functioning; the application server is not functioning; or the network between the server and WSAM is not functioning, or the Data Collector has stopped.

The Data Collector displays in blue when it is disabled by the user. It will not collect any performance and availability data from the WebSphere Application Server until it is enabled again.

When the data exceeds the threshold, the server name and the column will be highlighted in yellow.

Note: If WSAM shows a server is available, it doesn't necessarily mean that it is processing requests. For example, if in four refreshes, there is no change in the number of requests, but the CPU is always significant. The server may not be doing anything useful; it could be looping.

Note: A server that has been up for 30 days may need to be recycled in order to correct memory leaks. If you want to restart a server, you may want to select the server with the longest Uptime.

SERVER SELECTION

Group: CICS22M1 Server: All Servers Add Server(s)

SERVER DETAIL (Next Refresh in 6 Seconds)

Pause Refresh Customize... 20 per Page

1 - 3 of 3 Results 1

Name	Status	Platform	Volume	JVM/Region CPU Δ (ms)	Total Volume	JVM/Region CPU%	Total CPU%	JVM/Region (DSA,EDSA) Memory Usage (MB)	Group Name	IP Address	Uptime
ADCDPL M1L2 P390 CICS2QA3 ...	Unavailable	z/OS	0	0	0	0.00	0.00	0	CICS22M1	192.168.3.64	N/A
qaapp-aix-s01_node server1 92992 (L3)	Available	AIX	108	2,240	14,947	1.88	2.62	131	AIX Cluster	192.168.4.8	19:13:53
qaapp-aix-s03_node server3 40288 (L1)	Available	AIX	0	240	0	0.25	5.50	42	AIX Cluster	192.168.4.16	00:12:32

Clear All

1 - 3 of 3 Results 1

Unavailable Threshold Exceeded Disabled Δ = 15 seconds

Figure 18. Server Statistics Overview

To display groups or servers on the page:

1. From the left navigation, click the **All Servers** right-pointing arrow icon.
The list expands and collapses all the Groups in WSAM.
2. On the left navigation, click the plus sign to view the servers in a group.
The list expands and collapses the servers in the selected group.
3. Click on the arrow next to the group or server to populate the Detail window.

Note: For the z/OS platform, the icon on the left navigation represents a server instance, where each detailed line represents a server region which belongs to the selected server instance.

To remove a server from the Server Statistics Overview page:

1. Click the **X** icon next to the server.
The server disappears from the display.
2. If you want to clear all the servers from the display, click **Clear All** at the bottom of the list.

The page refreshes clear of any servers or information.

Note: For the z/OS platform, when you remove a server instance from the detail page, the system removes all the server regions belonging to that server instance from the display since the system treats them as a group of clones.

Configuring the Server Statistics Overview

The Server Statistics Overview page can be configured for each session. You can set the threshold for any of the following statistics:

- Status
- Platform
- Delta Volume
- Delta JVM/Region CPU Usage
- Total Volume
- JVM/Region CPU %
- Total CPU%
- Memory Usage
- Group Name
- IP Address
- Uptime
- Delta Platform CPU
- Start Time
- Data Collector Uptime
- Paging Rate
- Active Sessions
- Average Response Time (1 minute)
- Application Server Platform

Set the warning threshold for each resource to greater than or less than depending on when you want the count for the warning to begin.

To configure the Server Statistics Configuration page:

1. From the top navigation, click **Availability > Server Statistics Overview**.
The Server Statistics Overview page opens.
2. Click **Customize** in the control box.
The Server Statistics Configuration window opens.

SERVER STATISTICS CONFIGURATION			
<input type="checkbox"/> Volume Δ	<	1	request(s)
<input checked="" type="checkbox"/> Total Volume	>	1	request(s)
<input checked="" type="checkbox"/> Total CPU%	>	1	%
<input checked="" type="checkbox"/> Group Name			
<input checked="" type="checkbox"/> Uptime	>	33	hour(s)
<input checked="" type="checkbox"/> Start Time			
<input type="checkbox"/> Platform			
<input checked="" type="checkbox"/> Live Sessions	<	0	
<input type="checkbox"/> Application Server Platform			
<input type="checkbox"/> JVMRegion CPU Δ	>	0	ms
<input checked="" type="checkbox"/> JVMRegion CPU%	>	1	%
<input type="checkbox"/> JVMRegion (DSA,EDSA) Memory Usage	>	2	MB
<input type="checkbox"/> IP Address			
<input type="checkbox"/> Platform CPU Δ	None		ms
<input type="checkbox"/> Data Collector Uptime			
<input type="checkbox"/> Paging Rate	None		KB/s
<input type="checkbox"/> Average Response Time (1 min)	None		ms
<input type="checkbox"/> Auto-Refresh		15	sec

Figure 19. Server Statistics Configuration

3. Click **Select All** or click the individual check boxes to select the resource you want to display.
4. Select an operator from the drop-down menu. None means no threshold monitoring required.
5. Enter a threshold limit that will cause the system to generate a warning on the detail page.
6. Click **Save**.

Note: For each data element on the Server Statistics Configuration page, set the range between 0-99999.

After setting the thresholds, the system alerts you when a threshold is crossed by highlighting the column and the server name in yellow, while an unavailable server displays in red.

Also, for the z/OS environment, the threshold monitoring is applied to all of the server regions in the Server Statistics Overview page.

Chapter 7. In-Flight Request Search

Purpose

Use In-flight Request Search to improve your chances of locating a malfunctioning application in a server farm. In-flight Request Search provides a snapshot of the transactions in progress, showing you hanging transactions.

Usage Overview

This feature helps you:

- Locate hanging transactions.
 - Search for a specific transaction that you suspect is in progress.
 - Search for a transaction on a specific server or across multiple servers.
- Pinpoint specific transaction details.
 - Obtain additional information, such as a Stack Trace, Method Trace, or Session objects for suspicious transactions.
- Fix a hanging request by canceling it or changing its priority.
- Discover the relationships that a hanging request has with other requests, running on other application servers, in a composite transaction that spans multiple application servers.
- Easily jump to other relevant product features to continue isolating problems.

User Scenarios

Scenario 1: Investigating a hanging transaction

Customers call and complain they are having trouble completing transactions. You go to In-flight Request Search to locate a hanging transaction and, upon finding one, view a method trace for the transaction. You can see that the transaction is waiting for the return of a specific SQL call. You forward the method trace to a database administrator for further analysis.

Scenario 2: Isolating a problem with CPU utilization

After looking at the Server Statistics Overview page, you notice that CPU utilization is very high. You go to the In-flight Request Search to see if a transaction is present. It appears the system is churning on a transaction. Through a method trace, you suspect the transaction is looping. You forward the method trace to a developer for further analysis.

Searching for an Application Request

The In-Flight Request Search page lets you search for open, troubled requests in your server farm.

From the search results you can follow any request's Thread/Task ID link to view the Request Detail for that request. Click on any column heading to sort the search results by that column. Click the column heading again to reverse the sort. In addition, click

the Tools button to view the Server Activity Display page or the System Resources page for that server.

To search for a request:

1. From the top navigation, click **Problem Determination > In-Flight Request Search**.

The In-Flight Request Search page opens.

2. Select a Group or Server from the drop-down menu.

Note: If you do not select a group or server, the system will search for in-flight requests from all servers.

3. Enter a string in the Search Request box.

Note: The system will search all active URL strings (for Web requests) and active class names (for remote EJB requests) for the string entered in Step 2. If any request contains the string (Web requests or remote EJB requests), the results page will display those requests. In addition, if you leave the Search Request box empty, all in-flight requests will be displayed.

4. Click **OK**.

All the active requests associated with your search display in the order of descending Total Resident Time.

Note: If you know the group where the request is located, but not the server, select the group and then select all servers. If you don't have enough information to locate a request by string or key word, leave the Search Request field blank, and all the currently processing requests will display sorted in descending order by Total Resident Time. Using this method, you can search requests that have been active for a long period of time. The search is case insensitive and the results include the name of the class that makes up the remote EJB or URL. The results contain the string but may not match it exactly. All the results that contain the string will be displayed in the results table.

Sorting Search Results

You can sort your search results in alphabetical order according to the Server Name, by Client Request/Transaction, or in numeric order with Start Date/Time, Total Resident Time and User ID.

To sort the search results:

1. Click a column heading to sort the results. You can only sort by columns with underlined headings.
2. When the page refreshes, the results display sorted by the selected heading.
3. Click the column heading a second time to sort the results in reverse order.

Chapter 8. Server Activity

Purpose

The Server Activity Display helps you troubleshoot and fix hanging requests and evaluate the current performance of your applications.

Usage Overview

This feature helps you:

- Identify hanging requests.
- Fix a hanging request by canceling it or changing its priority.
- Isolate the particular method(s) or component(s) that cause a request to hang.
- Discover the relationships that a hanging request has with other requests, running on other application servers, in a composite transaction that spans multiple application servers.
- Get a flavor of what the most recently completed requests on a server were, along with their vital statistics.

User Scenarios

Scenario 1: Troubleshooting an application that hangs.

Several users of application Z have reported that they can't update their user preferences: application Z times out after a minute of not responding. You look for the application Z requests that have long resident times in the Active Requests tab of the Server Activity Display. View the Request Detail for one of these requests to determine why or where it is hanging.

Scenario 2: Understanding immediate workload.

While performing normal monitoring of your servers, you notice that a server's average response time has recently increased, with no appreciable change in throughput. You begin by looking at the Recent Requests tab of the Server Activity Display to see what the most recently completed requests have been on that server. You can see whether the requests are uniformly slow, or if there is variation among requests; this may help you isolate whether it is a problem with the server (uniformly slow), or with an application (certain requests are slow). You can see whether the slow requests are CPU-heavy, or if they are spending too many moments idle.

Server Activity Display

The Server Activity Display is useful when you are looking for a problematic transaction, one that is looping, hanging or slow. If the Server Activity Display shows a number of threads waiting, your system maybe running slowly or overloaded.

The Server Activity Display page has two tabs: Active Requests and Recent Requests. Each of these tabs has summary information in the Server Info and Recent Activity Sections.

Active Requests

The Active Requests tab presents information about requests that are still active.

Click the name of the request in the Client Request column to review more detailed information about the request, including a Stack Trace, Method/Component Trace and the Request/Session Object. You may email or export the Method/Component Trace.

Click the Composite Request Indicator to investigate the relationship of a request to requests on other application servers. See “Viewing Composite Requests” on page 145 for more information about using Composite Request features.

Note: Data in this section is constantly fluctuating. Active Requests displays a snapshot of the data. Therefore, requests may complete and disappear from the display upon refresh, or by the time you drill down.

To open the Server Activity Display (Active Requests) page:

1. From the top navigation, click **Problem Determination > Server Activity Display**.

The Server Activity Display Server Selection page opens.

2. Select a group and a server from the drop-down menu.

The Server Activity Display (Active Requests) page opens. The information for the selected server group displays.

Note: Threads perform the work for the server. The Active Requests page shows all the threads running within the monitored application server. To see threads outside of the application server, use the JVM Thread Display.

Note: To access the Memory Analysis Report, click the links from the Recent Activity (Last Minute) table. To go to the JVM Thread Display page or the System Resources page, click the tools button on the Server Info table.

Note: Use Idle Time as a measure of whether the selected request has been idle longer than other requests. It may be useful to break down idle requests into the components that cause a request to be idle, by using the Method/Component Trace.

Note: In a normal environment, data will pass through the system quickly and may not be accessible by the Server Activity Display. Transactions in a smoothly running environment are processed efficiently, and requests may execute so quickly that a snapshot does not capture any active requests.

Note: From the Request Detail page, you can check CPU Utilization and Resident Time. For instance, eighty percent utilization may be considered high and could explain why resident time is taking longer or why things are running slowly.

Note: Requests that participate in a Composite Request are identified by the Composite Request icon. Click on this icon to see the Composite Request View of the Method Trace.

SERVER INFO		RECENT ACTIVITY (Last Minute)	
Snapshot Date	Sep 15, 2004	Application Server Name	server1
Snapshot Time	2:14:02 PM	Application Server IP Address	192.168.4.8
Platform CPU % Utilization	12.38%	Total Thread Count	5
JVM CPU	5.75%	JVM Heap Size (MB)	168
# of Requests	334	Avg. Response Time (ms)	545
# of Live Sessions	931		

ACTIVE REQUESTS										
Filter By	Thread Type	Thread Status	Refresh							
Client Request	Client Request Start	Thread ID	Resident Time (ms)	Accumulated CPU (ms)	Idle Time (ms)	Thread Status	Last Known Class	Last Known Method	Last Known Action	User ID
/trade/scenario	Sep 15, 2004 2:13:56 PM	1495864064	9054	0.000	9054	Waiting	N/A	N/A	JCA Request	N/A
/trade/scenario	Sep 15, 2004 2:13:57 PM	1495539584	7162	10.000	7152	Runnable	N/A	N/A	JCA Request	N/A
/trade/scenario	Sep 15, 2004 2:13:59 PM	1552454144	5903	20.000	5883	Runnable	N/A	N/A	JCA Request	N/A
/trade/scenario	Sep 15, 2004 2:13:59 PM	1533192448	5483	10.000	5473	Runnable	N/A	N/A	JCA Request	N/A
/trade/scenario	Sep 15, 2004 2:14:00 PM	1508505856	5064	0.000	5064	Runnable	N/A	N/A	JCA Request	N/A

Figure 20. Server Activity Display (Active Requests)

To filter the Active Requests data:

1. Select a filter from Thread Type and/or Thread Status from the drop-down menus.
2. Click **Refresh**.

The Active Requests data displays based on the selected filter.

To sort the Active Requests data:

1. Click a heading link: **Client Request**, **Resident Time**, **Accumulated CPU**, **Idle Time**, **Thread Status**, or **Last Known Class Name**, **Last Known Method**, **Last Known Action** or **User ID**.

The data refreshes sorted by the selected heading.

2. Click the heading link a second time to invert the sorting.

Recent Requests

See the most recently completed requests for a server on the Recent Requests tab of the Server Activity Display page.

The default maximum number of completed requests in the recent activity data is 100. If you want to view more completed requests, refer to the Managing Server and Data Collector Tuning chapter in the Operator’s Guide.

Note: The maximum number of requests in the recent activity data applies to each server. When the queue is full, the newest request data will replace the oldest data.

To open the Server Activity Display (Recent Requests) page:

1. From the top navigation, click **Problem Determination > Server Activity Display**.

The Server Activity Display Server Selection page opens.

2. Select a group and a server from the drop-down menu.

The Server Activity Display (Active Requests) page opens. The information for the selected server group displays.

3. Click **Recent Requests**.

The Recent Requests tab opens displaying the 100 most recently completed requests.

SERVER SELECTION					
Group	AIK Cluster	Server	qaapp-aix:01_node.server1.92992 (L3)		
Active Requests		Recent Requests			
SERVER INFO		RECENT ACTIVITY (Last Minute)			
Snapshot Date	Sep 15, 2004	Application Server Name	server1		
Snapshot Time	2:17:30 PM	Application Server IP Address	192.168.4.8		
Platform CPU % Utilization	4.25%	Total Thread Count	100		
		JVM CPU	3.00 %		
		JVM Heap Size (MB)	145		
		# of Requests	478		
		Avg. Response Time (ms)	180		
		# of Live Sessions	775		
RECENT REQUESTS					
Filter By	Thread Type	Any			
Refresh					
Client Request	Client Request Start	Response Time (ms)	Accumulated CPU(ms)	Idle Time (ms)	User ID
/trade/scenario	Sep 15, 2004 2:17:21 PM	322	80.000	242	N/A
/trade/scenario	Sep 15, 2004 2:17:23 PM	262	90.000	172	N/A
/trade/scenario	Sep 15, 2004 2:17:22 PM	208	40.000	168	N/A
/trade/scenario	Sep 15, 2004 2:17:18 PM	171	30.000	141	N/A
/trade/scenario	Sep 15, 2004 2:17:18 PM	170	10.000	160	N/A
/trade/scenario	Sep 15, 2004 2:17:29 PM	154	40.000	114	N/A
/trade/scenario	Sep 15, 2004 2:17:22 PM	149	20.000	129	N/A

Figure 21. Server Activity Display (Recent Requests)

To filter the Recent Requests Data:

1. Select a filter from the Thread Type drop-down menu.
2. Click **Refresh**.
The recent request data displays based on the selected filter.

To sort the Recent Requests Data:

1. Click a heading link: **Client Request**, **Client Request Start**, **Resident Time**, **Accumulated CPU**, **Idle Time** or **User ID**.
The data refreshes sorted by the selected heading.

Viewing Request Detail

The Request Detail page provides data for one selected request. Typically, you arrive on this page by clicking a name in the Client Request column of the Server Activity Display (in the Active Requests tab.)

Through the left navigation of the Request Detail page, you can obtain a Stack Trace, Method/Component Trace or view the Request/Session Object. If necessary, you can cancel a request and change the thread's priority or status.

To open the Request Detail page:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
4. Click the link in the Client Request column.
The Request Detail page for that request opens. This page displays data for that request only.

REQUEST PROPERTIES			
Snapshot Date	Sep 15, 2004	Application Server Name	server1
Snapshot Time	2:21:13 PM	Application Server IP Address	192.168.4.8
Platform CPU % Utilization	19.88%	Total Thread Count	8

REQUEST DETAIL			
Thread ID	1533192448	Last Known CPU	10.000 ms
Client Request	/trade/scenario	Idle Time	9727 ms
Client Request Start Date	Sep 15, 2004	Thread Type	Servlet
Client Request Start Time	2:21:05 PM	Last Known Class	N/A
Resident Time	9737 ms	Last Known Method	N/A
User ID	N/A	Last Known Action	JCA Request
Priority	5	Thread Status	Waiting
Change Priority	<input type="text" value="No Change"/>	Change Thread Status	<input type="text" value="No Change"/>

Figure 22. Request Detail

Note: The Request Detail page allows you to take action on a request. To obtain further details on the request, use the left navigation to view a Stack Trace, Method/Component Trace or the Request/Session Object

- The Stack Trace shows the outstanding methods from a request that has not yet finished executing.
- The Method/Component Trace shows all method and component events that have executed for the specific request.
- The Request/Session Object provides the contents for the requests and session objects, including Session Creates, Time and Last Access Time.

Suspending a Thread

An executing thread is active, and a paused thread is suspended. You may want to suspend a thread if you suspect there is a problem with it and want to uncover the cause.

To suspend a thread:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
4. Click the link in the Client Request column.
The Request Detail page for that request opens.
5. From the Change Thread Status drop-down menu, select **Suspend**.

6. Click **OK**.

Note: When suspending a thread, there is a danger that the request may hold database locks or system resources. After you suspend the request, any other requests that require the removal of those locks or monitors will also be suspended.

Any locks in the application server and database server will not be released after the system suspends a thread. This can cause other applications to fail or hang.

Activating a Thread

A thread is executing if it is active, and the thread is paused when it is suspended. Select **Active** status to re-activate a suspended thread.

To activate a thread:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
4. Click the link in the Client Request column.
The Request Detail page for that request opens.
5. From the Change Thread Status drop-down menu, select **Active**.
6. Click **OK**.

Canceling a Request

If an application is looping or abusing resources, it may be necessary to cancel the request. This will terminate the request by throwing a run-time exception.

To cancel a request:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
4. Click the link in the Client Request column.
The Request Detail page for that request opens.
5. Click **Cancel Request**.
A confirmation box displays.
6. At the confirmation box, click **OK**.
The system terminates the request.

Note: Upon examination of the Request Detail page, you may discover that a thread is misbehaving. For example, it might be looping or sleeping while holding a lock thereby preventing other requests from proceeding. In such cases, you may decide to cancel the request (which occurs by throwing an exception.)

You may experience unexpected side effects when you cancel a request, due to multiple requests sharing data, and leaving the data in an inconsistent state. Use the Cancel Request function only when you are sure it is safe to do so. In addition, whether or not a request can be canceled—based upon the state of the thread—is a decision eventually made by the JVM.

Changing a Thread's Priority

If a thread is executing too slowly, you can increase the thread's priority. This will move the thread up in the stack so it will execute more quickly. Alternatively, you can decrease a thread's priority to allow other threads to execute more quickly.

To change a thread's priority:

1. From the top navigation, click **Problem Determination > Server Activity Display**.

The Server Activity Display Server Selection page opens.

2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.

The Server Activity Display (Active Requests) page opens.

4. Click the link in the Client Request column.

The Request Detail page for that request opens.

5. From the Change Priority drop-down menu, select a priority.

Priority 1 is the lowest and priority 10 is the highest.

6. Click **Save**.

Note: When changing a thread's priority, be aware that the new priority remains for the life of the thread. As a result, any new requests issued after the change will hold that priority even though the new priority was not intended for the new request.

Viewing the Request Object and Session Object

The Request/Session Object page lists information for the current request object and session object.

To view the Request Object and Session Object trace:

1. From the top navigation, click **Problem Determination > Server Activity Display**.

The Server Selection page opens.

2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.

The Server Activity Display page opens.

4. Click the link in the Client Request column.

The Request Detail page for that thread opens.

5. Click **Request/Session Object** from the left navigation.

The Request Object and Session Object page opens.

Note: In many applications, state information is stored in the request object and the session object. WSAM provides the ability to view the content on the requests and session objects, as well as, Session Creates Time and Last Access Time. The

developer responsible for investigating a misbehaving request may find this information useful when resolving issues.

REQUEST/SESSION OBJECT PROPERTIES			
Snapshot Date	Sep 15, 2004	Application Server Name	server1
Snapshot Time	2:31:28 PM	Application Server IP Address	192.168.4.2
Platform CPU % Utilization	5.50%	Total Thread Count	3
REQUEST/SESSION OBJECT			
Remote IP	192.168.4.102		
Remote Host	192.168.4.102		
Request URL	http://qaapp-sun-s01.cyanea.com:7001/cyanea_one/testware/consumeConnPool?ttl=2410&consumeTime=10&reqname=ConsumeConnectionPool&appname=ConsumeConnectionPool		
Method	GET		
User ID	N/A		
Session ID	BLq4R8vKFQ41JRVJDTpQsM41sNjTmhvgCX9BGQcY9g8FPdfTzLTmI1166361828I1095281400149		
Session Size	0 KB		

Figure 23. Request Object and Session Object

Viewing a Stack Trace

The Stack Trace page displays a list of method calls, starting with the method being executed when the stack trace was requested, in last-in first-out order. For each method, the list includes the Class Name, Method Name and (optionally) a line number.

To view a stack trace:

1. From the top navigation, click **Problem Determination > Server Activity Display**.

The Server Activity Display Server Selection page opens.

2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.

The Server Activity Display (Active Requests) page opens.

4. Click the link in the Client Request column.

The Request Detail page for that request opens.

5. Click **Stack Trace** from the left navigation.

The Stack Trace page opens. The most recently executed method displays first in the Stack Trace.

Note: The Stack Trace shows the outstanding methods to be completed as a result of the request. This trace reports the data unfiltered so you will see every class. In

a normal environment, a request executes quickly so it may be difficult to catch a stack trace before completion. This is meant for troubleshooting a hanging request.

STACK TRACE PROPERTIES			
Snapshot Date	Sep 15, 2004	Application Server Name	server1
Snapshot Time	2:35:37 PM	Application Server IP Address	192.168.4.2
Platform CPU % Utilization	3.00%	Total Thread Count	11

STACK TRACE					
Depth	0	Class	java.lang.Thread	Method	sleep
Depth	1	Class	com.testware.web.consumeConnPool.ConsumeConnPoolAction	Method	processRequest
Depth	2	Class	com.testware.web.framework.ControllerServlet	Method	doAction
Depth	3	Class	com.testware.web.framework.ControllerServlet	Method	doGet
Depth	4	Class	javax.servlet.http.HttpServlet	Method	service
Depth	5	Class	javax.servlet.http.HttpServlet	Method	service
Depth	6	Class	weblogic.servlet.internal.ServletStubImpl\$ServletInvocationAction	Method	run
Depth	7	Class	weblogic.servlet.internal.ServletStubImpl	Method	monitored_original\$invokeServlet
Depth	8	Class	weblogic.servlet.internal.ServletStubImpl	Method	invokeServlet
Depth	9	Class	weblogic.servlet.internal.ServletStubImpl	Method	invokeServlet
Depth	10	Class	weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction	Method	run
Depth	11	Class	weblogic.security.service.SecurityServiceManager	Method	runAs

Figure 24. Stack Trace

Viewing a Method/Component Trace

About the Method/Component Trace

The Method/Component Trace presents the path of execution for a request, in several formats.

The data available in the Method/Component Trace depends on the Monitoring on Demand level of the Data Collector, and the thread status.

Monitoring on Demand Level

The Monitoring on Demand level of the Data Collector determines the level of depth of data of the Method Trace:

- Full method traces are available at L3, which include events from both nested request components and application methods.
- Traces at L2 have events from nested request components, but not from application methods.
- Traces at L1 have only top-level JSP or Servlet calls.

Thread Status

The Method/Component Trace may represent complete or incomplete requests:

- For incomplete requests, the Method/Component Trace has two tabs: Flow View and Search. These tabs present the trace in terms of entry and exit events.
- For completed requests, the Method/Component trace has tabs in addition to Flow View and Search: Nesting Summary and Drilldown View. These tabs present the trace in terms of completed events, rather than raw entry/exit events.

(Traces of incomplete requests come through Server Activity Display, In-Flight Request Search, or when collected through in-flight Trap Actions. Traces of complete requests (historical data) come through Performance Analysis & Reporting or through trap actions based on completed requests.)

Using the Flow View

The Flow View lists the method flow of the current request, in terms of the method/component entry and exit events, in the order of last-in first-out.

- Identify entry and exit events by method/component name in the Event Data column, where they are, by default, listed in order of execution.
- Identify the time of execution and the CPU time consumed for each event, relative to the start of the request.
- Identify the relative level of nesting, Depth describes the nesting of methods: depth increases when a new method call begins, and depth decreases when a method ends.
- Identify whether an event is from an application method or a call to a nested request (component) by looking at the Event Type column.

Note: You can choose how many methods per page you want to see using the pagination drop-down menu. Quickly locate aberrant methods using the Threshold Highlighter, as described in “Threshold Highlighter”.

Note: You can sort the data according to the values in any column by clicking the column name once.

Note: The entire Flow View can be exported to a file, viewed as a PDF, or emailed as a PDF for further analysis by clicking the icons next to the pagination drop-down menu.

Threshold Highlighter

In order to help you locate the problem spots, there are two columns that work with the Threshold Highlighter tool. Together, they break out the contribution of response time and CPU time consumed between each two consecutive events: Delta Elapsed Time and Delta CPU Time.

To quickly scan for events whose delta values are exceptional, specify thresholds for each of these columns in the Threshold Highlighter. Once you apply these thresholds, any event that has a value that exceeds one or both of the thresholds you set will be shown in bold.

To view the Flow View of a method trace:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
4. Click the link in the Client Request column.

The Request Detail page for that request opens.

5. Click **Method/Component Trace** from the left navigation.

The Method/Component Trace page opens. The method that has executed most recently appears first in the Method/Component Trace.

METHOD/COMPONENT TRACE PROPERTIES

Application Server Name	server1	Request	http://qaapp-sun-s01.cyanea.com:7001/cyanea_one/testware/consumeConnPool
Execution Start Time	Sep 15, 2004 2:40:04 PM	Request Type	Servlet
Resident Time (ms)	218772	CPU Time (ms)	6.783

Nesting Summary | Drilldown View | Flow View | Search

Threshold Highlighter

Elapsed Time >= (ms) CPU Time >= (ms)

COMPLETE FLOW VIEW (appServer_linux_server_no02)

1 - 3 of 3 Results

Depth	Event Type	Event Data	Elapsed Time (ms)	CPU Time (ms)	Δ Elapsed Time (ms)	Δ CPU Time (ms)
0	Servlet Entry	/cyanea_one/testware/consumeConnPool?ttl=360&consumeTime=30&reqname=ConsumeConnectionPool&appName=ConsumeConnectionPool	0	0	0	0
1	JNDI Entry	Provider URL:local:// Lookup Name:jdbc/CyaneaDataSource	**5**	**5.215**	5	5.215
1	JNDI Exit	Provider URL:local:// Lookup Name:jdbc/CyaneaDataSource	6	6.783	1	1.568

1 - 3 of 3 Results

Figure 25. Method/Component Trace

6. Enter the Delta Elapsed Time and the Delta CPU Time value under the Threshold Highlighter table to highlight data you want to view throughout the whole trace.
7. Click **Apply**.

The Complete Flow View table displays the method flow list with the highlighted data that you selected to view.

8. Click **Reset to Default** for using the default threshold highlighter value, if necessary.
9. Click to select the number of rows of data that you want to view per page, from the Pagination drop-down menu.

Note: The Flow View tab refreshes displaying the number of rows of data you selected to view on each page.

Note: The Method/Component Trace shows the path of execution for a request. The data displays how the request arrived at the current point in its execution. Additionally, the trace may reveal looping behavior: When you see excessive,

repeated entry and exit records for the same method, it could indicate the method is caught in a loop.

Note: WSAM shows the cumulative CPU time and the cumulative elapsed (wall clock) time for each event. In other words, you can compare the resources consumed by the CPU and the elapsed time while going from entry to exit, or entry to entry, or exit to exit, or exit to entry by subtracting the CPU time of the previous trace record from that of the current one. Using this information, you can ascertain which methods are taking the longest to execute and/or using large amounts of CPU.

Emailing a PDF File

You can email a PDF file of the Method Trace/Component Trace to one, or a group of WSAM users. Separate multiple addresses with a comma. Recipients must have valid user accounts and proper permissions in order to view the report.

To email a PDF file:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display page opens.
4. To view the detail, click the link in the Thread ID column.
The Request Detail page for that thread opens.
5. Click **Method /Component Trace**.
The Method/Component Trace page opens.
6. Click **Email**.
The email page opens.
7. Enter the email address of the recipient. Separate multiple addresses with a comma.
8. Click **OK**.

Viewing a PDF File

Before emailing the PDF file, you may view the file by downloading it.

To view a PDF file:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display page opens.
4. Click the link in the Thread ID column.
The Request Detail page for that thread opens.
5. Click **Method/Component Trace**.
The Method/Component Trace page opens.
6. Click **View PDF**.

7. From the File Download window, click either **Open** to view the file immediately or click **Save** to download the file.

Exporting to a File

You may export the trace data to a comma-delimited file format.

To export to a file:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display page opens.
4. Click the link in the Thread ID column.
The Request Detail page for that thread opens.
5. Click **Method/Component Trace**.
The Method/Component Trace page opens.
6. Click the **Export to File** button.
7. Click either **Open** to view the file immediately or click **Save** to download the file.

Searching a Method/Component Trace

The Search allows you to specify any of the columns available in the Flow View (Elapsed Time, CPU Time, Delta Elapsed Time, Delta CPU Time, Event Type or Event Data), together with a numerical threshold (or a string), and presents a list of events from the method trace whose metrics cross the threshold (or match the string).

Note: The Event Type and Event Data searches are case sensitive.

Results are sorted in descending numerical order, or alphabetically, depending on the choice of metric. You can sort the results by any column by clicking the column name once.

Since the metrics are only partly related to the events (they describe relationships among adjacent events), the search is an aid to locating problem spots. The real problem may be either before or after the event that stands out. Therefore, it is important to investigate the context of problem methods or API calls before escalating your findings.

To search a method trace:

1. From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
2. Select a group from the Group drop-down menu.
3. Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
4. Click the link in the Client Request column.
The Request Detail page for that request opens.
5. Click **Method/Component Trace** from the left navigation.

The Method/Component Trace (Flow View) page opens. The method that has executed most recently appears first in the Method Trace.

6. Click the **Search** tab.

The Search tab opens.

Depth	Event Type	Event Data	Elapsed Time (ms)	CPU Time (ms)	Elapsed Time (ms)	CPU Time (ms)
0	Servlet Entry	/cyanea_one/teststeware/consumeConnPool?ttl=360&consumeTime=30&reqname=ConsumeConnectionPool&appname=ConsumeConnectionPool	0	0	0	0
1	JNDI Entry	Provider URL:local:// Lookup Name:jdbc/CyaneaDataSource	5	5.215	5	5.215
1	JNDI Exit	Provider URL:local:// Lookup Name:jdbc/CyaneaDataSource	6	6.783	1	1.568

Figure 26. Method/Component Trace: Search

7. Enter the search criteria and the search value.
8. Click **Search**.

The Method Trace page refreshes and displays the results of your search.

Note: Clicking the result in the Event Data column opens the Flow View tab to the corresponding line. For example, if the first result in the Search tab is the twentieth method on the Flow View page, then clicking the Event Data link of the first result will bring up the Flow View tab starting with the page that includes the twentieth record.

Additional Method/Component Trace Features for Completed Requests

For traces of completed requests, there are two additional tabs within the Method/Component Trace: Nesting Summary and Drilldown View.

These features present round-trip calculations from corresponding entry and exit events, rather than presenting the raw event data, and allow you to navigate logically through the trace.

Nesting Summary

The Nesting Summary allows you to quickly identify problems with external resources used by a request.

If you suspect the problem with a request is because of an external resource, rather than the application code itself, the Nesting Summary may help you identify the source of the problem.

Compare the number of calls in each category of nested request components (the J2EE API calls), as well as the average response time and the average CPU time of each category. (The L1 traces show top-level JSP and servlet events only.)

Evaluate the 10 slowest nested request components in the 10 Slowest Components section. Jump to the Drilldown View of any component, to see the immediate context of the call, by clicking the component name in the Event Data column.

Drilldown View

Toggle between two options within the Drilldown View using the drop-down: Depth Drilldown Detail and Depth Drilldown Report.

Depth Drilldown Detail

Navigate through the trace one level at a time using the Depth Drilldown Detail.

A single page in the Depth Drilldown Detail represents a single method, and provides a chronological list of the calls made by that method (and the corresponding exits.) In addition, a summary for the method is provided in terms of Depth, Resident Time and CPU Time.

See if methods/components have children by looking at the values in the Number of Children column.

Drill down to child calls by clicking on the name of the method/component in the Event Data column. Move up to parent calls by using the Up One Level drop-down.

A Threshold Highlighter, similar to the one described for “Using the Flow View” on page 64, operates on the Resident Time and CPU Time of the methods/components: methods/components that exceed your thresholds appear in gold.

To view the Depth Drilldown Detail of a method trace:

1. From the top navigation, click **Problem Determination > Server Activity Display**.

The Server Activity Display Server Selection page opens.

2. Select a group from the Group drop-down menu.

3. Select a server from the Server drop-down menu.

The Server Activity Display (Active Requests) page opens.

4. Click the link in the Client Request column.

The Request Detail page for that request opens.

5. Click **Method/Component Trace** from the left navigation.

The Method/Component Trace page opens.

6. Click the Drilldown View tab.

The Drilldown View page opens to the Depth Drilldown Detail.

DEPTH DRILLDOWN REPORT				
SUMMARY				
Depth: 0 Resident Time (ms): 0 CPU Time (ms): 0				
NESTING REPORT				
		Servlet	EJB	JDBC
TOTAL		2	3	1
AVERAGE RESPONSE TIME (ms)		1,018	338	1
AVERAGE CPU TIME (ms)		20	3,333	0
5 SLOWEST RESPONSE TIME				
Rank	Depth	Event Type	Event Data	Response Time (ms)
1	1	Servlet	/cyanea_one/testware/dsStateless?ttl=1&sqlStatement=1&lookup	1,018
2	2	EJB	EJB Name: com.testware.ejb.dsStateless.TextstringStatelessBean Method: sleepInTx	1,008
3	2	EJB	EJB Name: com.testware.ejb.dsStateless.TextstringStatelessBean Method: findTextstringAll	6
4	3	JDBC	Data Source Name:Ga Type 2 Data Source SQL Statement:select * from textstring	1
5	2	EJB	EJB Name: com.testware.ejb.dsStateless.TextstringStatelessBean Method: create	0
5 SLOWEST CPU TIME				
Rank	Depth	Event Type	Event Data	CPU Time (ms)
1	1	Servlet	/cyanea_one/testware/dsStateless?ttl=1&sqlStatement=1&lookup	20
2	2	EJB	EJB Name: com.testware.ejb.dsStateless.TextstringStatelessBean Method: findTextstringAll	10
3	2	EJB	EJB Name: com.testware.ejb.dsStateless.TextstringStatelessBean Method: create	0
4	3	JDBC	Data Source Name:Ga Type 2 Data Source SQL Statement:select * from textstring	0
5	2	EJB	EJB Name: com.testware.ejb.dsStateless.TextstringStatelessBean Method: sleepInTx	0

Figure 27. Method/Component Trace: Depth Drilldown Detail

Depth Drilldown Report

A second feature of the Drilldown View is the Depth Drilldown Report. Use the Depth Drilldown Report to quickly identify problems with categories of nested request components used by a method and its children, by comparing the number of calls, Average Response Time and Average CPU Time.

Note: The Nesting Summary section of the Depth Drilldown Report includes the nested request component calls made by a method and its children.

Evaluate the five slowest and most CPU-intensive nested request component calls (made by the method or its children) in the 5 Slowest Response Time and 5 Slowest CPU Time sections.

Jump to the Depth Drilldown Detail of any component, to see the immediate context of the call, by clicking the component name in the Event Data column of either of the two 5 Slowest sections.

To view the Depth Drilldown Report of a method trace:

- From the top navigation, click **Problem Determination > Server Activity Display**.
The Server Activity Display Server Selection page opens.
- Select a group from the Group drop-down menu.
- Select a server from the Server drop-down menu.
The Server Activity Display (Active Requests) page opens.
- Click the link in the Client Request column.
The Request Detail page for that request opens.
- Click **Method/Component Trace** from the left navigation.
The Method/Component Trace page opens.
- Click the **Drilldown View** tab.
The Drilldown View page opens to the Depth Drilldown Detail.
- Choose Depth Report from the drop-down menu.
The Depth Drilldown Report opens.

Chapter 9. Recent Activity

Purpose

Use Recent Activity to discover problems related to memory or other resources.

Usage Overview

This feature helps you:

- Identify JVM-related issues.
- Recognize when memory-related problems are compromising other parts of the system or other resources.
- Identify which resource is limiting recent performance.

User Scenarios

Scenario 1: Evaluating the impact of garbage collection

You suspect that frequent garbage collection calls are affecting the performance of a server, so you go into Recent Activity and set up the first graph to display the Number of Garbage Collections metric for the last 48 hours. In the second graph, you roll through the different metrics possibly affected by frequent garbage collection.

Notes

Note: The Garbage Collection option is not supported for either CICS or IMS.

Creating a Recent Activity Report

Use Recent Activity when you need to investigate potential memory problems relating to garbage collection and the JVM heap size. At times garbage collection may not clean up properly or the heap may have too little memory allocated.

To create a Recent Activity report:

1. From the top navigation, click **Availability > Recent Activity Display**.
The Recent Activity page opens.
2. Select a Group and a Server from the drop-down menus.
3. For Metric 1 and Metric 2, select the two metrics you want to compare from the drop-down menus.
4. For time, select the time when you want the system to extract the data.
Data is also aggregated by the time frame.
The Recent Activity report displays.



Figure 28. Recent Activity report

- Using the Recent Activity Options, you can select a different group or server, compare two different metrics, or view a different time increment.

A new report displays based on your selection. You can click on the bars in the graph for further details.

Note: Either heap size or garbage collection can cause a slow down in your server's performance. Find out if your heap size is too small for the number of users using the system or too small for the current workload on the system. At times, garbage collection can cause high JVM CPU usage, slow transaction time, or a delay that impacts throughput. Analyze the memory in your system using the Recent Activity Display and then make the necessary adjustments.

Chapter 10. Memory Diagnosis

Purpose

Gain insight into the JVM's heap and memory information through Memory Diagnosis. Use this information to tune the JVM parameters, assess your resources and find evidence of memory leaks.

Usage Overview

This feature helps you:

- View an analysis of the heap and make adjustments to the JVM parameters based on your findings.
- Evaluate the resources being used by your system and make capacity planning decisions.
- Uncover memory leaks and find the classes that are memory leak candidates.

User Scenarios

Scenario 1: Detecting a memory leak

After creating a Memory Analysis report that compares JVM Heap Size to Average Response Time, you think there is a memory leak. Access the Memory Leak feature to see if the amount of uncollected memory is increasing. You set up a candidate for the server in question. This tells the system to collect heap data now and again after a specified amount of time. Then you can compare the heap data for the two periods of time to determine if there is evidence of a memory leak.

Scenario 2: Supporting your claim that the purchase of new servers is necessary

The year end budget is due and you need to project whether you will need to buy more servers for your environment. You create a Memory Analysis report during peak usage and compare JVM Heap Size to the Number of Sessions. The number of servers is close to maxing out the current environment. As a capacity planner, you recommend that the company increase the number of servers currently servicing the environment.

Notes

Note: The Memory Analysis (Garbage Collection) option is not supported for CICS or IMS. The Memory Analysis (Java Heap Size option) is not available for IMS. The Heap Analysis and Memory Leak features are not available for CICS or IMS.

Memory Analysis

Creating a Memory Analysis Report

Use Memory Analysis when you need to investigate potential memory problems relating to garbage collection and the JVM heap size. At times, garbage collection may not cleanup properly, or the heap may have allocated insufficient memory.

To create a Memory Analysis report:

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Analysis**.
The Memory Analysis page opens.
2. Select a Group and a Server from the drop-down menus.
3. Select the Analysis Type: Garbage Collection or Java Heap Size.
4. Click **Next**.
5. In the Metric Selection, select the option that contains the two metrics you want to compare.
6. Click **View Results**.

The Memory Analysis report displays.

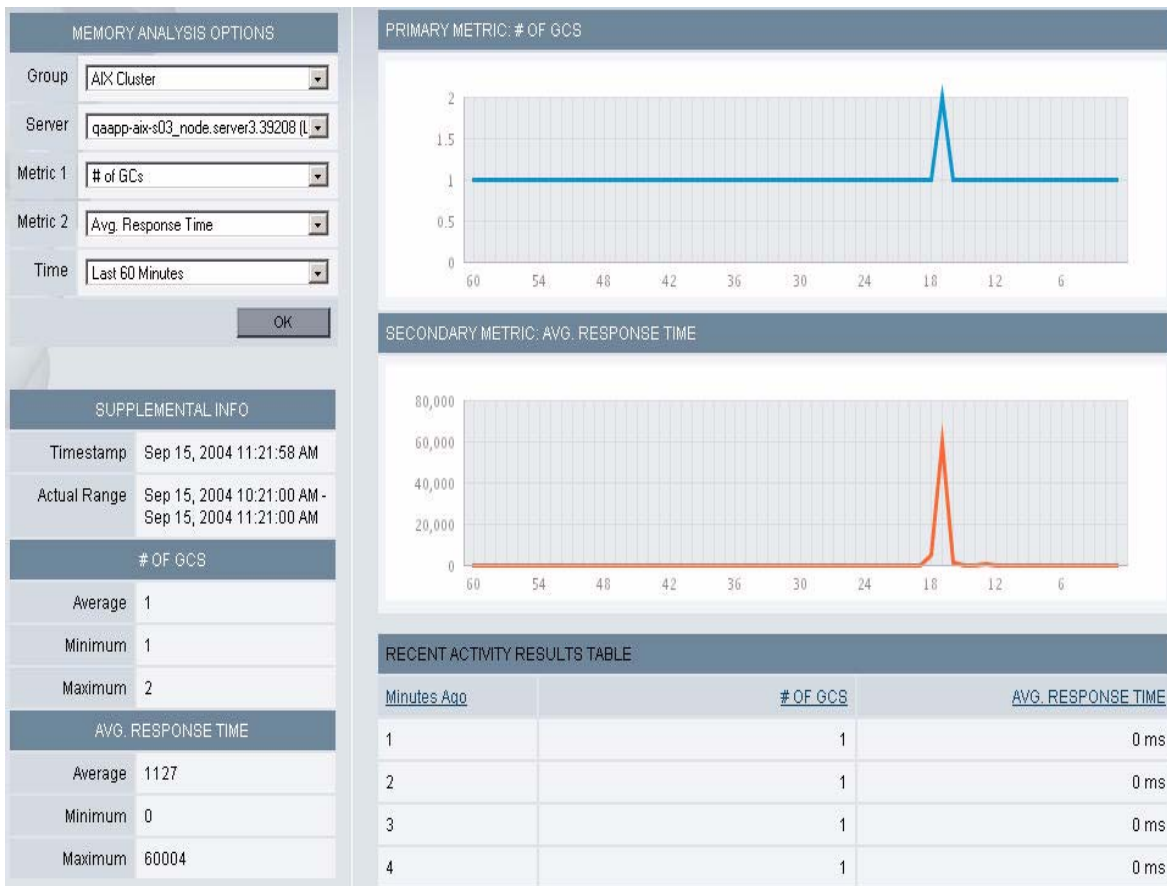


Figure 29. Memory Analysis Report

7. Using the Memory Analysis Options, you can select a different group or server, compare two different metrics or view a different time increment.

A new report displays based on your new selections.

Note: When there is over 24 hours of data, your reports will show the last 48 hours. In all other cases, the last 60 minutes of data will display.

Note: Either heap size or garbage collection can cause a slow down in your server's performance. Find out if your heap size is too small for the number of users using the system or too small for the current workload on the system. At times, garbage collection can cause high JVM CPU usage, slow transaction response time, or a delay that impacts throughput. Analyze the memory in your system using Memory Analysis and then make the necessary adjustments.

Heap analysis

Setting up a Heap Analysis

Query a server and learn how the server uses the heap memory. The system takes a snapshot of the heap and breaks the data down by class name. Additionally, you have the option to force a full garbage collection prior to taking a snapshot of the heap.

Note: Using this function may cause a significant effect on system performance, especially if you have a large heap. In addition, with the Sun JDK only, the heap analysis report may take a long time to generate.

To set up a Heap Analysis:

1. Click **Problem Determination > Memory Diagnosis > Heap Analysis**.
The Heap Analysis page opens.
2. Select a Group and a Server.
3. Select **Yes** or **No** to perform a garbage collection on the heap prior to the Heap Analysis snapshot.
4. Click **OK**.

The Heap Analysis results display in the same window.

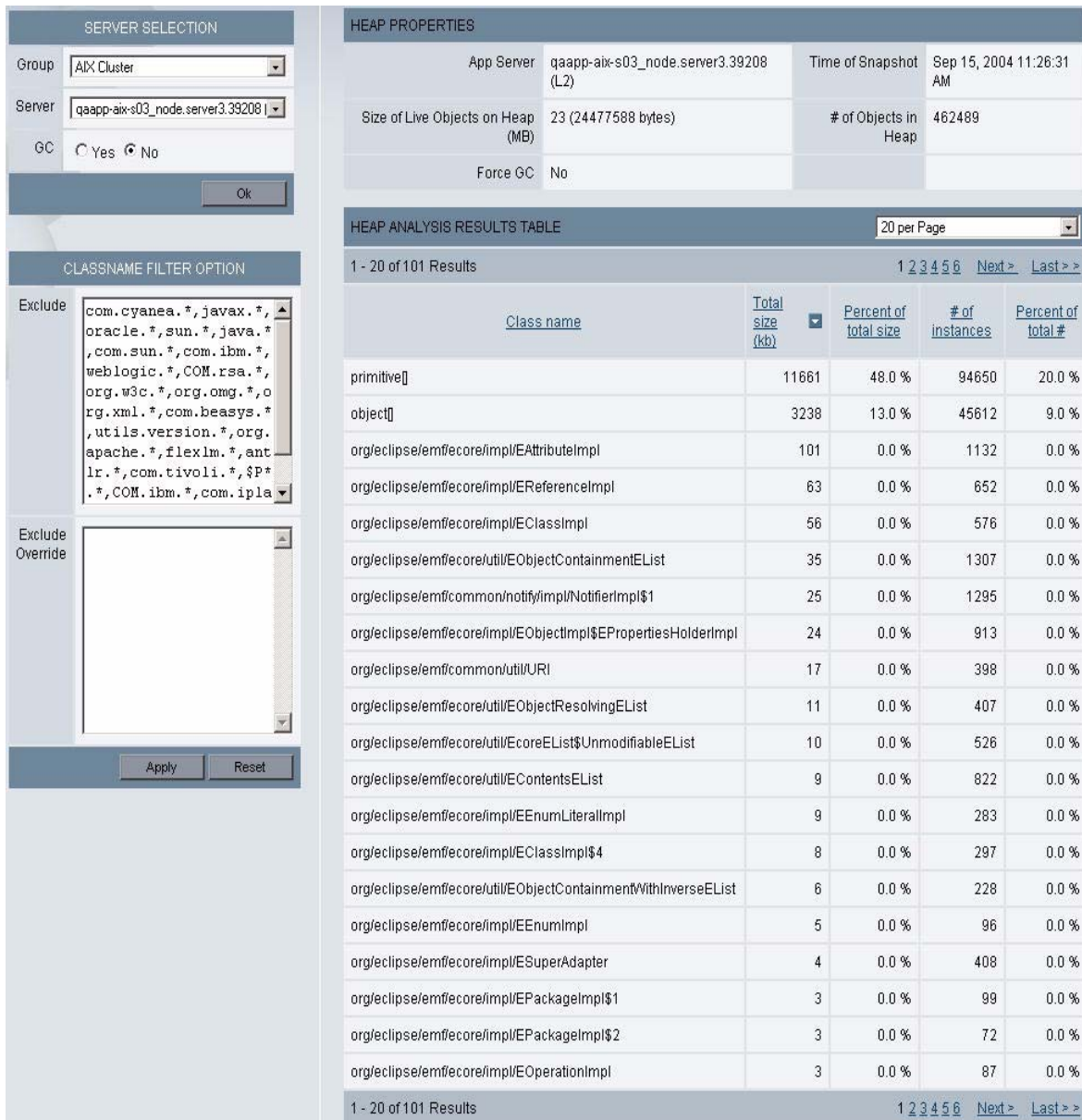


Figure 30. Heap Analysis results

- If you want to narrow the results, enter the names of the classes you want to ignore into the Exclude (Classname) list. If you specify regular expressions in the Exclude list, but want to monitor a subset of these, enter the names of classes you want to monitor into the Exclude Override (Classname) list.
- Click **Apply**.
The new Heap Analysis displays.
- Click **Reset** to return the classname filters to their original settings.

Note: At times, the system may not be releasing memory for a specific class properly. You can use the Heap Analysis to check the heap on your server. If the same classname is being allotted memory in the heap, you may have a memory leak. Use the Memory Leak feature to further investigate the possibility.

Memory Leak

Creating a Memory Leak Confirmation report

Uncover a memory leak trend using the Memory Leak Confirmation report. Compare heap size to several load-oriented metrics to determine that there is in fact a leak, not just a change in workload. The system highlights a leak trend by comparing the average heap size after a garbage collection with a memory increase, increase in users, or increase in volume.

To create a Memory Leak Confirmation report:

1. Click **Problem Determination > Memory Diagnosis > Memory Leak**.

The Memory Leak Overview page opens.

2. Select a Group and a Server.
3. Select the Report Metric type.
4. Click **View Report**.

The Memory Leak Confirmation report opens.

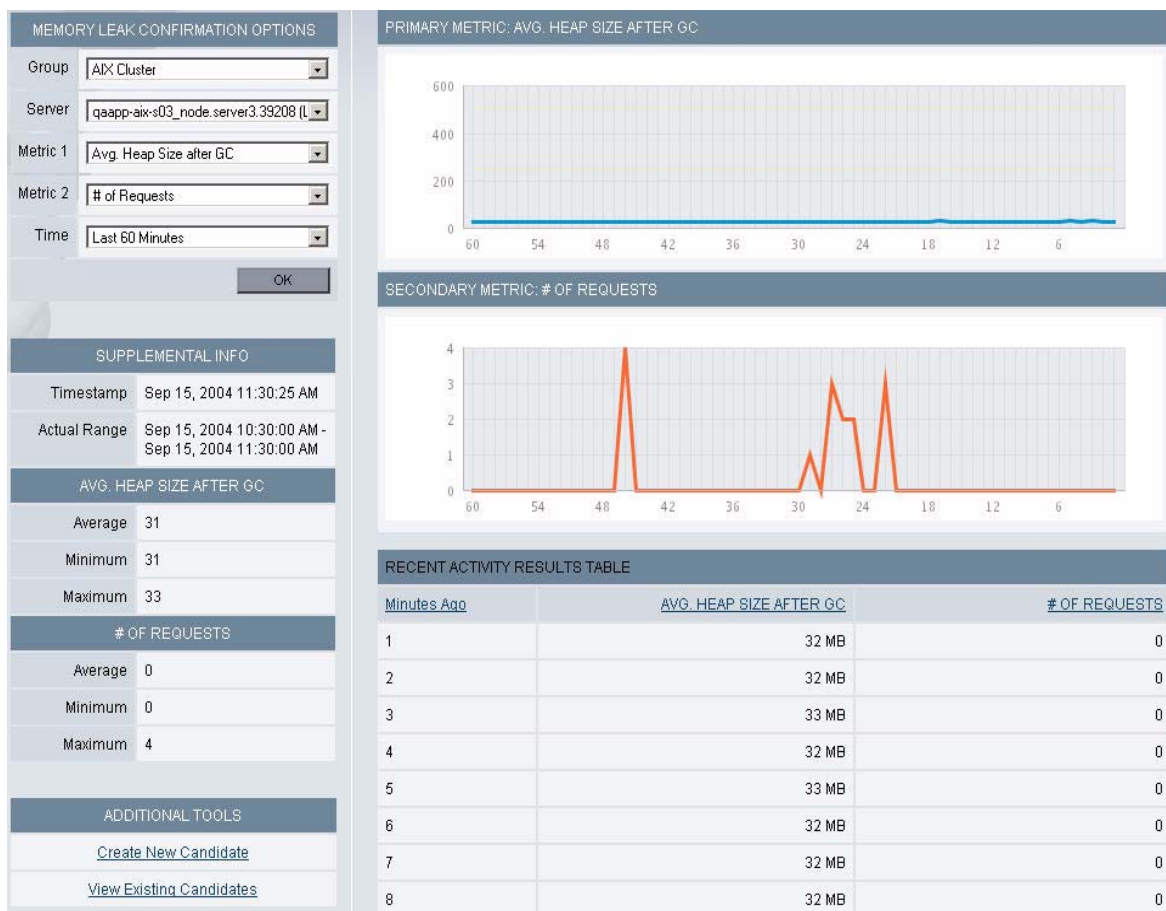


Figure 31. Memory Leak Confirmation report

5. Use the drop-down menus in the left navigation to select a new server, comparison metric, or time.

Note: If there is over 24 hours of data available, your report will show the last 48 hours. Otherwise, your report will display the last 60 minutes.

Creating a Memory Leak Candidate Finder Report

The Memory Leak Candidate Finder Report lets you compare two heap snapshots. Taking two heap snapshots will show if, over time, the number of instances of a specific class is increasing. The report demonstrates whether the number of instances of a class continues to rise over a period of time, which may be a leak candidate.

To create a Memory Leak Candidate Finder Report:

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**.

The Memory Leak Overview page opens.

2. At the bottom of the page, click **Create New Candidate** link.

The Create New Candidate page opens.

3. Select a Group and a Server.
4. Enter the Wait Time and select hours or minutes from the pull-down menu. (There is a 48 hour maximum.) The Wait Time is the amount of time the system waits before taking the second heap snapshot.
5. Click **Save**.

The Memory Leak Candidate Finder Management page displays the report with a Waiting status. Check the report for results after your wait time elapses.

Note: If you receive a "Failed" status on your Memory Leak Candidate Finder report this indicates that either the Data Collector restarted, the Managing Server is down, or there is not enough memory to run the report.

Viewing a Memory Leak Candidate Finder Report

The Memory Leak Candidate Finder Report provides a comparison of the data between two heap snapshots. You can further narrow the results by filtering the class names using the Exclude (Classname) and Exclude Override (Classname) lists.

To view a Memory Leak Candidate Finder Report:

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**.

The Memory Leak Overview page opens.

2. At the bottom of the page, click the **View Existing Candidates** link.

The Memory Leak Candidate Finder Management page opens.

3. The Status for your report should be **Completed**. Click the name of the server, in the Server Name column, link to open your previously created report.

The Memory Leak Candidate Finder Report opens.

4. Click **Comparison Data** on the left navigation.

The comparison data displays with the data for each Heap snapshot.

MENU

- [Management Overview](#)
- [Comparison Data](#)
- [Heap 1 Data](#)
- [Heap 2 Data](#)

CLASSNAME FILTER OPTION

Exclude

Exclude Override

HEAP PROPERTIES

App Server	qaapp-aix-s03_node.server3 (L2)	Time of Snapshot	Sep 15, 2004 11:36:37 AM
Size of Live Objects on Heap (MB)	18 (19679996 bytes)	# of Objects in Heap	395787
GC	Yes		

HEAP ANALYSIS RESULTS TABLE 20 per Page

1 - 20 of 101 Results 1 2 3 4 5 6 Next > Last >>

Class name	Total size (kb)	Percent of total size	# of instances	Percent of total #
primitive[]	8575	44%	78390	19%
object[]	2555	13%	39689	10%
org/eclipse/emf/ecore/impl/EAttributeImpl	101	0%	1132	0%
org/eclipse/emf/ecore/impl/EReferenceImpl	63	0%	652	0%
org/eclipse/emf/ecore/impl/EClassImpl	56	0%	576	0%
org/eclipse/emf/ecore/util/EObjectContainmentEList	35	0%	1307	0%
org/eclipse/emf/common/notify/impl/NotifierImpl\$1	25	0%	1295	0%
org/eclipse/emf/ecore/impl/EObjectImpl\$EPropertiesHolderImpl	24	0%	913	0%
org/eclipse/emf/common/util/URI	17	0%	398	0%
org/eclipse/emf/ecore/util/EObjectResolvingEList	11	0%	407	0%
org/eclipse/emf/ecore/util/EcoreEList\$UnmodifiableEList	10	0%	526	0%
org/eclipse/emf/ecore/util/EContentsEList	9	0%	822	0%

Figure 32. Memory Leak Candidate Finder Report

5. To view each heap individually, click either Heap 1 or Heap 2 on the left navigation.
6. To filter your data more precisely, enter the classes you don't want to analyze into the Exclude (Classname) list. If you specify regular expressions in the Exclude list, but want to monitor a subset of these classes, enter the classes you want to monitor into the Exclude Override (Classname) list. The report will refresh and display with the current data.

Note: When the Comparison for the Memory Leak Candidate Finder Report displays the heap snapshot data, the data includes the classname, the change in the number of instances, and the change in total size. Watch the change in the number of instances; increasing numbers are an indicator of a memory leak in your system.

Chapter 11. JVM Thread Display

Purpose

Use the JVM Thread Display to view all threads running within an application server's JVM.

Usage Overview

This feature helps you:

- View hanging processes in the application server.
- Change the priority or obtain a stack trace of an active thread.
- View a thread dump.

User Scenarios

Scenario 1: How to alleviate high server response time

You are asked to investigate server A where response time and JVM CPU% are higher than expected, but throughput is normal. You don't see any active requests in the In-flight Request Search, so you suspect there may be threads running outside the application server. You access the JVM Thread Display and notice a couple of suspect threads. After taking a thread dump for the JVM, find the details of the current thread that is misbehaving and either reprioritize or cancel the thread.

Notes

Note: This feature is not available for CICS or IMS.

Viewing the JVM Thread Display

The JVM Thread Display page provides information about the Server, Active Thread Groups and their properties. Threads are organized within thread groups. A thread group can contain other thread groups and threads. By default, the top level thread group will be displayed first and you can drill down to view the thread group's contents.

To go to the JVM Thread Display:

1. From the top navigation, click **Problem Determination > JVM Thread Display**.
The JVM Thread Display Server Selection page opens.
2. Select a Group and a Server to view the running threads.
The JVM Thread Display page opens showing all top level thread groups running in the selected server.

SERVER SELECTION

Group: AIX Cluster Server: qaapp-aix-s03_node.server3.39208 (L2)

SERVER PROPERTIES Thread Dump

Snapshot Date	Sep 15, 2004	Application Server Name	server3
Snapshot Time	11:41:25 AM	Application Server IP Address	192.168.4.16

ACTIVE THREADS

[/ system /](#)

Name

- [main](#)
- [RMI Runtime](#)
- [DG event write thread](#)
- [Finalizer](#)
- [GC Daemon](#)
- [GC Helper 1](#)
- [Reference Handler](#)
- [RMI ConnectionExpiration-\[192.168.4.5:9118,com.cyanea.kernel.rmi.CynClientSocketFactory@d05\]](#)
- [RMI ConnectionExpiration-\[192.168.4.5:9118,com.cyanea.kernel.rmi.CynClientSocketFactory@d05\]](#)
- [RMI ConnectionExpiration-\[192.168.4.5:9119,com.cyanea.kernel.rmi.CynClientSocketFactory@d05\]](#)
- [RMI ConnectionExpiration-\[192.168.4.5:9119,com.cyanea.kernel.rmi.CynClientSocketFactory@d05\]](#)
- [RMI LeaseChecker](#)

THREAD GROUP PROPERTIES

Name	system
Active Thread Count	97
Active Thread Group Count	2
Max Priority	10
Daemon Thread Group	No
Destroyed	No

Figure 33. JVM Thread Display

Viewing an Active Thread

Click underlined links in the Active Thread column to drill down into a thread group. From the Thread Group Properties table, you can view:

- Active Thread Count
- Active Thread Group
- Maximum Priority
- Daemon Thread Group
- Destroyed Thread

On the thread level, detailed information such as thread priority, status, stack trace and attributes will be displayed in the Thread Properties table.

To view an active thread:

1. Click **Problem Determination > JVM Thread Display**.
The JVM Thread Display Server Selection page opens.
2. Select a group and a server to view the running threads.
The JVM Thread Display page opens showing all top level thread groups running in the selected server.
3. Select and click the link in the Active Threads column to view the thread groups or the threads inside, or click an active thread for detailed information.

Change a Thread's Priority

If a thread is executing too slowly, you can change its priority by moving it up in the stack, so that it can process a request quickly.

To change a thread's priority:

1. From the top navigation, click **Problem Determination > JVM Thread Display**.
The JVM Thread Display Server Selection page opens.
2. Select a Group and a Server to view the running threads.
The JVM Thread Display page opens showing all top level thread groups running in the selected server.
3. Select and click a thread for detailed information.
4. From the Priority drop-down menu in the Thread Properties table, select a number. Priority 1 is the lowest and priority 10 is the highest.
5. Click **Change Priority**.
The Priority drop-down menu displays the priority you selected for the thread to execute on the request.

Note: When changing a thread's priority, be aware that the new priority remains for the life of the thread. As a result, any requests issued after the change will hold that priority during that request's lifetime.

Viewing a Stack Trace

The Stack Trace page displays the sequence of method execution in a last-in-first-out order. The most recently executed methods will be displayed first in the Stack Trace. For each method, the list includes the Class Name, Method Name and Depth. If a repetitive pattern of method execution is found, the thread may be in an application loop. If a thread is sleeping while holding a lock, thereby preventing other threads from proceeding, the thread may need to be canceled in order to let other threads proceed.

To view a Stack Trace:

1. From the top navigation, click **Problem Determination > JVM Thread Display**.
The JVM Thread Display Server Selection page opens.
2. Select a group and a server to view the running threads.
The JVM Thread Display page opens showing all top level thread groups running in the selected server.
3. Select and click a thread for detailed information.
The Thread Properties table displays the detailed information of the thread that you selected.
4. Click **View Stacktrace**.
The Stack Trace page opens.

Note: The Stack Trace shows the outstanding methods waiting to execute as a result of the request. This trace reports the data unfiltered, so you will see every class. In a normal environment, a request executes quickly so it may be difficult to catch a Stack Trace before completion.

Canceling a Thread

If a thread is misbehaving such as looping, sleeping or abusing resources, it may be necessary to cancel the thread and terminate the executing Java thread to let other threads proceed. However, canceling a thread may result in a number of dangerous

side effects that jeopardize the integrity of the entire JVM including, possible object inconsistency, loss of object integrity, loss of threads in a thread pool and the entire JVM may collapse. Therefore, **the Cancel Thread function should be used only in an emergency with careful consideration of the consequences.**

Note: By default, only the “Administrator” role will have access to the canceled thread functionality.

To cancel a thread:

1. From the top navigation, click **Problem Determination > JVM Thread Display**. The JVM Thread Display Server Selection page opens.
2. Select a group and a server to view the running threads. The JVM Thread Display page opens showing all top level thread groups running in the selected server.
3. Select and click a thread for detailed information. The Thread Properties table displays the detailed information of the thread that you selected.
4. Click **Cancel Thread**.
5. Click **OK** in the confirmation box. The JVM Thread Display page refreshes displaying without the canceled thread.

Note: If the thread is sharing data with another thread and the data is in an inconsistent state at the time of cancellation, unexpected and arbitrary side effects can result. Therefore, this function should only be used when the user is sure that no such side effects can occur. Additionally some JDKs contain a bug that, as the result of canceling a thread, the entire JVM collapses.

Thread Dump

To troubleshoot a problematic multithreading application with a hung thread or looping thread, you may need to view the Thread Dump page for detailed information about memory allocation of threads in a JVM.

When the user clicks Thread Dump on the JVM Thread Display page, a snapshot is taken that shows data about all threads, along with the following information:

- Timestamp
- JVM Signature
- Java Invocation
- System Properties
- Current Thread Dump
- Operating Environment
- Application Environment
- Full Thread Dump
- Monitor Pool Information
- Monitor Pool Dump
- JVM System Monitor Dump
- Thread Identifiers
- Java Object Monitor Dump

To view the Thread Dump page:

1. From the top navigation, click **Problem Determination > JVM Thread Display**. The JVM Thread Display Server Selection page opens.

2. Select a group and a server to view the running threads.
The JVM Thread Display page opens showing all top level thread groups running in the selected server.
3. Click **Thread Dump**.
The Thread Dump page opens.

Chapter 12. Software Consistency Check

Purpose

Use the Software Consistency Check to troubleshoot aberrant servers in an otherwise homogenous server group.

Usage Overview

This feature helps you:

- Detect mismatches in software in a “clone” environment.
- Compare a properly functioning server with other servers in your server farm.

User Scenarios

Scenario 1: Comparing a non-functioning server with working servers

After an upgrade to Application B, which is deployed on multiple servers, requests on Server D are occasionally hanging while all the other servers are working fine. As an Operator, you check the Runtime Environment and compare the server having problems with one of the properly functioning servers. Go to the Installed Binary Check to see if the files on both servers are the same. You find that one of the files on Server D is not the same as the file on the server that is properly functioning. Install the proper file to correct the problem.

Notes

Note: This feature is not available for CICS or IMS.

The Installed Binary Files

Setting up an Installed Binary Comparison

Analyze the data from the Installed Binary Comparison to find out whether your servers contain the same installed binaries. The Installed Binary Comparison allows you to compare the installed binaries on a chosen server (the Authoritative Server) with up to 10 additional servers (the Comparison Servers). The comparison describes whether or not your servers contain the same installed binaries.

To set up an Installed Binary Comparison:

1. From the top navigation, click **Problem Determination > Software Consistency Check > Installed Binary Comparison**.

The Installed Binary Comparison page opens.

2. Under the Authoritative Server, select a Group and a Server.
3. Under the Comparison Servers, select a Group and a Server, or select multiple servers within that group by clicking **Ctrl+the server name**.
4. Click **Next** to continue.

The File Selection page opens.

- Click to select the File Source (EAR file or Class Path) and the File Types (JAR, Web, Class, or Image files).
- Click **OK**.

The Installed Binary Comparison results page displays the Overview data first with the results of the comparison.

Viewing the Results of the Installed Binary Comparison

Review the comparison to find the differences among installed binaries on your servers. Differences in the installed binaries in a server farm can cause unexplained behavior.

- Navigate the results of the Binary Comparison by clicking the expansion icon (+) next to the server name on the left navigation.
- To view further details, click the server name and select either the **Matched** or **Unmatched** folders.
- To view the folder contents, in the Matched folders, select **Full Match**, **File Name/Path/Size Match**, or **File Name Match**, and in the Unmatched folders, select either **Authoritative Only** or **Comparison Only**.

COMPARISON PROPERTIES						Change Comparison
Authoritative Server	qaapp-alc-s01_node.server1.92992 (L3)					
Comparison Server	qaapp-alc-s03_node.server3.40288 (L1)					
File Sources	testware.ear, Trade3.ear, properties, properties, bootstrap.jar, j2ee.jar, Improxy.jar, urlprotocols.jar					
File Types	JAR/ZIP Files, Class Files, Properties Files					
FULL MATCH						
File	Authoritative Date/Time-stamp	Comparison Date/Time-stamp	Authoritative Size (bytes)	Comparison Size (bytes)		
javax/servlet/http/HttpServletResponseWrapper.class (/opt/WebSphere/AppServer51/lib/j2ee.jar)	9/30/02 8:05 PM	9/30/02 8:05 PM	3028	3028	Perform MD5	
com/ibm/xml/parser/TXNodeList.class (/opt/WebSphere/AppServer51/properties/logbr/logbrnml.jar)	8/30/99 9:05 PM	8/30/99 9:05 PM	2952	2952	Perform MD5	
org/apache/html/dom/HTMLAreaElementImpl.class (/opt/WebSphere/AppServer51/installedApps/qaabws5/testware.ear/cyaneaaux-testware.jar)	8/20/01 3:32 PM	8/20/01 3:32 PM	1997	1997	Perform MD5	
/opt/WebSphere/AppServer51/installedApps/qaabws5/testware.ear/testware.web-ws51.war/WEB-INF/classes/com/testware/web/exception/GenerateExceptionAction.class	8/25/04 5:07 PM	8/25/04 5:07 PM	3490	3490	Perform MD5	
org/apache/xerces/framework/XMLContentSpec.class (/opt/WebSphere/AppServer51/installedApps/qaabws5/testware.ear/cyaneaaux-testware.jar)	8/20/01 3:32 PM	8/20/01 3:32 PM	5185	5185	Perform MD5	
com/ibm/ws/security/util/AccessController.class (/opt/WebSphere/AppServer51/lib/bootstrap.jar)	6/27/04 9:31 AM	6/27/04 9:31 AM	1651	1651	Perform MD5	

Figure 34. Folder contents in the Matched folders

- To perform an MD5 checksum on a file, click **Perform MD5**. You can only perform an MD5 checksum on files that are a Full Match or a File Name/Path/Size Match.

Note: The files in the Matched folders contain files that match to varying degrees:

Full Match - indicates that everything matched, including the file name and path, size and file system timestamp. These files are likely to be identical to each other. However, you can perform an MD5 checksum on the files. An MD5 checksum is a unique numeric signature that is different for each file when the contents of the files are different, even if the creation date and the file names coincide.

File Name/Path/Size Match - includes the files with matched file name and path, and size, but not timestamp. These files are likely to be the same. You can perform an MD5 checksum on the files.

File Name Match - indicates that only the file names matched. The files are unlikely to be the same.

The files in the Unmatched folders contain files that exist on either the Authoritative Server or the Comparison Server but not on both:

An **Authoritative Only** indicates that the file only exists on the Authoritative Server.

A **Comparison Only** indicates that the file only exists on the Comparison Server.

Running the Installed Binary Check

The Installed Binary Check provides a list of the installed binaries deployed to the selected server. Use the check to see the details of the installed binaries on your server.

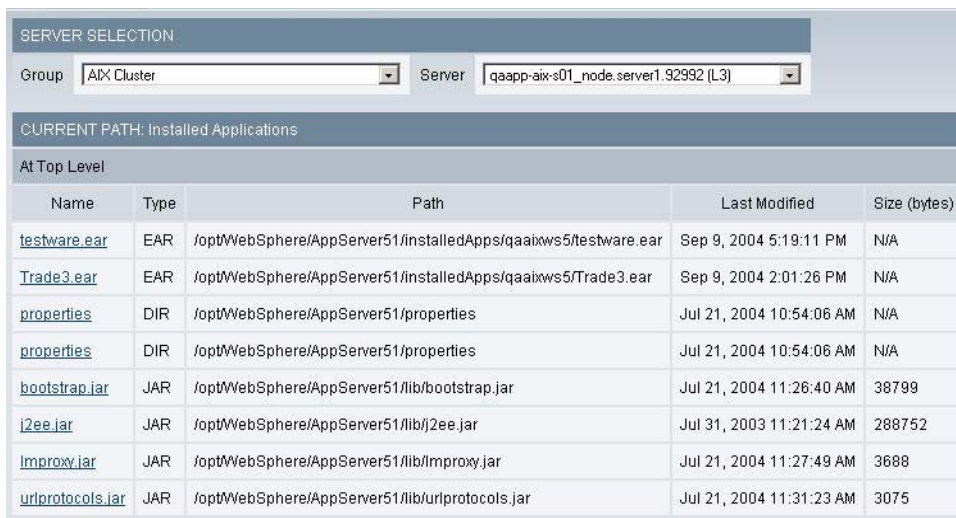
To run the Installed Binary Check:

1. From the top navigation, click **Problem Determination > Software Consistency Check > Installed Binary Check**.

The Installed Binary Check page opens.

2. Select the Group and a Server from the drop-down menus.

Details are displayed of the selected server's installed binaries.



The screenshot shows a web interface for the Installed Binary Check. At the top, there is a 'SERVER SELECTION' section with two dropdown menus: 'Group' set to 'ADX Cluster' and 'Server' set to 'qaapp-aix:s01_node.server1.92992 (L3)'. Below this is a 'CURRENT PATH: Installed Applications' section. Underneath, there is a table titled 'At Top Level' with columns for Name, Type, Path, Last Modified, and Size (bytes). The table lists several files and directories, including testware.ear, Trade3.ear, properties, bootstrap.jar, j2ee.jar, lmpoxy.jar, and urlprotocols.jar.

Name	Type	Path	Last Modified	Size (bytes)
testware.ear	EAR	/opt/WebSphere/AppServer51/installedApps/qaabxws5/testware.ear	Sep 9, 2004 5:19:11 PM	N/A
Trade3.ear	EAR	/opt/WebSphere/AppServer51/installedApps/qaabxws5/Trade3.ear	Sep 9, 2004 2:01:26 PM	N/A
properties	DIR	/opt/WebSphere/AppServer51/properties	Jul 21, 2004 10:54:06 AM	N/A
properties	DIR	/opt/WebSphere/AppServer51/properties	Jul 21, 2004 10:54:06 AM	N/A
bootstrap.jar	JAR	/opt/WebSphere/AppServer51/lib/bootstrap.jar	Jul 21, 2004 11:26:40 AM	38799
j2ee.jar	JAR	/opt/WebSphere/AppServer51/lib/j2ee.jar	Jul 31, 2003 11:21:24 AM	288752
lmpoxy.jar	JAR	/opt/WebSphere/AppServer51/lib/lmpoxy.jar	Jul 21, 2004 11:27:49 AM	3688
urlprotocols.jar	JAR	/opt/WebSphere/AppServer51/lib/urlprotocols.jar	Jul 21, 2004 11:31:23 AM	3075

Figure 35. Installed Binary Check

Viewing the Installed Binary Check Detail

From the Installed Binary Check which is a list of the installed binaries deployed to the selected server, you can drill down to see details of a specific binary. The details include the name, type, path, last modified timestamp and size.

To view the Installed Binary Check Detail page:

1. From the top navigation, click **Problem Determination > Software Consistency Check > Installed Binary Check**.

The Installed Binary Check page opens.

2. Select a Group and a Server from the drop-down menus.

The Installed Binary Check page opens displaying the details of the selected server's installed binaries.

3. Click the Name's link to drill down to the detail of the specific binary.

The Installed Binary Check Detail page opens.

The Runtime Environment

Running the Runtime Environment Comparison

Analyze the data in the Runtime Environment Comparison and find out if the runtime environments on all your clone servers are the same. The Runtime Environment Comparison allows you to compare the runtime environment on a chosen server (the Authoritative Server) with up to 10 additional servers (the Comparison Servers). If you are experiencing strange behavior in your server farm, a runtime environment comparison shows whether or not the servers in your farm have identical environments.

To use the Runtime Environment Comparison:

1. From the top navigation, click **Problem Determination > Software Consistency Check > Runtime Environment Comparison**.

The Runtime Environment Comparison page opens.

2. Under the Authoritative Server, select a Group and a Server.
3. Under the Comparison Servers, select a Group, and then select multiple servers within that group by clicking **Ctrl+server name**.
4. Click **Next** to continue.

The Runtime Environment Comparison Results page displays the data.

5. For specific data on the servers, click any of the options in the left navigation under System Runtime Environment, Java Runtime Environment, and the AppServer Runtime Environment.

The specific data displays in the main window.

6. For a complete detail report on a particular server, click the server's name. The Runtime Environment Check page displays all the available data on the System Runtime Environment, Java Runtime Environment, and the AppServer Runtime Environment for the selected server.
7. Click **Change Comparison** to set up another runtime environment comparison.

Running the Runtime Environment Check

The Runtime Environment Check page provides runtime environment details for a selected server.

To select a server's runtime environment:

1. From the top navigation, click **Problem Determination > Software Consistency Check > Runtime Environment Check**.

The Runtime Environment Check page opens.

2. On the left navigation, select the Group and a Server from the drop-down menus. Details of the selected server's runtime environment are displayed.

SYSTEM RUNTIME ENVIRONMENT	
CPU Speed	602 MHz
# of CPU's Online/Total	4/4
Memory	2047 MB
Operating System Info	AIX 5.1
JAVA RUNTIME ENVIRONMENT	
JDK Version	IBM Corporation J2RE 1.4.2 IBM AIX build ca1420-20040626 (JIT enabled: jitc) Classic VM
Initial Java Heap Size	256 MB
Maximum Java Heap Size	512 MB
Installation Directory	/opt/WebSphere/AppServer51/java/bin/.jre
Class Path	/opt/WebSphere/AppServer51/properties
	/opt/WebSphere/AppServer51/properties
	/opt/WebSphere/AppServer51/lib/bootstrap.jar
	/opt/WebSphere/AppServer51/lib/j2ee.jar
	/opt/WebSphere/AppServer51/lib/improxy.jar
Library Path	/opt/WebSphere/AppServer51/lib/urlprotocols.jar
	/opt/WebSphere/AppServer51/java/bin/.jre/bin
	/opt/WebSphere/AppServer51/java/jre/bin/classic
	/opt/WebSphere/AppServer51/java/jre/bin
	/opt/dc_ws511_aix_31 cluster/lib
	/opt/WebSphere/AppServer51/java/bin/.jre/bin
	/opt/WebSphere/AppServer51/java/jre/bin/classic
	/opt/WebSphere/AppServer51/java/jre/bin
	/opt/WebSphere/AppServer51/bin
	/usr/mqm/java/lib
	/usr/opt/wemps/lib
	/home/instver8/sqllib/lib
/usr/lib	
APPLICATION SERVER RUNTIME ENVIRONMENT	
App Server	IBM WebSphere Application Server 5.1.1
Startup Directory	/opt/WebSphere/AppServer51
Listening Ports	:9143:HTTPS
	*:9183:HTTP
	*:9181:HTTP
	*:9447:HTTPS
	*:9445:HTTPS
	:9190:HTTP
# EAR Files	2
# Registered Web Modules	2
# Registered EJB Modules	2
JDBC Connection Pools	2

Figure 36. Runtime Environment Check

The Million Instruction Per Second (MIPS) power of the MVS™ machine is computed from an empirical formula derived from the System Resources Manager (SRM) service units/second, derived itself from the RMCTADJC field of the RMCT. RMCTADJC is the CPU rate adjustment expressed in the number of sixteenths of one CPU microsecond per CPU service unit.

Note: This feature does not apply to the non-z/OS platform.

The computation algorithm is as follows:

1. Get RMCTADJC from the RMCT data area.
2. Compute service units per CPU second $SU=16000000/RMCTADJC$.
3. Estimate the number of MIPS per CPU by $SU/48.5$ (*).
4. Finally compute the estimated MIPS power of the MVS machine by multiplying by the number of CPUs.

(*) The number 48.5 is borrowed from the recommendations of Thierry Falissard's home page on OS/390, and is used by some MVS utilities in capacity planning (in particular the SHOWMVS utility).

Note: This information is only an estimation of the machine's CPU power.

Chapter 13. Trap & Alert Management

Purpose

Use Trap & Alert Management to monitor server health and determine problems with applications.

Prevent disruptions in service by receiving alerts before problems arise. Gather data that helps you pinpoint the root cause of difficult-to-reproduce problems.

Usage Overview

This feature helps you:

- Monitor a group of servers or a selected server.
- Find out immediately when servers, applications, components or methods are not healthy, and obtain the data necessary for diagnosis.

User Scenarios

Scenario 1: Diagnosing Garbage Collection

It was observed on server J that, every so often, garbage collection takes over five minutes and, during these times, requests that typically complete in a few milliseconds take ten seconds to complete. Since this problem does not occur frequently, you set a trap so that you can find out immediately when server J's garbage collection is churning. In particular, you choose a Server Resource Trap for Garbage Collection Time with a Threshold of 120,000 ms (two minutes), choose the Alert Action to Send Email to your pager, and apply this trap to server J. When you receive the page, you have about three minutes to investigate server J (assuming that this is an example of where the garbage collection underway will take five minutes).

Scenario 2: Debugging complex applications

You are monitoring application A, which has a J2EE component on server S and a legacy CRM back end. The Java component of application A frequently exhibits idle times of several seconds, even when there is not much load on server S. You do not wish to run at L3, but you want to see in what methods the Java application is waiting. You set an Application Trap for Wait Time with a Threshold of 2,000 ms, by Request for application A, choose the Stack Trace Data Action and apply this trap to server S. The next time a request for application A takes longer than two seconds, the system will take a stack trace of server S. Look in the Trap Action History to obtain the stack trace, to determine where application A is waiting.

Notes

Note: Traps may add to the overhead used by your system, so use them sparingly.

Note: The following metrics are not supported for z/WebSphere 4:

- JDBC Pool
- JDBC Concurrent Waiters
- JCA Pool

- Thread Pool

This feature is not available for CICS or IMS.

Managing Traps and Alerts

Manage the software traps and alerts set on your system by adding and deleting them when necessary on the Trap and Alert Management page. Active traps display in the Active Trap table at the top of the page; a list of all the traps created displays in the Trap Profiles table at the bottom of the page. Maintain your existing traps and alerts by modifying them as needed.

To manage traps and alerts:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.

ACTIVE TRAPS									
Trap Name	Server	Suppression	Duration	Time Left	Iterations Left	Time Set	Set By	Modify Activation	Deactivate
1Occur	qasunw17.server1		Infinite	N/A	1	Sep 15, 2004 12:19:36 PM	admin		
3ResTime	qasunw17.server1		Infinite	N/A	1	Sep 15, 2004 12:20:02 PM	admin		
6UncaughtExceps	qasunw17.server1		Infinite	N/A	3	Sep 15, 2004 2:57:55 PM	admin		

TRAP PROFILES						
Trap Name	Description	Created By	Activate	Modify	Duplicate	Delete
10GCFreq		admin				
11GCTime		admin				
12JVMHeapafterGC		admin				
13JDBCPool		admin				
14Waiters		admin				
15ThreadPool		admin				
16JCAPool		admin				
17ReqFreq		admin				
18Sessions		admin				
19Availability		admin				
1Occur		admin				
20AvgRespTime		admin				
21JavaExceptions		admin				
2CPUTime		admin				
3ResTime		admin				
4WaitTime		admin				
5InFlight		admin				
6UncaughtExceps		admin				
7AvgPlatCPU		admin				
8AvgJVMCPU		admin				
9JVMHeap		admin				
AADD	as	admin				
AAJDBC	as	admin				
arun	arun	admin				
arunsrin	as	admin				
gre1	as	admin				
gre2	as	admin				
gre3	as	admin				
gre4	sa	admin				
gre5	as	admin				
gre7	asa	admin				
gre8	as	admin				
gre9	as	admin				
greA	as	admin				
greC	as	admin				
greJDBC	as	admin				
hello		admin				
ni1	as	admin				
occurtest		admin				
trapJDBC	as	admin				

Figure 37. Trap & Alert Management

- From the management page, you may create a new trap, activate/deactivate, modify, duplicate or delete existing traps.

Note: For z/OS WebSphere version 4, the Application Server Name is identified using the standard method: <hostname>.<generic server name>.<server instance name>, for example, hostname = ADCDPL, (generic) server name = BBOASR2, and server instance name = QATEST1. For z/OS WebSphere version 5, the Application Server Name is identified using <short cell name>.< short node name>.<long server name>.

Setting an Application Trap

An Application Trap detects metrics in a request, method, SQL or MQI call. The system triggers the trap after the monitored server exceeds the threshold for the metric you set.

When the trap is triggered, and when the action conditions are met, then any alerts you have activated (whose conditions have been met) will be sent, and any actions you have specified (for this trap) will be performed.

For example, you may want to know when a server receives more than 10 requests that are named "login". You can activate a trap and have it send an email message to broadcast the occurrence, and to collect a Method Trace of the request. These actions you specify will take place when the trap is triggered by the 10th occurrence of the "login" request.

To set an Application Trap, you must define the trap, set alerts and data actions, and then activate the trap on one or more application servers.

To define an Application Trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.

2. On the left navigation, click **Create Trap**.

The Trap Type Selection page opens.

3. Select **Application Trap** as the Trap Type.

The Target Type drop-down is repopulated with Application Trap options.

4. Select one of five **Target Types** from the drop-down menu:

- Occurrences
- CPU Time
- Resident Time - Completed
- Wait Time
- Resident Time - In-Flight

5. Click **Next**.

The Step 2--Define Trap page opens.

6. Select one of the three trap types in the **Trap Definition** section:

- Request
- Method
- SQL

7. Complete the rest of the fields in the Trap Definition section, to restrict which events will trigger the trap. No blank fields are allowed.

Note: The Request field is interpreted as a substring match: any request name that contains the substring you enter will match. The value "*" matches all requests. You must enter something in the Request field.

Note: The Method field is interpreted as an exact match: the only methods that match are those whose names are exactly the same as the string you enter. For the Method field, entering nothing is equivalent to "*", which matches all methods.

Note: The additional fields for SQL traps are treated like the Method field: as substring matches, where leaving a field blank, or entering "*", matches all values.

This completes the trap definition.

Click **Next** to proceed to the Step 3--Set Trap Alerts page. See "Setting Alert Actions and Data Actions".

Setting a Server Resource Trap

A Server Resource Trap measures a variety of Target Types. The system will trigger a trap after exceeding the threshold for the selected target type. When the trap definition is met, the alert actions occur.

For example, if you set a trap to alert you via email when a server is unavailable two times, when the server becomes unavailable a second time, the system will send you an email.

To set a Server Resource Trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.

2. On the left navigation, click **Create Trap**.

The Trap Type Selection page opens.

3. Select Server Resource Trap as the Trap Type.

4. Select one of the Target Types from the drop-down menu.

5. Click **Next**.

The Define Trap page opens.

6. Enter a threshold at which you want the trap to trigger.

7. Click **Next**.

The Set Trap Alerts page opens. See "Setting Alert Actions and Data Actions" for details on setting trap actions.

Setting Alert Actions and Data Actions

Regardless of the trap type, you must specify trap actions as part of the trap definition.

Trap actions include alerts and data actions. Alerts include messages sent by email or SNMP, whereas data actions capture Method Traces, Stack Traces or Thread Dumps.

Trap actions occur when a trap triggers. You may configure alert actions to be suppressed, to avoid getting spammed by alerts.

Note: Data actions are not available for Server Resource traps.

To set alert actions and data actions:

1. You can set trap alerts on the Step 3--Set Trap Alerts page, which is part of the trap creation process. See one of these two procedures for details on how to arrive at the Step 3--Set Trap Alerts page: "Setting an Application Trap" on page 96 or "Setting a Server Resource Trap" .

TRAP TYPE	
Trap Type	Server Resource Trap
Target Type	Average Platform % CPU

TRAP DEFINITION	
Threshold	Average Platform % CPU (% usage) > 10

TRAP ALERT SETTINGS			
Condition	If Trap Definition occurs <input type="text"/> time(s)		
Severity	<input type="text" value="Low"/>		
Alert Action(s)	<input type="checkbox"/> Send Email (comma separated) <input type="text"/>		
	<input type="checkbox"/> Send SNMP Message		
<input type="button" value="Add"/>			
Condition(s)	Severity	Alert Action(s)	Delete

DEFAULT SUPPRESSION SETTINGS (optional)	
After an alert is sent, don't repeat the same alert for	<input type="text"/> minutes
<input type="button" value=" < Back"/> <input type="button" value=" Cancel"/> <input type="button" value=" Next >"/>	

Figure 38. Set Trap Alerts

- For the Trap Alert Settings, in the Condition field, enter the number of times the trap should trigger before the action is taken.

Note: This value will be applied to all the trap actions defined in the next two steps. If you want to define multiple actions, each with a different condition, repeat steps 2-4 once for each distinct condition.

- Click to select the Severity level from the drop-down menu.

Note: WSAM has three severity levels. Since WSAM provides SNMP integration with Tivoli®, the three severity levels of WSAM are mapped to the warning levels of Tivoli as below:

WSAM Severity Level	Tivoli Warning Level
Low	Harmless
Medium	Minor
High	Critical

- Add at least one action, either an Alert Action (email or SNMP message) or a Data Action (Method Trace, Stack Trace or Thread Dump). (The Thread Dump is not available on the Windows® platform.)

To select an action, click its checkbox. For the email action, also enter the list of email addresses to which the message will be sent.

Click **Add** to add the actions to your trap. Repeat this step until you have added all the actions you want. You can change the values of the Condition and Severity fields (steps 2 and 3) each time you add a new action.

Note: When looking for a trace, consider selecting both Method Trace and Stack Trace as Data Actions, since a request executes quickly and it is difficult to catch a Stack Trace before completion.

Note: Method Trace actions will only contain method-level information if the Data Collector to which the trap is applied is running at L3 when the trap is triggered.

5. Set the Default Suppression Setting if you want to avoid getting spammed by Alert Actions that might occur in rapid succession.

The actions you specify are taken when the trap is triggered, and the alerts' conditions met. To see data associated with triggered traps, use the Trap Action History page.

Setting a suppression duration means that, once an alert is sent, recurring alerts will not be sent for that duration. Once that duration has elapsed, an alert will be sent the next time the trap triggers that action. Recurring alerts will again be suppressed for the suppression duration. This will continue until the suppression has been deactivated.

Trap alert suppression can be created in the trap definition, which is called the default, or while activating a trap, where you can override the default suppression period.

You can override this default value when you apply a trap to a server.

The suppression duration you apply when you activate a trap (either the default or the overridden value) applies to all alert actions in the trap definition. However, each alert action handles this duration independently. Therefore, when different alerts have different conditions, and fire at different times, one alert's suppression will not affect the other.

Click **Next** to proceed.

The Name Trap page opens.

6. Enter a Name and descriptive text for your trap.

Click either **Save** or **Save & Activate**

If you click **Save**, the Trap and Alert Management page opens displaying your new trap.

If you click **Save & Activate**, the Activate page opens. To activate a trap, see "Activating a Trap".

Activating a Trap

For your convenience, you can turn traps off and on by activating and deactivating them. Since traps add overhead to your system, you may want to turn them on only at the times necessary. The traps in the Trap List are not active.

To activate a trap:

1. You may arrive at the Activate page by clicking the **Save & Activate** button on the Name Trap page; see "Setting an Application Trap" on page 96 or "Setting a Server Resource Trap" on page 97 to start from the beginning of the trap creation process.

You may arrive at the Activate page by activating an existing trap, as follows: from the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.

2. In the Trap Profiles list, click **Activate** next to the trap you want to activate.

The Activate page opens.

TRAP PROPERTIES	
Trap Name	Tech Writing
Description	
Created By	admin

SERVER SELECTION	
Server Filter	Select a Group
Server	

ALERT SUPPRESSION SETTINGS	
<input checked="" type="radio"/> Trap Default	
<input type="radio"/> Override Default	After an alert is sent, don't repeat the same alert for <input type="text"/> minutes

DEACTIVATION SETTINGS	
If neither option is enabled, the trap will run indefinitely.	
<input type="checkbox"/> Deactivate after <input type="text"/> minutes	
<input type="checkbox"/> Deactivate after <input type="text"/> occurrences	

Figure 39. Activate page

3. Select a Group and a Server.

Note: If you select All Servers, the trap will only apply to the servers in the group at the time the system activates the trap. Any new servers created will not use the trap. In addition, there is one trap per server, not one trap that accumulates actions for all servers.
4. Set the Alert Suppression Settings by entering the amount of time you want to suppress alerts after the first alert is sent.

Click the **Trap Default** radio button to use the default suppression for the trap, or click the **Override Default** radio button to set a specific suppression duration for this particular trap activation.

If you do not want to suppress any alerts, enter a value of 0, or leave the field blank.
5. If you want the trap to run indefinitely, do not check either of the checkboxes in the Deactivation Settings section.

If you want the trap to deactivate, click one or both of the checkboxes for **Deactivate after ... minutes** or **Deactivate after ... occurrences**, and fill in the value(s) for minutes or occurrences.

If both deactivation settings are selected, the trap will deactivate when the first of the two deactivation conditions is met.
6. Click **Activate**.

The Trap and Alert Management page displays the trap in the Active Traps table at the top of the page.

Deactivating a Trap

Deactivate your traps when they are not being used since they can add overhead to the system. The traps in the Trap Profiles table are not active.

To deactivate a trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.
2. In the Active Traps table, click **Deactivate** next to the trap you want to deactivate.

3. Click **OK** at the confirmation box.

The trap displays in the Trap Profiles table as deactivated.

Note: A trap must be deactivated prior to modification.

Modifying a Trap

After creating a trap, you can modify most of the parameters of a trap. Change the Group, Server, Trap Type, Target Type, Alert Conditions and the Action that occurs when the server meets the conditions. Using this method you can reuse and modify old traps for different servers.

Note: A trap must be deactivated prior to modification (See “Deactivating a Trap” on page 100).

To modify a trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.

2. In the Trap Profiles, click **Modify** next to the trap you want to modify.

The Modify page opens.

3. If you want to change the Trap Definition, you can modify any of the available fields.

4. If you want to change the name, enter a new Name and descriptive text for your trap. This will replace the old name when saved.

5. Click **Save**.

The Trap and Alert Management page opens displaying your modified trap.

Duplicating a Trap

Save time by duplicating traps. Duplicating a trap allows you to quickly create a new trap based on the settings of an existing trap.

To duplicate a trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.

The Trap and Alert Management page opens.

2. In the Trap Profiles table, click **Duplicate** next to the trap you want to duplicate.

The Duplicate page opens.

3. Select the trap you want to duplicate from the drop-down menu.

4. Enter a name for the new trap.

5. Click **Save**.

The new trap displays in the Trap and Alert Management page.

Deleting a Trap

Manage your traps by keeping them up-to-date. Delete existing traps from the system that are no longer in use.

Note: A trap must be deactivated prior to deletion (See “Deactivating a Trap” on page 100).

To delete a trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.
The Trap and Alert Management page opens.
2. In the Trap Profiles table, click **Delete** next to the trap you want to delete.
3. Click **OK** at the confirmation box.
The Trap and Alert Management page opens without the deleted trap.

Viewing a Trap Action History

The Trap Action History page provides a record of traps that met the set conditions. You may view the trap history such as the date and time that the action was taken, trap properties, server name, severity, and the type of action that was taken.

Note: You may activate, modify, duplicate or delete a fired trap on the Trap Properties page. Click the link in the Trap Name column to open the Trap Properties page.

To view a fired trap:

1. From the top navigation, click **Problem Determination > Trap & Alert Management**.
The Trap and Alert Management page opens.
2. On the left navigation, click **Trap Action History**.
The Trap Action History opens displaying the information for the fired traps.
3. Click **Show Filters**.
4. You may either filter the information by server or by server and trap name but not by the trap name only. Click to select the group name and the server name, and a trap name (if applicable), then click **Filter**.
The Trap Action History page refreshes displaying the filtered trap information that you selected.
5. To delete a fired trap history, check the Delete box next to the trap that you want to delete and click **Delete**.
The Trap Action History page refreshes displaying without the deleted trap history.

Chapter 14. System Resources

Purpose

System Resources helps you tune your application servers.

Usage Overview

This feature helps you:

- Find bottlenecks in application server resources.
 - Database Connection Pools
 - Thread Pools
 - JCA Connection Pools
- Gather the information you need in order to tune an application server's managed resources.
- Understand the internals of an application server and how they are utilized by your workload:
 - JVM/System
 - Server
 - Web Applications
 - Servlet/Session Manager
 - EJB
 - JMS
 - JCA
 - JTA
 - ORB
 - SQL
 - MQI
 - Execute Queues

User Scenarios

Scenario 1: Eliminating bottlenecks

The response time of application A becomes unacceptable once the server is experiencing modest throughput. You see that much of the resident time is spent idle. To see if the cause is a bottleneck in the application server pools, use System Resources during these times to view the percentage of threads used in the Database Connection Pools, Thread Pools and/or JCA Connection Pools. If any pool is at or near 100%, it is likely that demand for application A is saturating those resources. You may be able to fix the problem by creating more or larger pools.

Scenario 2: Diagnosing imbalanced performance

You have several supposedly identical servers in server group G that host the same applications and have similar workloads. However, one of your servers in server group G is noticeably more sluggish than the others. To investigate more specifically the differences in performance and resource usage among these servers, you use System Resource Comparison to compare these servers, one resource at a time. You may find that they have different resources available, are configured differently, or serve different workloads.

Notes

Note: This feature is not available for CICS or IMS.

Viewing the System Resources Overview - Non-z/OS

Typically, you access this page by using the top navigation as described below or by clicking on the Tools button on the Server Statistics Overview page and the Server Overview page. After selecting an application server from a group, the System Resources Overview page displays data for all the resources on the application server.

To open the System Resources Overview page:

1. From the top navigation, click **Availability > System Resources**.

The System Resources Overview selection page opens.

2. On the left navigation, select the Group and the Server from the drop-down menu. The System Resources Overview page opens displaying the information for the selected Group and Server.

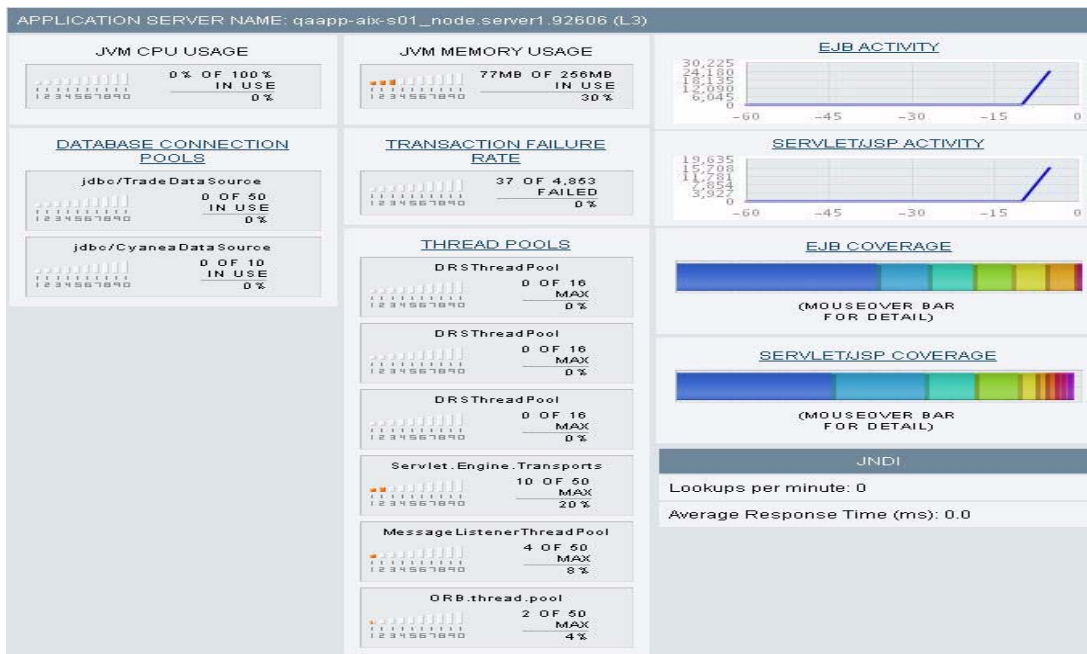


Figure 40. System Resources Overview

Note: When an application server malfunctions, the user can go to the System Resources Overview page to view a quick summary of the resources managed by the application server, such as, database connection pools, thread pools, JVM CPU usage, memory usage, transaction failure rate, EJB activity/coverage, servlet/JSP activity/coverage, and JNDI etc. When reviewing the data, pay close attention to the graphs; when there is a problematic situation, the graphs display the data in red.

To change the application server:

1. On the left navigation, select a Group from the drop-down menu.
2. On the left navigation, select a Server from the drop-down menu.

Viewing the System Resource Overview - z/OS

The System Resources Overview page displays SMF Data or PMI Data for all the resources on the selected application server.

To open the System Resources Overview page:

1. From the top navigation, click **Availability > System Resources**.
The System Resources Overview selection page opens.
2. On the left navigation, select the Group and the Server from the drop-down menu.

The information for the selected Group and the Server displays.

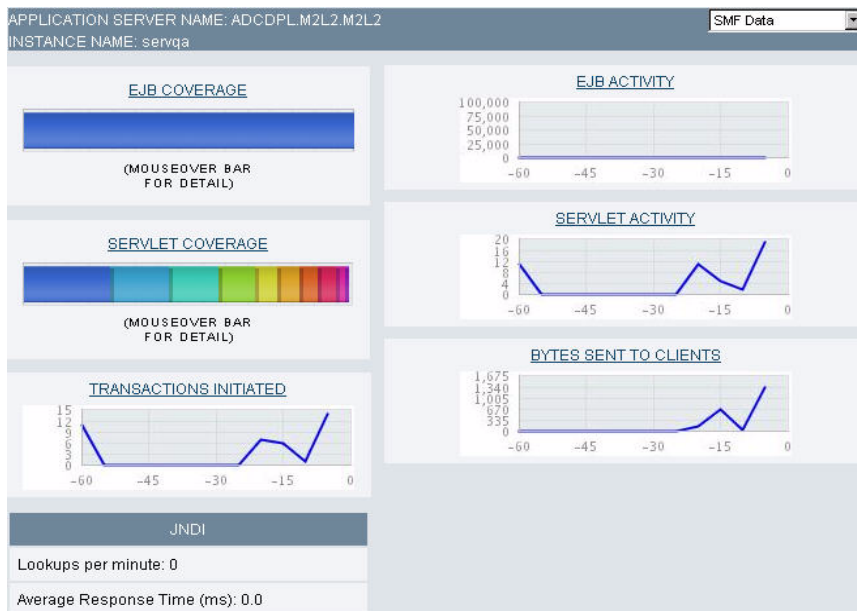


Figure 41. System Resources Overview - SMF Data

3. Click to select **PMI Data** from the drop-down menu at the upper right corner.
The System Resources Overview - PMI Data page opens.

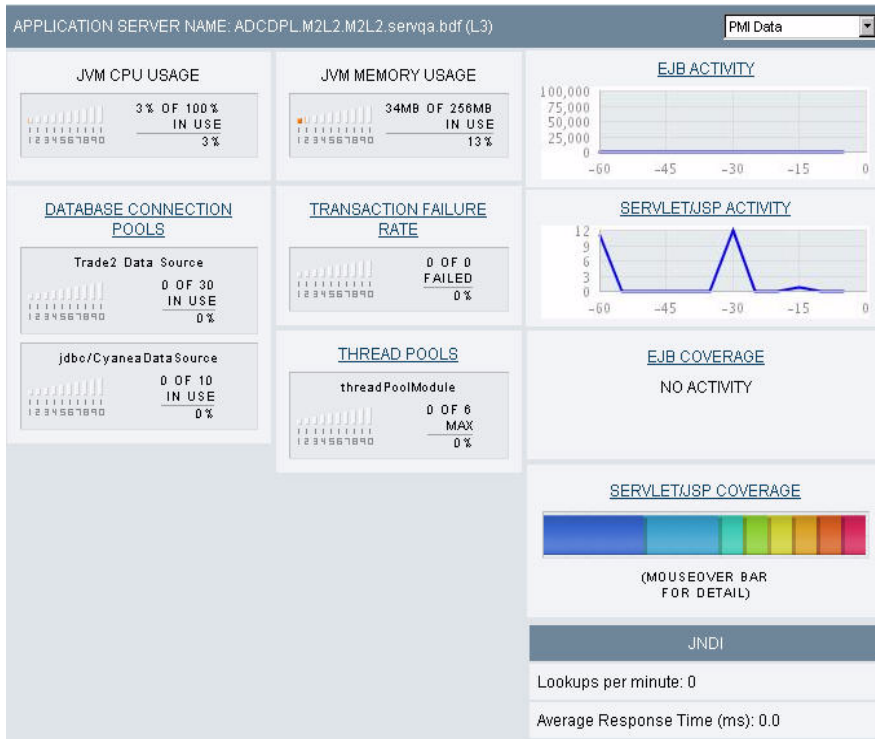


Figure 42. System Resources Overview - PMI Data

General

The following section addresses the metrics and modules that apply to the general data.

The following information provides the metrics for the general data type and the application servers it supports. The tables provide information for the following modules:

- CICS Transactions
- Queue Manager
- SQL

Table 2. CICS Transaction data

CICS Transaction Metrics	WebSphere 4	WebSphere 5
Average Response Time	x	x
Program Executions	x	x

Table 3. Queue Manager data

Queue Manager Metrics	WebSphere 4	WebSphere 5
MQCONN(X) Average Response Time	x	x
MQPUT(1) Average Response Time	x	x
MQGET Average Response Time	x	x

Table 3. Queue Manager data

Queue Manager Metrics	WebSphere 4	WebSphere 5
The following metrics report data for both Executions and Average Response Time:		
MQBACK	x	x
MQBEGIN	x	x
MQCLOSE	x	x
MQCMIT	x	x
MQCONN	x	x
MQCONNX	x	x
MQDISC	x	x
MQGET	x	x
MQINQ	x	x
MQOPEN	x	x
MQPUT	x	x
MQPUT1	x	x
MQSET	x	x

Table 4. Queue data

Queue Metrics	WebSphere 4	WebSphere 5
MQPUT(1) Executions	x	x
MQPUT(1) Average Response Time	x	x
MQGET Executions	x	x
MQGET Average Response Time	x	x

Table 5. SQL data

SQL Metrics	WebSphere 4	WebSphere 5
The following metrics report data for both Average Response Time and Calls per Minute:		
Delete	x	x
Insert	x	x
Select	x	x
Stored Procedure	x	x
Unidentified	x	x
Update	x	x

WebSphere

PMI Data

It is by default that when you choose a WebSphere 5 application server, the System Resources Overview - PMI page will open. You may drill down into different pages of the resources to view the detailed information for EJBs, Database/JCA Connection Pools, Servlet/Session Manager, Thread Pools, JTA Transactions, Web Applications, SQL Data, ORB, and JVM/System.

SMF Data

From this page, you can view detailed information on Server, EJBs, Servlet Session Manager, Web Applications, Server Regions and SQL.

The source of the data comes primarily from the SMF records published periodically by WebSphere. As these records are published, WSAM intercepts the transfer of the records and makes a copy in real time before writing it to the SMF dataset. The system collects and presents records captured in real time in this section.

Note: While you can access the System Resources at the server region level, the system displays the data aggregated at the server instance level.

Please refer to the WebSphere breakdown table below for information on whether PMI or SMF data is supported.

Table 6. WebSphere Breakdown

	PMI	SMF
WebSphere 4	x	
WebSphere 5	x	
z/WebSphere 4		x
z/WebSphere 5	x	x

WebSphere - PMI

The following information provides the metrics for the WebSphere - PMI data type and the application servers it supports. The tables provide information for the following modules:

- Database Connection Pools
- Enterprise Java Beans
- JCA Connection Pools
- JTA Transactions
- JVM/Systems
- ORB Detail/Interceptor
- Web Applications
- Session Manager
- Thread Pools

Table 7. Database Connection Pool data

Database Connection Pool Metrics	WebSphere 4	WebSphere 5
Avg. Waiting Threads	x	x
Percent Used	x	x
Avg. Wait Time		x
Avg. Time in Use		x
Connection Pool Faults	x	x
Avg. Pool Size	x	x
# Creates	x	x
# Destroys	x	x
Free Pool Size		x
# Managed Connections		x
JDBC Operation Timer		x
# Allocations	x	x
# Prepared Stmts Cache Discards	x	x
# Returns	x	x
Percent Maxed	x	x
# Connections		x

Table 8. EJB data

EJB Metrics	WebSphere 4	WebSphere 5
Avg. Method Response Time	x	x
Total Method Calls	x	x
# Ready Beans/Concurrent Actives	x	x
# Removes	x	x
# Destroys	x	x
Avg. Method Resp. Time for Create		x
Total Method Calls	x	x
Gets Found	x	x
Returns Discarded	x	x
Avg. Drain Size	x	x
# Creates	x	x
# Instantiates	x	x
# Concurrent Lives	x	x
Avg. Method Resp. Time for Remove		x
Active Methods	x	x
Gets from Pool	x	x

Table 8. EJB data

EJB Metrics	WebSphere 4	WebSphere 5
Returns to Pool	x	x
Drains from Pool	x	x
Avg. Pool Size	x	x
# of Invocations	x	x
# of Concurrent Requests	x	x

Table 9. JCA Connection Pool data

JCA Connection Pool Metrics	WebSphere 4	WebSphere 5
Concurrent Waiters		x
Percent Used		x
Faults		x
Avg. Wait Time		x
Avg. Use Time		x
# Managed Connections		x
# Managed Connections Created		x
# Managed Connections Destroyed		x
# Managed Connections Allocated		x
# Managed Connections Freed		x
# Connections		x
Free Pool Size		x
Pool Size		x
Percent Maxed		x

Table 10. JTA Transaction data

JTA Transaction Metrics	WebSphere 4	WebSphere 5
Global Trans Begun	x	x
Local Trans Begun	x	x
Active Global Trans	x	x
Active Local Trans	x	x
Global Trans Duration	x	x
Local Trans Duration	x	x
Global Prepare Duration	x	x
Global Before Completion Duration	x	x
Local Before Completion Duration	x	x
# Optimizations	x	x

Table 10. JTA Transaction data

JTA Transaction Metrics	WebSphere 4	WebSphere 5
Global Trans Committed	x	x
Local Trans Committed	x	x
Global Trans RolledBack	x	x
Local Trans RolledBack	x	x
Global Trans Timeout	x	x
Local Trans Timeout	x	x
Global Trans Involved	x	x
Global Commit Duration	x	x
Local Commit Duration	x	x

Table 11. JVM/System data

JVM/System Metrics	WebSphere 4	WebSphere 5
Free Memory (JVM)	x	x
Used Memory	x	x
Total Memory	x	x
Percent CPU Usage		x
Free Memory (System)		x
Avg. CPU Usage		x

Table 12. ORB Detail/Interceptor data

ORB Detail/Interceptor Metrics	WebSphere 4	WebSphere 5
Reference Lookup Time		x
Number of Requests		x
Concurrent Requests		x
Processing Time		x

Table 13. Web Application data

Web Application Metrics	WebSphere 4	WebSphere 5
# Loaded Servlets	x	x
# Reloads	x	x
Response Time	x	x
# Errors	x	x
Total Requests	x	x
Concurrent Requests	x	x

Table 14. Session Manager data

Session Manager Metrics	WebSphere 4	WebSphere 5
Created Sessions	x	x
Invalidated Sessions	x	x
Live Sessions		x
Session Lifetime	x	x
Active Sessions	x	x
No Room for New Session		x
Cache Discards		x
External Read Time		x
External Read Size		x
External Write Time		x
External Write Size		x
Affinity Breaks		x
Attempt to Activate Nonexistent Session		x
Invalidated via Time Out		x
Serializable Session Object Size		x
Time Since Last Activated		x

Table 15. Thread Pool data

Thread Pool Metrics	WebSphere 4	WebSphere 5
Thread Creates	x	x
Thread Destroys	x	x
Active Threads	x	x
Pool Size	x	x
% Time Max in Use	x	x

WebSphere - SMF

The following information provides the metrics for the WebSphere SMF data type and the application servers it supports. The tables provide information for the following modules:

- Database Connection Pools
- Enterprise Java Beans
- Server
- Servlet and Session Manager
- Web Applications

Table 16. Database Connection Pool data

Database Connection Pool Metrics	WebSphere 4	WebSphere 5
Plan Name	x	x
User Name	x	x
Last known SQL Statement	x	x

Table 17. EJB data

EJB Metrics	WebSphere 4	WebSphere 5
EJB Type	x	x
Reentrant	x	x
Methods	x	x
UUID	x	x
Method Signature	x	x
Invocations	x	x
Average Response Time	x	x
Max Response Time	x	x
Transaction Policy	x	x
EJB Roles	x	x
ejbLoad Invocations	x	x
ejbLoad Average Execution Time	x	x
ejbLoad Maximum Execution Time	x	x
ejbStore Invocations	x	x
ejbStore Average Execution Time	x	x
ejbStore Maximum Execution Time	x	x
ejbActivate Invocations	x	x
ejbActivate Average Execution Time	x	x
ejbActivate Maximum Execution Time	x	x
ejbPassivate Invocations	x	x
ejbPassivate Average Execution Time	x	x
ejbPassivate Maximum Execution Time	x	x
Average CPU Time		x
Minimum CPU Time		x
Maximum CPU Time		x

Table 18. Server data

Server Metrics	WebSphere 4	WebSphere 5
Sample Start Time	x	x
Sample End Time	x	x
Global Transactions	x	x
Local Transactions	x	x
Existing Sessions	x	x
Active Sessions	x	x
Local Existing Sessions	x	x
Local Active Sessions	x	x
Remote Existing Sessions	x	x
Remote Active Sessions	x	x
Bytes Received	x	x
Bytes Sent	x	x
Local Bytes Received	x	x
Local Bytes Sent	x	x
Remote Bytes Received	x	x
Remote Bytes Sent	x	x
Beans	x	x
HTTP Sessions	x	x
Active HTTP Sessions	x	x
HTTP Bytes Sent	x	x
HTTP Bytes Received	x	x
Total CPU Time	x	x

Table 19. Servlet and Session Manager data

Servlet and Session Manager Metrics	WebSphere 4	WebSphere 5
Sessions Created	x	x
Sessions Invalidated	x	x
Active Sessions	x	x
Minimum Active Sessions	x	x
Maximum Active Sessions	x	x
Average Sessions Lifetime	x	x
Average Invalidation Time	x	x
Finalized Sessions	x	x
Total Sessions	x	x
Minimum Live Sessions	x	x

Table 19. Servlet and Session Manager data

Servlet and Session Manager Metrics	WebSphere 4	WebSphere 5
Maximum Live Sessions	x	x

Table 20. Web Applications data

Web Applications Metrics	WebSphere 4	WebSphere 5
Response Time	x	x
Requests	x	x
Average Response Time	x	x
Maximum Response Time	x	x
Minimum Response Time	x	x
Errors	x	x
Load Timestamp	x	x

WebSphere on z/OS Only

The following information provides the metrics for the z/WAS only data type and the application servers it supports. The table provides information for the Regions module.

Table 21. Server Regions' data

Server Regions' Metrics	WebSphere 4	WebSphere 5
JVM CPU Usage	x	x
JVM Memory Usage	x	x
Database Connection Pool	x	x
Average Execution Response Time	x	x
# of Live Sessions	x	x
The following metrics provide data for both Average Response Time and Calls per Minute:		
Receive	x	x
Browse	x	x
Send	x	x
Publish	x	x

Chapter 15. Daily Statistics

Purpose

Use Daily Statistics to see snapshots of the daily use of your z/OS WebSphere application server instances.

Usage Overview

This feature helps you:

- Understand the workload on z/OS WebSphere application server instances.

User Scenarios

Scenario 1: WSAM Data Collector downtime

You must take down the Data Collector that monitors your WebSphere application server on z/OS in order to reconfigure it, but you wish to view the activity during this downtime. WSAM will not be able to collect the PMI statistics during the time the Data Collector is down. However, you can get a view of the activity on the WebSphere z/OS application server using Daily Statistics, because Daily Statistics information comes from SMF.

Notes

Note: This feature is not available for distributed (UNIX/Windows®) versions of WebSphere. This feature is not available for CICS or IMS.

Accessing the Daily Statistics

The Daily Statistics section provides daily statistics snapshots for z/OS WebSphere servers. This feature is not available for distributed (UNIX/Windows) versions of WebSphere. Every night at midnight, the Application Monitor gathers the day's SMF data for all running z/OS WebSphere instances. In addition, the system handles situations when outages occur by continuously capturing and archiving the data at the appropriate times. As a result, the system may produce more than one report per day. WSAM presents the information to the user via the Daily Statistics section. The Daily Statistics snapshots are presented in a manner similar to that found in the Server Resources section.

To open the Daily Statistics page:

1. From the top navigation, click **Performance Management > Daily Statistics**.
The Daily Statistics Selection page opens with the previous day's data.

Report Date	Sep 14, 2004 12:00:00 AM		
# of Reports	3		
Server Name	Servlet Volume	EJB Volume	Transaction Volume
ADCDPL.M2L2.M2L2.servga	0	0	-1
ADCDPL.M2L2.M2L2.servga	0	0	-1
ADCDPL.M2L2.M2L2.servga	0	0	-1

Figure 43. Daily Statistics Selection

Note: To change the date of the report, select a month, a date and a year from the drop-down menu on the left navigation.

To delete old Daily Statistics snapshots:

1. From the top navigation, click **Performance Management > Daily Statistics**. The Daily Statistics page opens.
2. Use the left navigation to select a month, day, and year under the "Delete Reports Older Than" heading.

Report Date	Sep 14, 2004 12:00:00 AM		
# of Reports	3		
Server Name	Servlet Volume	EJB Volume	Transaction Volume
ADCDPL.M2L2.M2L2.servga	0	0	-1
ADCDPL.M2L2.M2L2.servga	0	0	-1
ADCDPL.M2L2.M2L2.servga	0	0	-1

Figure 44. Daily Statistics

3. Click **Delete**.
4. Click **Yes** in the confirmation box. The system deletes all reports created earlier than the date you select.

Viewing the Daily Statistics Overview

This page displays overview information related to the selected Daily Statistics snapshot for a selected server.

To open the Daily Statistics Overview page:

1. From the top navigation, click **Performance Management > Daily Statistics**. The Daily Statistics page opens with the previous day's data.
2. If snapshots from a different date are desired, from the left navigation, select a month, day, and year under the "Enter Report Date" heading and click **Go**.

3. Click on a Server Name to view the Daily Statistics Overview, where the snapshot data will be presented.

To get more detailed information, the side navigation can be used to find out more information regarding: Server, EJBs, Servlet Session Manager, and Web Applications.

Descriptions of the information found in the detail sections of the Daily Statistics can be found in the System Resources Section of the User Guide.

Note: You can use the Servlet, EJB, and Transaction volume fields to quickly gauge workload that the application server handled for the day. This information allows you to identify servers that have behaved abnormally.

Chapter 16. System Resource Comparison

Purpose

Use the System Resource Comparison to compare a selected resource across all servers in a group.

Usage Overview

This feature helps you:

- Compare the System Resources across a group of servers.
- Understand the utilization of your application server group and its workload:
 - JVM CPU Usage
 - JVM Memory Usage
 - DB Connection Pools
 - Transaction Failure Rates
 - Thread Pools
 - EJB Activity
 - Servlet/JSP Activity
 - EJB Coverage
 - Servlet/JSP Coverage

User Scenarios

Scenario 1: Verifying Memory Utilization

You notice that memory usages for server Trade_01, in the Trade group, is very high and you want to know if this is abnormal. You perform a comparison and view the JVM Memory Usage for all the servers in the server group Trade and see that other servers in this group are not utilizing memory at the same pace. You can now go to Memory Analysis or Server Statistics Overview and begin to work out the problem.

Scenario 2: Confirming Resources in Preproduction

You have two servers with the same applications installed. Before you place them both into production, you perform a System Resource Comparison to see the difference in their resources. You see that server Quote_03 has 20 Database Connection Pools while server Quote_02 has only 10 pools. Increase the number of Database Connection Pools on server Quote_02.

Notes

Note: This feature is only applicable to PMI data. For z/OS platform, choose WebSphere 5 (PMI data) to do the comparison. This feature is not available for CICS or IMS.

To set up a System Resource Comparison:

1. From the top navigation, click **Availability > System Resource Comparison**.

The System Resource Comparison Selection page opens.

2. Select a Group and a Resource to Compare from the drop-down menu.
3. Click **OK**.

The System Resource Comparison page displays with the selected resource data specified for all the servers in the group.

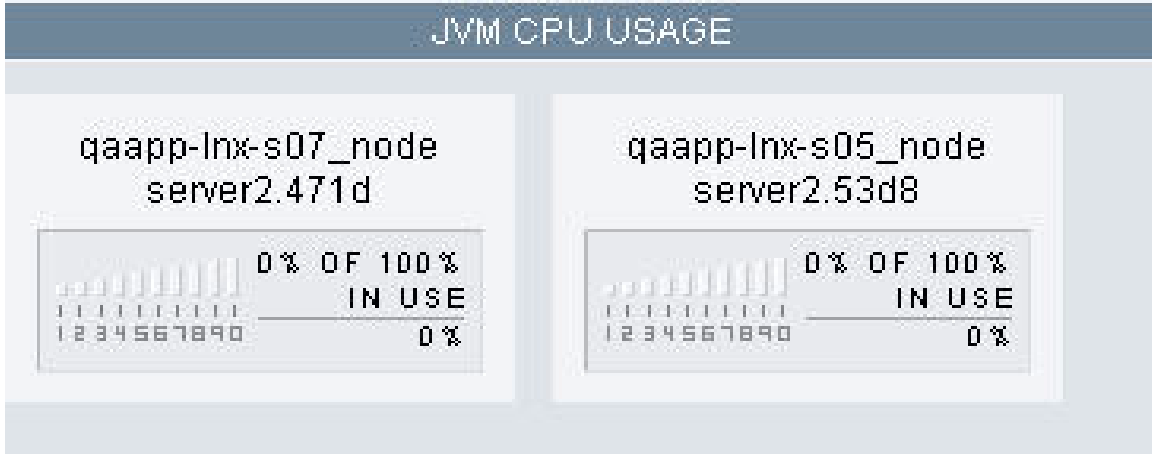


Figure 45. System Resource Comparison

Chapter 17. Performance Analysis & Reporting

Purpose

Use Performance Analysis & Reporting to analyze historical data. This helps you understand the performance of your applications and the utilization of your servers.

Usage Overview

This feature helps you:

- Isolate performance bottlenecks.
 - Find problematic method calls by drilling down from high level trends to detailed traces.
 - Use top reports to identify the “hot spots” in your application.
 - Understand the behavior of your transactions by decomposing your transaction into different parts.
 - Identify slow transactions, SQL, or MQI calls.
 - Understand where time is spent in composite transactions that span multiple application servers.
- Tune your application server.
 - View server resource trends so that you can tune your application server accordingly.
- Predict your server capacity.
 - Identify peak usage and usage patterns.
 - Decompose the workload that is being driven against your server.
 - Predict the capacity of your servers against various types of demands.
- View your server’s availability.
 - Produce SLA information for you and your management.
- Manage your reports.
- Schedule reports to show up in your email daily.
- Perform further analysis offline by exporting reports into PDF or CSV formats.
- Email your findings to your colleagues.

User Scenarios

Scenario 1: Investigating poor response time claims

Customers have been complaining about poor performance on Application A. As a performance analyst, you go into WSAM and draw up a response Trend Report for Application A for the last week to verify the customers’ claims. Once you are able to see that there indeed are instances of poor response time, you decompose the problematic period to see how different requests impact the response time. Drill down to a method trace of an actual instance of a slow transaction, and email this Trace Report to the developers so they can determine why the transaction was slow.

Scenario 2: Predicting how your servers will handle a new workload

Marketing is going to launch a new campaign to bring more visitors to your site. Your manager wants to make sure that there is enough capacity to handle the projected workload without degrading response times. As a capacity planner, you need to project how well your current servers will perform under the new workload. You create a Capacity Analysis report to compare throughput versus response time. You can use the trend line to estimate at what throughput the response time will be unacceptable.

Create Reports

Set different requirements for generating reports to analyze the performance of application servers.

Creating a Request/Transaction Analysis Report

The Request/Transaction Analysis Report provides a whole picture about the behavior of the application server. After defining the Request/Transaction Analysis Report, several reports become available: Trend Report, Decomposition report, Request Report Detail (including Detail, Summary and Worst Performers tabs) and Trace Report. Each of these reports provides more specific data for understanding the application's performance at every level.

To define a Request/Transaction Analysis Report:

1. From the top navigation, click **Performance Management > Create Reports > Application Report > Request/Transaction**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.
For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.
3. Select the Group or the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Report Filtering Options page opens. It displays the options based on the Report Type you select.
5. Set the following options to filter the records returned in the report:
 - **Metric** - the item you want to measure: Throughput per Second, Throughput per Minute, Throughput per Hour, Response Time, or CPU Time.
 - **Request/Transaction Type** - All, EJB, JSP, or Servlet.
 - **Request/Transaction Name** - Unless you know exactly what the request string is, leave the field blank to return all requests. Enter the specific request name, if you know the specific request/transaction name.
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
7. Set the parameters to restrict the data returned in your report. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
8. Click **View Report** to view the report. If you want to get a second data set for comparative analysis, click **Next** to open the Report Comparison page.
The Report Comparison page opens.

9. Select a report comparison type and view the comparison report by clicking **View Report**.

Creating a Method/Program Analysis Report

The Method/Program Analysis Report shows you the performance of the methods in the requests that have been processed by the Application Servers. After defining the Method/Program Analysis Report, a Trend Report, Decomposition report, and detailed Method Report (including Detail, Summary and Worst Performers tabs) are available.

To define a Method/Program Analysis Report:

1. From the top navigation, click **Performance Management > Create Reports > Application Report > Method/Program**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.
For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.
3. Select the Group or the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Report Filtering Options page opens. It displays the options based on the Report Type you select.
5. Set the following options to filter the records returned in the report:
 - **Metric** - the item you want to measure: Throughput Per Second, Throughput per Minute, Throughput per Hour, Response Time, or CPU Time.
 - **Method/Program** - leave the field blank to return all methods or type in the specific method name.
 - **Request/Transaction Type** - All, EJB, JSP, or Servlet.
 - **Request/Transaction Name** - Unless you know exactly what the request string is, leave the field blank to return all requests. Enter the specific request name, if you know the specific request/transaction name.
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
7. Set the parameters to restrict the data returned in your report. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
8. Click **View Report** to view the report. If you want to get a second data set for comparative analysis, click **Next** to open the Report Comparison page.
The Report Comparison page opens.
9. Select a report comparison type and view the comparison report by clicking **View Report**.

Creating a SQL Analysis Report

The SQL Analysis Report provides the information for the SQL calls’ performance in the requests that have been processed by the application server. You may also view the Trend Report, Decomposition report, and detailed SQL Report (including Detail, Summary and Worst Performers tabs) after defining the SQL Analysis Report.

To define a SQL Analysis Report:

1. From the top navigation, click **Performance Management > Create Reports > Application Report > SQL**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.
For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.
3. Select the Group and the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Report Filtering Options page opens. It displays the options based on the Report Type you select.
5. Set the following options to filter the records returned in the report:
 - **Metric** - the item you want to measure: Throughput or Response Time
 - **SQL Call** - select the correct operator from the drop-down menu.
 - **Table Name** - leave the field blank to return all tables or type in the specific table name.
 - **Request/Transaction Type** - All, EJB, JSP, or Servlet.
 - **Request/Transaction Name** - Unless you know exactly what the request string is, leave the field blank to return all requests. Enter the specific request name, if you know the specific request/transaction name.
 - **Method/Program** - leave the field blank to return all methods. Type in the specific method name if you know the method name you are looking for.
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
7. Set the parameters to restrict the data returned in your report. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
8. Click **View Report** to view the report. If you want to get a second data set for comparative analysis, click **Next** to open the Report Comparison page.
The Report Comparison page opens.
9. Select a report comparison type and view the comparison report by clicking **View Report**.

Creating an MQI Analysis Report

The MQI Analysis report provides the information for the MQI calls’ performance in the requests that have been processed by the application server. You may also view the Trend Report, Decomposition report, and detailed MQI Report after defining the MQI Analysis Report.

To define an MQI Analysis Report:

1. From the top navigation, click **Performance Management > Create Reports > Application Report > MQI**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.

For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.

3. Select the Group and the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Report Filtering Options page opens. It displays the options based on the Report Type you select.
5. Set the following options to filter the records returned in the report:
 - **Metric** - the item you want to measure: Throughput or Response Time
 - **MQI Call** - select the correct operator from the drop-down menu.
 - **Queue Manager** - leave the field blank to return all queue managers, or type in the specific queue manager name.
 - **Queue Name** - leave the field blank to return all queues, or type in the specific queue name.
 - **Request/Transaction Type** - All, EJB, JSP, or Servlet.
 - **Request/Transaction Name** - Unless you know exactly what the request string is, leave the field blank to return all requests. Enter the specific request name, if you know the specific request/transaction name.
 - **Method/Program** - leave the field blank to return all methods. Type in the specific method name if you know the method name you are looking for.
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
7. Set the parameters to restrict the data returned in your report. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
8. Click **View Report** to view the report. If you want to get a second data set for comparative analysis, click **Next** to open the Report Comparison page.
The Report Comparison page opens.
9. Select a report comparison type and view the comparison report by clicking **View Report**.

Creating a Top Report

Top Reports are a quick and convenient way to run a report for request, method, or SQL data. Top Reports provide the top 100 results records for the selected metric.

To define a Top Report:

1. From the top navigation, click **Performance Management > Create Reports > Application Report > Top Reports**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.
For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.
3. Select the Group and the Server on which you want to report from the drop-down menus.
The Report and Date Range Selection page opens.
4. Select a Top Report type from the drop-down menu.

- Set the Start Date, End Date, Start Time, and End Time. If applicable, set the Advanced Filtering to extract the data of a specific time period. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
- Click **Finish** to view the report.

The Top Report opens.

REPORT PROPERTIES

Report Name	Untitled
Report Type	Top Requests Used Analysis
Report Period	Sep 8, 2004 12:00 AM to Sep 15, 2004 12:00 AM
Server Scope	All Servers on All Groups

TOP REPORTS DATA TABLE 20 per page

1 - 20 of 100 Results 1 2 3 4 Next > Last >>

RANK	REQUEST/TRANSACTION NAME	COUNT
1	/trade/scenario	2362
2	/cyanea_one/testware/dsStateful?ttl=30&sqlStatement=1&lookup	270
3	/cyanea_one/testware/method?ttl=30&depth=1&repeat=1&map=Meth	266
4	/cyanea_one/testware/ejbStateless?ttl=30&txcommit=&lookup=St	246
4	/cyanea_one/testware/sql	246
6	/cyanea_one/testware/dsStateless?ttl=30&sqlStatement=1&looku	245
7	/cyanea_one/testware/jsp/jspRequestResult.jsp?ttl=30&map=JSP	244
8	/cyanea_one/testware/ejbStateful	242
9	/cyanea_one/testware/session?ttl=30&oneMeg=false&timeout=fal	241
10	/cyanea_one/testware/object?ttl=30&repeat=1&map=Java+Object+	238
11	/cyanea_one/testware/ejbEntity	236
12	/cyanea_one/testware/jndi?ttl=30&depth=jdbc&map=JNDI+Lookup	235
13	/cyanea_one/testware/threadkill	233
14	/cyanea_one/testware/stack?ttl=30&depth=6&repeat=1&map=Stack	227
15	/cyanea_one/testware/cpu?ttl=30&repeat=20&map=CPU+Consumer	219
16	/cyanea_one/testware/session	218
17	/cyanea_one/testware/ejb	214
18	onMessage	162
19	/cyanea_one/testware/ejbStateless?ttl=1&txcommit=&lookup=Sta	68
20	/cyanea_one/testware/object?ttl=1&repeat=1&map=Java+Object+G	60

1 - 20 of 100 Results 1 2 3 4 Next > Last >>

Figure 46. Top Report

Creating a System Resource Analysis Report

The System Resource Analysis Report gives you the information of the utilization of the memory, and database connection pools for the application servers. You may also view a Trend Report and Decomposition report after defining the System Resource Analysis Report.

Note: This feature is not available for the z/OS data collector.

To define a System Resource Analysis Report:

- From the top navigation, click **Performance Management > Create Reports > Server Reports > System Resource**.

The Create Report page opens.

- Select Yes or No to decide if you want the report to recur and click **Next**.

For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.

- Select the Group and the Server on which you want to report from the drop-down menus.
- Click **Next** to continue creating the report.

The Report Filtering Options page opens. It displays the options based on the Report Type you select.

5. Set the following options to filter the records returned in the report:
 - **Metric** - the item you want to measure: JDBC Connection Pool Size, Average % of Pool in Use, Concurrent Waiters, Average Connection Wait Time (ms), Average Connection Pool Timeouts, Amount of Free Memory (MB), and Amount of Memory Used (MB).
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
7. Set the parameters to restrict the data returned in your report. For detailed instructions on setting the parameters, see “Understanding the Date Range Settings” on page 137.
8. Click **View Report** to view the report. If you want to get a second data set for comparative analysis, click **Next** to open the Report Comparison page.
The Report Comparison page opens.
9. Select a report comparison type and view the comparison report by clicking **View Report**.

Creating a Server Availability Analysis Report

The Server Availability Analysis Report shows the percentage of the server availability. In the group situation, availability is defined as the total amount of time when one or more servers of the group are up divided by the total elapsed time.

To define a Server Availability Analysis Report:

1. From the top navigation, click **Performance Management > Create Reports > Server Reports > Server Availability**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.
For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report” on page 130.
3. Select the Group and the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Date Range Settings page opens.
5. Set the parameters to restrict the data returned in your report. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
6. Click **View Report** to view the report. If you want to get a second data set for comparative analysis, click **Next** to open the Report Comparison page.
The Report Comparison page opens.
7. Select a report comparison type and view the comparison report by clicking **View Report**.

Creating a Capacity Analysis Report

The Capacity Analysis Report provides you with the necessary information to evaluate the capacity of your system using supply and demand metrics.

To define a Capacity Analysis Report:

1. From the top navigation, click **Performance Management > Create Reports > Server Reports > Capacity Analysis**.
The Create Report page opens.
2. Select Yes or No to decide if you want the report to recur and click **Next**.
For the purpose of these instructions we are selecting No. If you want further instructions on scheduling reports, see “Creating a Scheduled Report”.
3. Select the Group and the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Report Filtering Options page opens. It displays the options based on the Report Type you select.
5. Set the following options to filter the records returned in the report:
 - **Demand Metric (X - Axis)** - Throughput per Minute and Users.
 - **Supply Metric (Y - Axis)** - System CPU (%), JVM/Process CPU (%), System Memory (MB), JVM/Process Memory (MB), JVM/Process Virtual Memory (MB), Disk Space (MB), Disk I/O (MB), Network I/O (MB), Thread Pool, Connection Pool and Response Time (ms).
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
7. Set the parameters to restrict the data returned in your report. For detailed instructions on setting the parameters, see “Understanding the Date Range Settings” on page 137.
8. Click **Next** to continue creating the report.
The Report sample page opens.
9. Click **Next** to save the report.
10. Click **Save** to save the report or click **Save & Activate** to save and activate the report at the same time.

Creating a Scheduled Report

Scheduling reports allows you to create a time for your reports to automatically activate at a time you preselect.

To define a Scheduled Report:

1. From the top navigation, click **Performance Management > Create Reports > Server Reports > Capacity Analysis**.
The Create Report page opens.
Note: We will use the Capacity Analysis report as an example. You can create a Scheduled Report from any of the available reports.
2. Select Yes to have the report recur and click **Next**.
3. Select the Group and the Server on which you want to report from the drop-down menus.
4. Click **Next** to continue creating the report.
The Report Filtering Options page opens. It displays the options based on the Report Type you select.
5. Set the following options to filter the records returned in the report:
 - **Demand Metric (X - Axis)** - Throughput per Minute and Users.

- **Supply Metric (Y - Axis)** - System CPU (%), JVM/Process CPU (%), System Memory (MB), JVM/Process Memory (MB), JVM/Process Virtual Memory (MB), Disk Space (MB), Disk I/O (MB), Network I/O (MB), Thread Pool, Connection Pool and Response Time (ms).
6. Click **Next** to continue creating the report.
The Date Range Settings page opens.
 7. Set the parameters to restrict the data returned in your report. For detailed instructions on setting the parameters, see “Understanding the Date Range Settings” on page 137.
 8. The report displays. Check the report to verify that the metrics were correctly selected. To set a schedule for the report, click **Next**.
 9. Set the schedule for the report and setup a distribution list for the people you want the report sent to when it is completed. Click **Save & Activate**.
Note: You can either save the report now and activate it later or you can save and activate the report at the same time.
 10. The Scheduled Reports page opens displaying your activated report in the list.

View Saved Reports

The following instructions show the actions you can take on saved reports.

Viewing the Reports

After defining a report other than a Top Report, there are six different reports that display various levels of detail: Trend Report, Decomposition report, Method Report, Request Report, SQL Report, and Trace Report. The reports that you have access to will vary depending on the criteria you select while creating your report. For example, on the Server and Report Type Selection page, depending on the Report Type you select, the following reports are available:

- **Request/Transaction Analysis** - displays Trend, Decomposition, Request Detail, and Trace reports.
- **Method/Program Analysis** - displays Trend, Decomposition, and Method Detail reports.
- **SQL Analysis** - displays Trend, Decomposition, and SQL Detail reports.
- **MQI Analysis**-
- **Server Availability Analysis** - displays the Trend report.
- **System Resource Analysis** - displays Trend and Decomposition reports.
- **Capacity Analysis** - displays scatter chart.

To view the reports:

1. From the top navigation, click **Performance Management >View Saved Reports**.
The Reports page opens.
2. Click **Run Report** next to the report you want to run.
The Trend Report opens first.



Figure 47. Trend report

Note: Use the left navigation to return to the Reports page, modify a report, save a report, email a link or PDF, or view a PDF.

3. Select an option from the Additional Details drop-down menu to decompose the Trend Report.
4. Click the bar displayed in the graph or a data point to view more details. The Decomposition report opens.

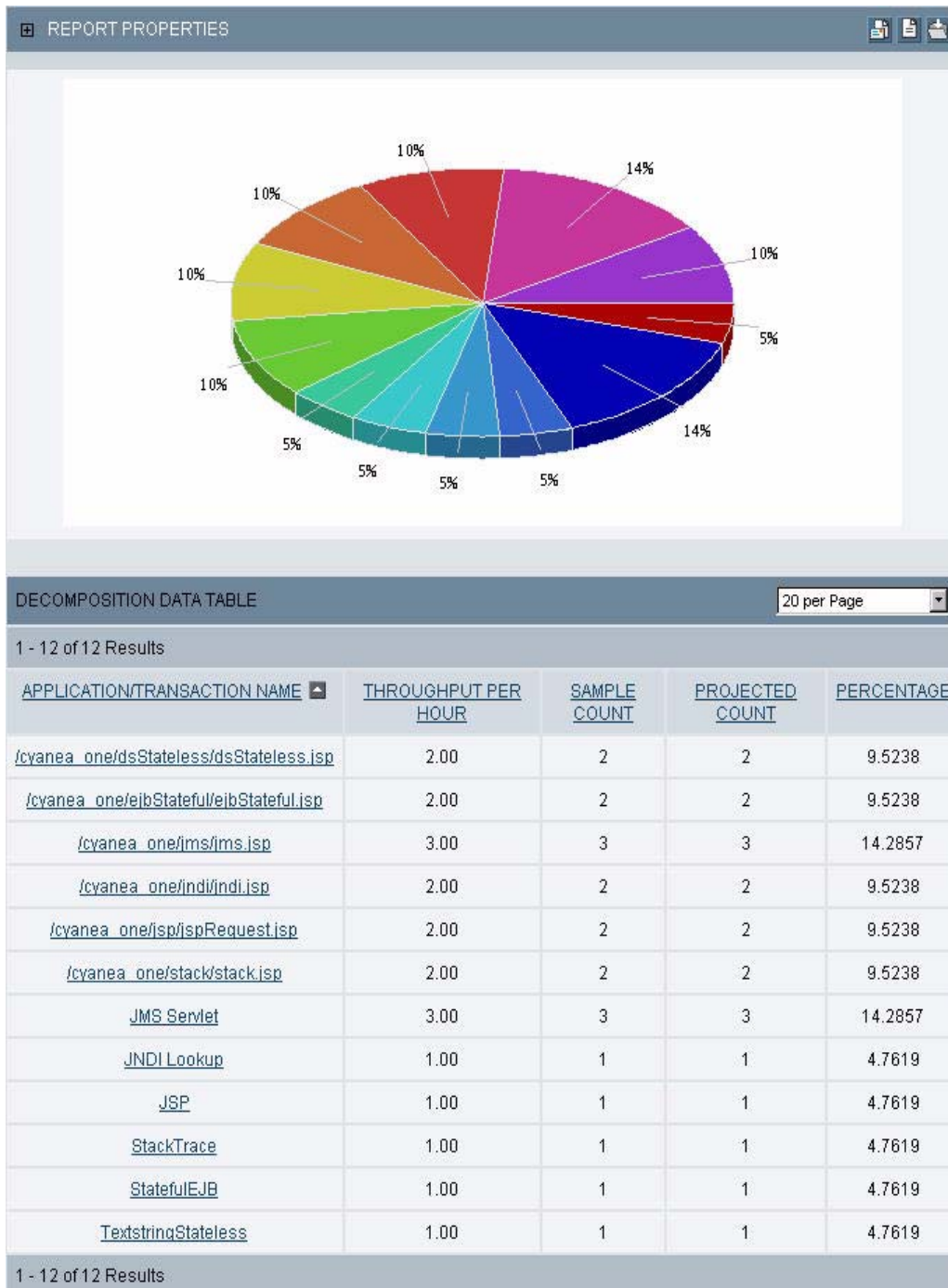


Figure 48. Decomposition report

- Click on a section of the chart or a data point to view more details.

The Request/Transaction Report Detail page opens displaying the Detail data, the Summary and the Worst Performers.

If you selected Request/Transaction Analysis as the Report Type, to access the Trace Report:

- Click on the Request Name to view the Trace Report.
- The Trace Report - Nesting Summary page opens.

The Trace Report is a form of Method Trace. See “Viewing a Method/Component Trace” on page 63 for more information on using the Trace Report.

Running a Report

Return to the Reports page to run a saved report and retrieve the current data. Additionally, you can save a report, email a link or PDF of a report, view a PDF report, export a PDF report to a comma delimited file format, for your own future reference. If you email a link, remember that the recipient must be a WSAM user with the appropriate rights to view the servers where the report runs.

To run a Report:

1. From the top navigation, click **Performance Management > View Saved Reports**.
The Reports page opens.
2. All previously defined and saved reports (except Scheduled Reports) display on the Reports Management page.
3. Click **Run Report** next to the report you want to run.

Note: The report opens displaying data based on the Metric selected on the Report Filtering Options page. The type of report and metric selected display in the page heading, for example, Trend Report – Throughput per Second Request Analysis.

Modifying a Report

After creating a report, you can modify the parameters of the report to suit your changing needs. Change the settings in the Server and Report Type Selection page, the Report Filtering Options page, the Date Range Settings page, and the Report Comparison page. Using this method, you can reuse, duplicate, and modify old reports for different application servers.

To modify a report:

1. From the top navigation, click **Performance Management > View Saved Reports**.
The Reports page opens.
2. Click **Modify** next to the report you want to modify.
The Server and Report Type Selection page opens.
3. Change the Group, Server, or Report Type selection, and click **Next**.
The Report Filtering Options page displays different options based on the Report Type you select.
Note: While you are choosing a server by navigating through the groups, it should be noted that the final group name does not affect the data to be extracted for the preparation of the report. The group name is immaterial to the selection process when the system gathers data. The report will compile all records that are generated by the chosen server regardless of which group it belongs to.
4. Select the filtering options for your report to examine and limit the type of records to include in the report.
5. Click **Next** to continue creating the report.
The Date Range Settings page opens.
6. Set the parameters to restrict the data returned in your report. For detailed instructions, see “Understanding the Date Range Settings” on page 137.
7. Click **View Report** to view the report. If you want to get a second data set, click **Next** to open the Report Comparison page.

The Report Comparison page opens.

8. Select a report comparison type and view the comparison report by clicking **View Report**.

The Trend Report opens.

Modifying a Top Report

After creating a Top Report, you can modify its parameters to suit your changing needs. Change the settings in the Server and Report Type Selection page, and the Report and Date Range Selection page. Using this method, you can reuse, duplicate and modify old reports for different application servers.

To modify a Top Report:

1. From the top navigation, click **Performance Management > View Saved Reports**.

The Reports page opens.

2. Click **Modify** next to the Top Report you want to change.

The Server and Report Type Selection page opens.

3. Change the Group, Server, or Report Type selection, and click **Next**.

Note: While you are choosing a server by navigating through the groups, it should be noted that the final group name does not affect the data to be extracted for the preparation of the report. The group name is immaterial to the selection process when data is gathered. The report will compile all records that are generated by the chosen server regardless which group it belongs to.

4. Click **Next** to modify the report type, date range, and the filtering options.

The Report and Date Range Selection page opens.

5. Select a Top Report type from the drop-down menu.

6. Set the Start Date, End Date, Start Time, and End Time. If applicable, set the Advanced Filtering to extract the data of a specific time period. For detailed instructions, see “Understanding the Date Range Settings” on page 137.

7. Click **Finish** to create the report.

The Top Report opens.

Deleting a Report

Manage your reports by keeping them up-to-date. Delete existing reports from the system that are no longer in use.

To delete a report:

1. From the top navigation, click **Performance Management > View Saved Reports**.

The Reports page opens.

2. Click **X** next to the report you want to remove.

3. A confirmation box opens. Click **OK** to delete the report.

The Reports page displays without the deleted report.

Emailing a Report/Link

You can email a PDF file of a report to either WSAM users or non WSAM users. You may also email a link of a report to a group of WSAM users. The recipient will be brought to a particular page by the link after logging in.

To email a report/PDF:

1. From the top navigation, click **Performance Management > View Saved Reports**.
The Reports page opens.
2. Click **Run Report** next to the report you want to run.
The selected report opens.
3. Click the **Email PDF** icon to email a PDF file of a report.
The Email page opens.
4. On the Email page, enter the email address of the recipient. Separate multiple addresses with a comma.
5. Click **OK** to email the report.

To email a link:

1. From the top navigation, click **Performance Management > View Saved Reports**.
The Reports page opens.
2. Click **Run Report** next to the report you want to run.
The selected report opens.
3. Click **Email Link** to email a link of a report.
The Email page opens.
4. On the Email page, enter the email address of the recipient. Separate multiple addresses with a comma.
5. Click **OK** to email the link of the report.

Note: When you email a link, the recipient must be a WSAM user with the appropriate rights to view the servers in the report.

Viewing a PDF File

You may view a PDF file of a report before you send the file to another recipient.

To view a PDF file:

1. From the top navigation, click **Performance Management > View Saved Reports**.
The Reports page opens.
2. Click **Run Report** next to the report you want to run.
The selected report opens.
3. Click the **View PDF** icon to download a PDF file of a report.
4. From the File Download window, click either **Open** to view the file immediately or click **Save** to save the file.

Exporting to a File

You may export the report to a comma delimited file format, if necessary.

To export to a file:

1. From the top navigation, click **Performance Management > View Saved Reports**.
The Performance Analysis & Reporting page opens.
2. Click **Run Report** next to the report you want to run.
The selected report opens.

3. Click the **Export to File** icon.
4. Click either **Open** to view the file immediately or click **Save** to download the file. The exported file downloads into the location you specify.

Understanding the Date Range Settings

The Date Range Settings page contains three main sections — **Date Range**, **Advanced Filtering (optional)**, **Graphing Option** and **Report Comparison**.

To set the Date Range Settings:

1. From the Date Range section, click to select a preset date range or enter a custom start date and end date for extracting only the data for the time period specified.
2. To extract the data of a specific time period, define your custom data set in the Advanced Filtering section:

DATE RANGE

Select a preset date range or enter a custom start date and end date.

Preset or

Start Date End Date

ADVANCED FILTERING (Optional)

Define your data set with accuracy using these filters.

Filters

HOUR		DAY OF THE WEEK		DAY OF THE MONTH					MONTH	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Select All	Deselect All	Select All	Deselect All	Select All	Deselect All	Select All	Deselect All	Select All	Deselect All	
<input type="checkbox"/>	00:00	<input type="checkbox"/>	Monday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	January
<input type="checkbox"/>	01:00	<input type="checkbox"/>	Tuesday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	February
<input type="checkbox"/>	02:00	<input type="checkbox"/>	Wednesday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	March
<input type="checkbox"/>	03:00	<input type="checkbox"/>	Thursday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	April
<input type="checkbox"/>	04:00	<input type="checkbox"/>	Friday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	May
<input type="checkbox"/>	05:00	<input type="checkbox"/>	Saturday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	June
<input type="checkbox"/>	06:00	<input type="checkbox"/>	Sunday	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	July
<input type="checkbox"/>	07:00	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	August
<input type="checkbox"/>	08:00	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	September
<input type="checkbox"/>	09:00	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	October
<input type="checkbox"/>	10:00	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	November
<input type="checkbox"/>	11:00	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	December

GRAPHING OPTION

Select an option to represent your data set on your graph's x-axis scale.

X-axis

Figure 49. Date Range Settings

- Unselect the hours of the day when you do not want data to display. For example, to select only data occurring between 9:00am and 5:00pm, unselect 0:00 to 08:00 hours and 18:00 - 23:00 hours.
- Unselect the days of the week when you do not want data to display. For example, to select only data occurring Monday through Friday, unselect Saturday and Sunday.

- Unselect the days of the month when you do not want data to display.
- Unselect the months of the year when you do not want data to display.

Note: By default, the Advanced Filtering section automatically selects all the options.

3. Select any of the following in the Graphing Option for analyzing certain patterns in the data based on time characteristics, or compiling large amounts of data over a long period and plotting all the points:
 - Time series in hour
 - Time series in day
 - Time series in week
 - Time series in month
 - Aggregate minute of the hour
 - Aggregate hour of the day
 - Aggregate day of the week
 - Aggregate month of the year

Note: This section is not applicable to defining a Top Report.

4. If you want to get a second data set for comparative analysis, you can compare your report to None, Stored and Secondary Range.
 - **None** - No comparison.
 - **Stored** - Compare to another report you have saved previously.
 - **Secondary Range** - Compare to a date range you select.

Chapter 18. Composite Requests

Purpose

Use Composite Request features to monitor transactions that utilize resources on more than one server.

Usage Overview

These features help you:

- Determine whether the reason a top-level request hangs is its use of resources on a different application server.
- Identify the origin (the application server and top-level request) that invoked a hanging request.
- Discover the inter-application architecture of complex workflows.

User Scenarios

Scenario 1: Discovering application architecture

Your manager asks you to provide an example of a complete transaction of an airline reservation application. This involves a web-based Java application, a CICS credit card processing application, a CICS ticket reservation application and a frequent-flyer account, which is also a CICS system.

You look in Performance Analysis & Reporting for examples of the airline reservation application, some of which will have the Composite Request indicator. Clicking on the indicator brings you to the Composite Request View of the Method Trace, which lets you navigate among these requests, so you can see which application calls which one, and by what mechanism (MQ, CTG or DPL). You can email PDFs of each request involved in the composite transaction to your manager.

Notes

Note: Your WSAM administrator must enable composite request support for all Data Collectors that participate in composite requests.

Product Overview

WSAM and Composite Requests

One aspect of WSAM is its ability to show individual requests/transactions. The benefit of Composite Requests is to identify individual requests/transactions that are related.

Finding Composite Requests: WSAM

In particular, the Composite Request Indicators appear in the following four sections of WSAM:

- Server Activity Display—View active requests/transactions on a specific server.
- In-Flight Request Search—Search for active requests/transactions on all servers, a group of servers, or a specific server.
- Performance Analysis & Reporting: Detail Reports derived from Request/Transaction Analysis Trend Reports—View the requests/transactions that comprise the Decomposition report.
- Request Detail—View the details of a request/transaction, and provide controls for changing its status (if it is an active request/transaction).

Viewing Composite Requests

Furthermore you can investigate Composite Requests in more detail through the following features:

- Composite Method Trace—Display the method traces of all requests/transactions in the Composite Request.
- Composite Stack Trace—Display the stack traces of all servers involved in the Composite Request that are still actively processing the request/transaction.

These features are described in detail in this chapter.

The Scope of Composite Request

To understand the scope of what WSAM can monitor, and how WSAM fits in, we introduce two terms: Managed Space and Composite Request Space.

- Managed Space is the term we use to describe the entire scope of what WSAM can monitor. Since WSAM can monitor servers and application servers, along with applications and J2EE components like EJBs, the Managed Space has many dimensions.
- Composite Request Space is the term we use to describe a subset of the Managed Space. Generally speaking, Composite Requests are those requests that conform to an Enterprise Application Integration (EAI) architecture, which is to say, requests for web-enabled legacy applications.

Both of these terms are described in more detail, following a review of the WSAM architecture.

WSAM Architecture: The Context of the Managed Space

The basic model of WSAM is to have a single Managing Server and many Data Collectors. The Data Collectors are dynamically controlled through the Managing Server, in terms of what data to collect, and the Data Collectors deliver their collected data to the Managing Server.

The Managing Server is the heart and brain of WSAM. It is the entity to which each of the many Data Collectors communicate, and provides the WSAM User Interface.

The Data Collectors are the eyes and ears of WSAM. For each Application Server being monitored, a Data Collector is deployed on the machine hosting the Application Server. (If a server has two application servers, then you must deploy two Data Collectors on the server in order to monitor both application servers.)

What Does It Mean to Be Managed?

Based on the preceding explanation of the WSAM architecture, we can describe in detail what is in the Managed Space.

Servers: Since WSAM Data Collectors obtain platform-level data, any server on which a Data Collector is installed will be in the Managed Space. For z/OS systems, a server is considered to be equivalent to an LPAR.

Application Servers: Since WSAM Data Collectors obtain application server-level information, any application server running in a JVM in which a Data Collector is installed will be in the Managed Space. CICS and IMS™ regions are considered to be application servers.

The architecture of WebSphere running on z/OS consists of a single application server definition and one or more application server regions. The definition and the regions, taken as a whole, are called an application server instance. What WSAM considers to be the application server depends on the context: in a few cases, the application server is either the entire application server instance (as in the case of MOD schedules,) but in most cases, the application server is an individual application server region.

Resources: WSAM monitors common resources that are made available through the application server and the J2EE APIs, such as EJB, JMS, JNDI, JDBC and JCA. If an application server is in the Managed Space, then the resources it provides are also in the Managed Space.

Applications: WSAM supports monitoring of any application which is served by an application server. If the application server is in the Managed Space, then the applications it serves will be in the Managed Space. As a corollary, standalone applications, which are not served through an application server, are not in the Managed Space.

Defining the Composite Request Space

Given an understanding of the WSAM Managed Space, we can now describe the subset of it covered by Composite Requests.

Although the Managed Space includes servers, application servers, requests and resources, the Composite Request Space covers only requests, and only a subset of those requests in the Managed Space.

It is necessary to discuss Enterprise Application Integration (EAI) architecture, in order to clearly describe how requests interact and to define the Composite Request Space.

Enterprise Application Integration

The fundamental notion of Enterprise Application Integration (EAI) is to make a legacy system accessible through the web. From the J2EE perspective, this means that an initial request, served by a J2EE application server, invokes a resource on a legacy system through the JCA API.

EAI Transaction Terminology: When describing EAI transactions, the name we use for the initial J2EE request is the Home Request, and its server is called the Home Server. We call the legacy transaction a Participating Request, and its server is called a Participating Server. There may be more than one Participating Request if the legacy resource invokes resources on other legacy systems.

EAI Transaction in the Managed Space: The first thing to note is that, if the legacy system is CICS, and a CICS Data Collector has been installed, then the legacy system will be within the Managed Space. Similarly, if the legacy system is IMS, and an IMS Data Collector has been installed, then the legacy system will be within the managed space. The J2EE application server will be within the Managed Space if its Data Collector is installed.

This means that, as part of normal WSAM operations, both the Home Request and the Participating Requests appear in WSAM. However, without the Composite Request enhancement, these requests appear independently, and there is not explicit indication that they are part of the same transaction.

Not only does the Composite Request enhancement make this relationship explicit, it also provides diagnostic tools, like Method Trace and Stack Trace, that you can apply across all requests in the Composite Request.

Monitoring CICS Transactions

The CICS Data Collector monitors all program invocations on the managed CICS Region, whether they come through a dumb terminal, Distributed Program Link (DPL), EXEC CICS START, or through the CICS Transaction Gateway (CTG).

Furthermore, for transactions invoked through CTG, it does not matter how CTG was accessed, which can include a variety of interfaces. However, WSAM will not track all such transactions as Composite Requests.

CICS and IMS Transactions in Composite Requests

Even though all transactions on a CICS or IMS Region in the Managed Space will appear in WSAM, they will not necessarily be treated as part of a Composite Request, even if they program invoke programs on other Regions.

A transaction on a CICS or IMS Region will be part of a Composite Request if it meets the following criteria:

- The CICS or IMS Region is in the Managed Space.
- The Home Server is in the Managed Space.
- The application server that serves the Home Request is a J2EE application server, and is in the Managed Space.
- For CICS: The application on the Home Server uses ECI to access CTG. (This includes applications that use CCI as their JCA resource adapter, since CCI uses ECI.)
- The ECI invocation is synchronous.
- The COMMAREA of the CICS program invocation has at least 84 bytes of available space.
- For IMS: The application on the Home Server uses ICHJ to access IMS connect.

If any of these criteria are not met for an EAI request, then WSAM will not identify the request as being part of a Composite Request. However, the core WSAM features will still be available for whatever parts of the transaction are in the Managed Space.

For example, if an application in C++ invokes a CICS program on a CICS Region in the Managed Space, through CTG, the CICS program will appear as a request within WSAM, but the C++ application request will not appear within WSAM, since WSAM does not monitor C++ applications. In this case, WSAM will not identify the CICS transaction as belonging to a Composite Request.

Likewise, if a Java application uses EPI to access CTG, WSAM will not track the EAI as a Composite Request, even if the application is in the Managed Space. In this case, the requests on both the J2EE application server and in the CICS Region will appear in WSAM, but will appear independently, and will not be identified as a Composite Request.

The final condition, based on the application's use of the COMMAREA, is due to the methodology of tracking Composite Requests, which involves use of the

COMMAREA. In practice, it is rare that program invocations use so much of the COMMAREA that there isn't room for this correlation information. In these exceptional cases, WSAM does not attempt to identify the EAI as a Composite Request, and the individual requests appear in WSAM as independent requests.

Multiple Hops

Composite Request are not restricted to single-hop transactions.

In particular, Composite Requests include cases where CICS programs make DPL calls to other CICS Regions. When such a call is made, we say that the depth of the Composite Request increases. WSAM can track requests with no limit to the depth of transaction "hops."

For IMS, any events with the same message tag from any IMS Region within IMS Network appear as a single transaction.

In addition, Composite Requests can include up to 100 Participating Requests made directly by each Home or Participating Request. Although Composite Requests can include an unlimited depth of "hops," Composite Requests place a limit on the number of trackable calls made by any single request.

Configuring Data Collectors that use MQ

If you are monitoring Composite Requests for applications that use MQ as a mechanism to bridge J2EE and CICS or IMS, then you must configure each participating Data Collector to monitor MQ.

Note: These instructions assume your Data Collectors have already been configured.

To enable MQ monitoring on a Data Collector within the Application Monitor

1. Open the Application Monitor.
2. Click the **Administration** tab on the top navigation.
3. Select **Server Management > Data Collector Configuration**
4. Follow the **Configuration List** link in the left navigation.
5. Locate the application server in the Associated Server column of the Configuration List table and click the **Modify** icon for that row.
6. Check the **Enable MQ** checkbox (if it is not already checked.)
7. Fill in the **Exclude (Queue)** and **Exclude Override (Queue)** lists to specify which queues you want to monitor.
8. Click the **Save** button.

Finding Composite Requests

To find Composite Requests, you can use features of WSAM to locate participating requests/transactions. Requests/transactions that participate in a Composite Request are distinguished from single-server requests/transactions by the Composite Request Indicator.

This section describes how to find Composite Requests in WSAM. The subsequent chapters describe the Composite Request features in detail.

Identifying Composite Requests in WSAM

The Composite Request Indicator

There are three ways to locate requests/transactions within WSAM. Two features help you locate active requests/transactions:

- In-Flight Request Search
- Server Activity Display

A third feature helps you locate completed requests/transactions:

- Performance Analysis & Reporting

Each of these features produces a list of requests/transactions which may participate in a Composite Request. The presence of the following icon indicates that a request/transaction participates in a Composite Request:



Figure 50. The Composite Request Indicator

Using the Composite Request Indicator

This section describes how to find requests/transactions that participate in Composite Requests, by looking for the Composite Request Indicator in the following WSAM features:

- In-Flight Request Search
- Server Activity Display
- Performance Analysis & Reporting

In-Flight Request Search

To search for in-flight requests/transactions that participate in Composite Requests, use the In-Flight Request Search.

Enter in the search argument and, if you like, restrict the search to a group of servers or a particular server. In-Flight Request Search displays a list of the active requests/transactions that contain the search string, on the servers you specified.

In addition to the normal output (which includes the Server Name, Client Request/Transaction, Start Date Time, Thread ID and Total Resident Time,) WSAM identifies those requests/transactions that are part of a Composite Request by displaying the Composite Request Indicator.

The following image shows an In-Flight Request Search result that includes both single-server requests/transactions and requests/transactions that participate in Composite Requests.

Server Activity Display

To search a server for resident requests/transactions that participate in Composite Requests, use the SAD.

Once you select a server, SAD displays a list of the active requests/transactions on that server (in the Threads section of the page).

In addition to the normal output (which includes the Thread ID, Priority, Client Request, Resident Time, Last Known CPU, Idle Time, Thread Status, Last Known Class and Last Known Method,) WSAM identifies those requests/transactions that are part of a Composite Request by displaying the Composite Request Indicator.

The following image shows an SAD result that includes both single-server requests/transactions and requests/transactions that participate in Composite Requests.

Performance Analysis & Reporting

To search for completed requests/transactions that participated in Composite Requests, use Performance Analysis & Reporting. You must start with a Trend Report, drill down to a Decomposition Report, and then to a Detail report in order to find individual requests/transactions that are part of Composite Requests.

Note: Performance Analysis & Reporting displays the Composite Request Indicator only for home requests, and not for the other participating requests/transactions.

The following procedure describes how to locate Composite Requests using Performance Analysis & Reporting.

To locate Composite Requests using the Performance Analysis & Reporting:

1. View a Request/Transaction Analysis Trend Report for servers that you believe may have served home requests of Composite Requests.

Choose appropriate **Report Filtering Options** and **Date Range Settings**.

A Trend Report is displayed.

2. Choose an appropriate Decomposition option (**Additional Detail** selection) and time period.

A Decomposition Report is displayed.

3. View the requests/transactions that comprise the Decomposition Report by selecting an appropriate segment of the Decomposition Report.

A Detail Report is displayed.

The resulting Detail Report displays a list of the requests/transactions included in the segment of the Decomposition Report you selected.

In addition to the normal output (which includes the Request/Transaction Name, Request/Transaction Type, Response Time, CPU Time, Server Name, Timestamp and Number of Records,) WSAM identifies that a request was a home request of a Composite Request by displaying the Composite Request Indicator next to its Request/Transaction Name.

Viewing Composite Requests

Once you have located individual requests/transactions that participate in Composite Requests (by using WSAM features described in the preceding section,) you can click on these request/transactions' Composite Request Indicator to access the Composite Request features.

This chapter describes how to use the Composite Request features of WSAM.

Composite Request Features

The Composite Request features include the following:

- Composite Method Trace—Displays the interrelated method traces across all requests involved in the Composite Request.
- Composite Stack Trace—Displays a continuous stack trace of all servers involved in the Composite Request which are still actively processing the request/transaction.

Viewing a Composite Method Trace

The Composite Method Trace shows the interrelated method traces of all requests involved in a Composite Request.

You arrive on this page by clicking a Composite Request Indicator. Composite Request Indicators are located next to requests that participate in Composite Requests on pages that display individual requests/transactions, such as In-flight Request Search. (There is no way to access this page directly from the top navigation.)

The Composite Method Trace page provides method trace data for one request at a time, and allows you to navigate to the other participating requests.

To view a Composite Method Trace:

1. From the top navigation, click **Problem Determination > Search Your Servers > In-Flight Request Search**.

The In-Flight Request Search page displays active requests. Requests that participate in Composite Requests are identified by the Composite Request Indicator.

2. To view the Composite Method Trace, click a request's Composite Request Indicator. The Composite Method Trace page opens to the method trace for that request, within the Flow View tab.

The screenshot displays the 'COMPOSITE METHOD TRACE' interface. At the top, there's a navigation bar with tabs like 'ADMINISTRATION', 'AVAILABILITY', 'PROBLEM DETERMINATION', 'PERFORMANCE ANALYSIS', 'LOGOUT', and 'HELP'. Below this, a 'MENU' box contains links for 'Select New Thread', 'Composite Stack Trace', and 'Composite Method Trace'. The main content area is titled 'TRANSACTION OVERVIEW' and includes a 'Threshold Highlighter' section with a 'Delta Elapsed Time' input set to 5 (ms). Below that, a 'COMPLETE EVENT FLOW VIEW' section shows a table of results. The table has columns for 'Event Type', 'IMS Region', 'Destination Name / Transcode', 'Elapsed Time', and 'Delta Elapsed Time'. A highlighted row shows 'IMS Link Receive Before' with an elapsed time of 36ms and a delta of 33ms. Other rows include 'IMS Terminal Receive Before', 'IMS Terminal Receive After', 'IMS OTMA Input Edit Before', and 'IMS OTMA Input Edit After'.

Event Type	IMS Region	Destination Name / Transcode	Elapsed Time	Delta Elapsed Time
IMS Terminal Receive Before	M1L4.ADCDPL.M1L4.M2L3.I74A.71 (L3)	IVTNO	0	0
IMS Terminal Receive After	M1L4.ADCDPL.M1L4.M2L3.I74A.71 (L3)	IVTNO	1	1
IMS OTMA Input Edit Before	M1L4.ADCDPL.M1L4.M2L3.I74A.71 (L3)	IVTNO	3	2
IMS OTMA Input Edit After	M1L4.ADCDPL.M1L4.M2L3.I74A.71 (L3)	IVTNO	3	0
IMS Link Receive Before	M1L4.ADCDPL.M1L4.M2L3.I74B.bba (L3)	IVTNO	36	** 33 **
IMS Link Receive After	M1L4.ADCDPL.M1L4.M2L3.I74B.bba (L3)	IVTNO	43	** 7 **

Figure 51. Composite Method Trace

The Transaction Overview section at the top of the page displays summary information about each request that participates in the Composite Request:

- Server Name
- Platform
- Start Time

- Resident Time (ms)
- Status

Click on any server name displayed in the Transaction Overview to arrive at the Request Detail page for that request, which provides further information about that request/transaction and allows you to take action on the thread if it is still active.

The Flow View section, located below the Transaction Overview section, displays the method trace content for a participating request.

See “Viewing a Method/Component Trace” on page 63 for more information about using method trace features, including Search. Within the Composite Method Trace, the Search applies only to the current request, and not the entire Composite Request.

Arrows are used within the method trace data to identify relationships with other requests. Click the links or use the controls within these arrows to navigate to the method trace page for associated requests.

The Modify View section lets you toggle the view between a complete view with all the method data, and a view with only the relationship arrows. The name of the server on which the current request is running appears in parenthesis after the title (either Complete Flow View or Composite Interactions).

Composite Method Trace and Monitoring On Demand

Like the (single-server) Method Trace, the availability of method-level data is contingent upon the configuration of the Data Collectors: they must be running at L3 in order to provide full method-level data, and they will provide Nested Request data if they are running at L2.

Since the monitoring levels of Data Collectors are independent, it is possible that method-level data is available for some, but not all, servers participating in a Composite Request. The Composite Method Trace presents all data it has, which means that the level of data presented from server to server may vary.

Navigating Within Composite Method Trace

From the left navigation in Composite Method Trace, you can proceed to the Composite Stack Trace page. The Composite Stack Trace link is only available if the request is still active. This feature is described in the subsequent sections of this chapter.

To return to In-Flight Request Search (or to proceed there if you arrived from some other feature), follow the Select New Request link in the left navigation. The Select New Request link always proceeds to In-Flight Request Search, regardless of how you arrived at Composite Request Detail.

Viewing a Composite Stack Trace

The Composite Stack Trace page displays the stack traces of each server involved in the Composite Request that are actively processing their request/transaction.

Note: The Composite Stack Trace is primarily useful for debugging Composite Requests that are hanging, since there will be no stack trace data available if a Composite Request has completed by the time you access it.

To debug completed Composite Requests, use Performance Analysis & Reporting to locate an example, and then use the Composite Method Trace.

You arrive on Composite Stack Trace by following the **Composite Stack Trace** link on the Composite Method Trace page. The stack trace data is gathered in real time at the point when you follow the Composite Stack Trace link.

To view a Composite Stack Trace:

1. From the top navigation, click **Problem Determination > Search Your Servers > In-Flight Request Search**.
The In-Flight Request Search page displays active requests/transactions. Requests/transactions that participate in Composite Requests are identified by the Composite Request Indicator.
2. To view the Composite Method Trace for a request/transaction that is participating in a Composite Request, click that request/transaction's Composite Request Indicator.
The Composite Method Trace page for that Composite Request opens.
3. Click **Composite Stack Trace** in the left navigation.
The Composite Stack Trace page opens and displays the stack traces of the servers that are actively executing participating requests/transactions.

The Composite Stack Trace page has two parts. The top portion of the page displays:

- Snapshot Date
- Snapshot Time
- Start Time
- Number of Active Requests
- Home Server Name
- Home App Server Type
- Resident Time

The second portion includes the stack trace content. Within the bottom portion of the page, the stack trace of each server is preceded by a line that includes:

- Server Rank
- Server Name
- Operating System
- Application Server Type

The servers are listed in order of Server Rank, which is the order in which the servers are invoked within the context of the Composite Request.

Each individual server's stack trace is a list of method/program calls, starting with the method/program being executed when the stack trace is obtained, and continuing in Last-In-First-Out order. The data includes the Depth, Class Name and Method/Program Name.

Authorization and Composite Requests

Authorization is enforced in WSAM in two ways: by feature and by server. Feature-based authorization limits access to top-level features based on the role assigned to a user. Assuming a user has access to a feature, the server-based authorization may further limit access to data about servers based on which group(s) a server is assigned to, and which groups the user has authority to view.

Since Composite Requests involve more than one server, the effects of server-based authorization play out in the following scenario.

A Composite Request's home request is on server A (which is in group A) and invokes a participating request on server B (which is in group B). There are two users who need to investigate this Composite Request: User A has access to servers in group A but not group B, and user B has access to servers in group B but not group A.

Assuming that each user uses In-Flight Request Search to locate the requests, the results for each user will differ, since the In-Flight Request Search limits results to those requests executing on servers in groups the user has access to. This means that user A will see only request A and user B will see only request B.

In both cases, the Composite Request indicator will appear next to the request, and will link to a similar Composite Request Detail page. However, the contents of the Composite Request Detail page will be different for each user.

Both users will see the complete Composite Request, including the Home Request on server A and the Participating Request on server B. However, the users will not have access to the Request Detail pages of all requests: User A will have access to the Home Request on server A (the request name will be linked), but not to the Participating Request on server B (the request name will not be linked). User B will not have access to the Home Request on server A (the request name will not be linked) but will have access to the Participating Request on server B (the request name will be linked).

Chapter 19. WebSphere Studio Workload Simulator Integration

WebSphere Studio Workload Simulator (WSWS) and WebSphere Application Monitor (WSAM) share data via HTTP calls. There are two points of integration between WSWS and WSAM. The first point exposes Real-Time Statistics. This allows WSWS to gather benchmarking information from the application servers. The second point of integration allows WSWS to automatically create a report on the behalf of the user, upon the completion of a benchmarking run.

In order to integrate with WSWS, you must create a WSAM user called *bmuser* (benchmark user). Authorization for calls coming from WSWS will be based on the permissions that the WSAM user *bmuser* has been configured with. There will not be any authentication; all calls are assumed to be associated with the *bmuser*. Requests sent before the user is created will automatically be rejected.

The WSWS integration supports two features: Real-Time Statistics Query and Auto-Generated Benchmark Report.

Note: This feature is not available for the Distributed platform.

Real-time Statistics Query

WSWS initiates workload to a WebSphere Studio Application. While the application is processing load WSWS sends all the performance data to WSAM. WSWS sends a URL to WSAM, for example,

```
http://localhost/cyanea_one/webapi/rtquery?groupname=wswsgroup
```

After receiving the URL, WSAM returns statistics to WSWS on all the servers in the group in XML format.

Auto-generated Benchmark Report

The Auto-Generated Benchmark Report provides a way to programmatically create a report for a particular server group. WSWS calls this feature immediately after a benchmarking run completes, in order to automatically create a report for the user. In other words, WSWS sends a URL, for example:

```
http://localhost/cyanea_one/webapi/createreport?duration=123456789&groupname=wswsgroup
```

Once the URL is received, WSAM creates a report in Performance Analysis and Reporting. You can then log onto WSAM and view the report.

Chapter 20. Audit Trails

Purpose

Audit trails provide a means for tracing user actions in the system. This helps with both accountability and troubleshooting.

Usage Overview

This feature helps you:

- Keep track of key events.
 - Trace events to a user.
 - Trace events to a date and time.
 - Follow the changes in the definitions of complicated tasks in features such as Trap & Alert Management and Performance Analysis & Reporting.

User Scenarios

Scenario 1: Verifying high server response time

Upon returning from vacation, you see that response time is high for most of the servers in the group ABC. You review the servers in the group and realize that two servers are missing. You enter the audit trail to find who took the servers offline. You contact the employee who took the servers offline and learn that the servers are being upgraded.

Scenario 2: Verifying report definition change

In your role as a Capacity Planner you run a report and notice that its results are abnormal. Upon review, you see the report's definition has changed. You go ask the Administrator to go into the audit trail to find out who changed the report's definition. You can now consult with your colleague about why the report's definition has changed.

Accessing the User Audit Trail

The User Audit Trail is a text file that contains a record of user activity.

To open the User Audit Trail:

1. Depending on your platform, go to the `/opt/cyaneaone/logs` directory on the server hosting WSAM.
2. In a text editor appropriate to your platform, open the `cyaneaone_audit.log` file. See the sample file below.

Note: The `/tmp` directory and `cyaneaone_audit.log` are configurable WSAM defaults. If you change the defaults during installation, these instructions will not apply.

Table 22. Sample from User **Audit Trail Log**

audit.log
2002-02-23 14:43:50,470 [audit] - htang logged in
2002-03-13 15:20:15,791 [audit] - htang modified thread 350009416 priority 4 ->5
2002-03-13 17:17:42,077 [audit] - htang session timed out

Chapter 21. Request Mapper

Purpose

Use the Request Mapper to customize how requests are named within the Application Monitor. Also, use the Request Mapper to display user names associated with requests.

Usage Overview

This feature helps you:

- Distinguish among requests that otherwise would have the same request name.
- Aggregate requests which otherwise would have distinct request names.
- Identify the User IDs under which requests run.

User Scenarios

Scenario 1: Aggregating Across Distinct ORS

The application you are monitoring uses a distinct URI to represent each specific application function, such as log in, check out, or log out. You wish to analyze all these requests as a single application. Use the Request Mapper to populate the Request Name field with a common application name.

Scenario 2: Differentiating a Uniform ORS

You are monitoring an application that uses session variables to represent the underlying function, while using the same request name throughout these different interactions. You want to compare the performance of different application functions, such as log in, check out, or log out, so you use the Request Mapper to assign each function a distinct request name.

Notes

Note: This feature is not available for IMS.

Data Used by the Request Mapper

Request Name

The Request Name allows the user to assign alternate request identifiers that are more meaningful and appropriate to the chosen programming model of the application.

The Request Name is provided because the Request String is just one way of identifying requests. There is data that is within the request that is not represented by the Request String. Furthermore, requests can be rather cryptic, so mapping them to something more immediately recognizable or understandable is useful.

For example, a Web request can be mapped by:

- URI: `/account/login`

- Servlet Class Name: **com.cyanea.web.AccountServlet**
- Struts Class Name:

```
http://www.cyanea.com/account/execute/login.do -->
com.cyanea.web.account.LoginAction
```
- Custom Naming Scheme: **account.login**

When the installed Request Mapper is invoked, data is passed into this plug-in class to assist the custom code developer to make a decision. This includes the Request Object and the Session Object in the case of a URL based request.

Application Name

The Application Name allows you to assign request identifiers that classify their requests into different applications. It is a means to aggregate different ORS into an application label.

The Application Name allows you to analyze their historical data from an application perspective.

For example, requests can be mapped to different names such as the following:

- Account Management
- Web Trading
- Order Management

User IDs

The Application Monitor has the ability to capture, display and store the user ID of a request that comes into the application server. By default, the user ID is captured by calling the following method:

```
javax.servlet.http.HttpServletRequest.getRemoteUser()
```

If your application stores user IDs in the session, configuration will be required. User IDs are defined as web-side identifiers of who initiated the transaction/request.

To capture the user ID from the session, you need to enable the data gathering from the session, and specify the attribute in the session that contains the user ID.

To enable the data gathering from the session, update the data collector properties as follows:

```
com.cyanea.mapper.userid.source=session
```

To capture the attribute called accountname from the session, update the data collector properties as follows:

```
com.cyanea.mapper.http.userid.attributename=accountname
```

Default Request Mapping Behavior

From the application server perspective, there are two major types of requests: JSP and Servlet. These calls come either from a Web server, or from an application server other than itself.

We call this request, generally expressed in the form of a string, the Original Request String (ORS). The ORS is composed of the URI plus the query string.

While a unique ORS can be used to represent a specific application function such as log in, check out, and log out, this may not always be the case. Other styles of application design utilize different programming techniques to represent the underlying function, while still maintaining a simple, uniform ORS throughout a series of interactions. When monitoring applications that use such a design, you can use the Request Mapper to distinguish among these different interactions that use the same ORS.

In addition, when performing workload characterization and understanding resource consumption, an analyst may sometimes find that it is neither possible nor effective to break down consumption simply by ORS, especially if there are too many of them. Aggregation of consumptions based on classification of ORS is more desirable.

The Request Mapper functionality is designed to resolve these types of problems. When an application server receives a request (ORS), the Request Mapper will allow the ORS to be rewritten into two other strings before it is passed on to WSAM:

- Request Name
- Application Name

If no request mapper is used, the Application Monitor will map the incoming ORS onto a Request Name and an Application Name using the following rule:

Request Name = ORS without the host name
 Application Name = URI of ORS

In-flight Request Search is conducted on the Request Name. Server Activity Display uses Request Name for the display. Performance Analysis & Reporting performs decomposition by Application Name.

Writing and Deploying a Request Mapper

Request Mapper is highly sensitive to performance since it is frequently invoked. A poor-performing Request Mapper can have an adverse effect on the overall performance of the application server in terms of Servlet response time as well as CPU costs.

For compilation, follow the standard Java compilation procedure.

For deployment, make sure the new class file is in a location specified in the classpath and restart the application server. A system property called `cyanea.requestmapper` should be set to the implementing class. For example,

```
.. -Dcyanea.requestmapper=com.cyanea.mapper.RequestMapperExample
```

Java docs and an example follow:

Package `com.cyanea.mapper`

Table 23. Interface Summary

Interface Summary	
<code>MappedRequest</code>	Interface used for providing the WSAM system with a Distinguishable Request String (DRS) and a Collapsible Request String (CRS) about a particular Servlet request.
<code>RequestMapper</code>	WSAM recognizes JSP and Servlet requests on an application server.

Interface MappedRequest

public interface MappedRequest

Interface used for providing the WSAM system with a DRS and a CRS about a particular servlet request.

Table 24. Method Summary

Method Summary	
java.lang.String	<u>getCRS ()</u>
java.lang.String	<u>getDRS()</u>

Interface RequestMapper

public interface RequestMapper

WSAM recognizes JSP and servlet requests on an application server. These requests are normally identified throughout the WSAM system using the URI of the request. In some situations, such as when a Struts design paradigm is used, a particular URI will be used to handle different types of business requests.

WSAM provides this interface as a mechanism for modifying WSAM's default behavior of using the URI to describe the request. An implementation of this interface can be installed by registering the classname with the Java executable as a system property.

To install, specify the system property "**cyanea.requestmapper**" with the implementing class as the value.

For example:

```
-Dcyanea.requestmapper=com.cyanea.mapper.RequestMapperExample
```

Table 25. Method Summary

Method Summary	
<u>MappedRequest</u>	mapRequest (java.lang.String servletClassName, javax.servlet.http.HttpServletRequest request) This stateless method should translate a servlet classname and a URL into a MappedRequest object.

Sample Request Mapper - mapRequest

```
public MappedRequest mapRequest(java.lang.String servletClassName,  
javax.servlet.http.HttpServletRequest request)
```

This stateless method should translate a servlet classname and a URL into a MappedRequest object. Any RequestMapper class should attempt to execute this method as quickly as possible, due to the fact that it lies directly in the path of the application server thread execution.

- **Parameters:**
 - **ServletClassName** - the name of the ServletClass handling this request.
 - **request** - the HttpServletRequest object for this request.
- **Returns:** an instance of MappedRequest indicating the DRS and CRS to be used by the WSAM system.

Request Mapper Example (1):

```
/*  
 * $RCSfile: MappedRequestExample.java,v $  
 * Copyright (c) 2002 Cyanea Systems Corp.  
 * All rights Reserved.  
 */  
  
package com.cyanea.mapper;  
  
public class MappedRequestExample implements MappedRequest {  
    private String CRS;  
    private String DRS;  
  
    /** Creates a new instance of MappedRequestExample */  
    public MappedRequestExample(String myCRS,String myDRS) {  
        CRS = myCRS;  
        DRS = myDRS;  
    }  
  
    public String getCRS() {  
        return CRS;  
    }  
  
    public String getDRS() {  
        return DRS;  
    }  
}
```

```
}
```

Request Mapper Example (2):

```
/*
```

```
 * $RCSfile: RequestMapperExample.java,v $
```

```
 * Copyright (c) 2002 Cyanea Systems Corp.
```

```
 * All rights Reserved.
```

```
*/
```

```
package com.cyanea.mapper;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
public class RequestMapperExample implements RequestMapper {
```

```
    /** static MappedRequest instance for welcome page requests
```

```
    */
```

```
    private static final MappedRequest welcomeRequest;
```

```
    /** static MappedRequest instance for quote page requests
```

```
    */
```

```
    private static final MappedRequest quoteRequest;
```

```
    /** static MappedRequest instance for buy page requests
```

```
    */
```

```
    private static final MappedRequest buyRequest;
```

```
    /** static MappedRequest instance for sell page requests
```

```
    */
```

```
    private static final MappedRequest sellRequest;
```

```
    /** static MappedRequest instance for portfolio page requests
```

```
    */
```

```
    private static final MappedRequest portfolioRequest;
```

```
    /** static MappedRequest instance for account page requests
```

```
    */
```

```
    private static final MappedRequest accountRequest;
```

```
    /** static MappedRequest instance for update page requests
```

```
    */
```

```
    private static final MappedRequest updateRequest;
```



```

/**
 * Static class variables are used to avoid continuous object creation
 * of redundant information on a per-client-request basis. An
 * unsynchronized, read-only HashMap can also be used for looking up
 * MappedRequest instances to gain a performance increase.
 */

static {
welcomeRequest = new MappedRequestExample("Welcome Page","welcome");
quoteRequest   = new MappedRequestExample("quote","quote");
buyRequest     = new MappedRequestExample("trade","buy");
sellRequest    = new MappedRequestExample("trade","sell");
portfolioRequest = new MappedRequestExample("overview","portfolio");
accountRequest = new MappedRequestExample("account","account");
updateRequest  = new MappedRequestExample("account","updateAccount");

}

/** Creates a new instance of RequestMapperExample */
public RequestMapperExample() {
}

/**
 * This example checks the HttpServletRequest object for the GET or POST
 * parameter "map". If the parameter "map" is not found, "action" is
 * used. This "action" string, is then used to look up the corresponding
 * MappedRequest object. If no MappedRequest object is found, a new
 * object is created and returned. This should be avoided, as it can be
 * an expensive operation.
 */
public MappedRequest mapRequest(String servletClassName,
HttpServletRequest request) {
    String action = request.getParameter("map");

    if ( action == null) {

```

```

        action = request.getParameter("action");
        if ( action == null )
            return welcomeRequest;
    }
    /* A HashMap lookup could also be performed here instead of iterating
    * a list of string comparisons. If a list of strings comparison are
    * used, it is desirable to list the most common action first.
    */
    if ( "quote".equals(action) )
        return quoteRequest;
    else if ( "buy".equals(action) )
        return buyRequest;
    else if( "sell".equals(action) )
        return sellRequest;
    else if( "portfolio".equals(action) )
        return portfolioRequest;
    else if( "account".equals(action) )
        return accountRequest;
    else if( "updateAccount".equals(action) )
        return updateRequest;
    else
        return new MappedRequestExample(action,action);
}
}

```

Glossary

A

Accumulated CPU. The approximate CPU time utilized under a thread since the current request started.

Acknowledge Mode. The acknowledge mode is one of the following: AUTO_ACKNOWLEDGE
CLIENT_ACKNOWLEDGE
DUPS_OK_ACKNOWLEDGE NO_ACKNOWLEDGE.

Action. The activity the system will take when a trap is triggered, such as email or method trace.

Activate Non Exist Sessions. The number of requests for a session that no longer exists.

Active. 1. Determines if the consumer is active.

2. Determines whether the consumer has a message listener set up, or if a synchronous receive is in progress.
3. Determines whether the subscription is being used by a durable subscriber.

Active Global Transactions. The number of concurrently active global transactions.

Active Local Transactions. The number of concurrently active local transactions.

- Active methods.** 1. The average number of concurrently active methods.
2. The average number of invocations being processed concurrently for all methods.

Active Requests. The requests being serviced.

Active Sessions. 1. The number of concurrent, active sessions.

2. The number of communication sessions active during the interval.
3. The current number of HTTP sessions actively referenced in the server during the interval.

Active Threads. (1) The number of concurrent, active threads. (2) The thread that is servicing a request.

Active Thread Count. The number of activated thread.

Active Thread Group Count. The number of activated thread group.

Additional Detail. 1. A dynamically generated list based on the selections made by the user when creating a report.

2. A drop-down menu for viewing the detailed report broken down by different criteria in a Trend Report.

Admin Server. The name of the administration server that oversees the functions of the application servers.

Admin Server Host. The address on which the admin server is listening for connections.

Admin Server Listen Port. The port on which the admin server is listening for connections.

Affinity Breaks. The number of HTTP session affinities broken, not counting intentional breaks of session affinity.

Alert Condition. The definition of when to trigger the selected action(s).

AMC Name. The AMC Name of the bean activated by the container (only the rightmost 256 characters are recorded).

Application Server. The name of the application server monitored by a Data Collector.

Application Server IP Address. The IP address for the selected application server.

- Application Server Name.** 1. The name of the selected application server.
2. The Sysplex node name concatenated with the server instance name.
 3. The name of the server where the session is executing.

Application Server Start Time. The time that the application server started running.

- Application Server Uptime.** 1. The amount of time that has elapsed since the application server started running.
2. The system highlights this number on the Server Statistics Overview page when the amount of time that has passed since the application server started running exceeds the threshold value.

Application Trap. A trap based on data from an application.

Archive Agent. Accepts the aggregated data from a publish server and performs fast data archiving into the database for reporting purposes.

Authentication. Verifies the identity of a user who logs into the Application Monitor.

Authoritative Date/Time Stamp. The authoritative date/time when data was frozen.

Authoritative Only. The file only exists on the authoritative server.

Authoritative Server. The server in a server group against which up to 10 other servers in the group are compared (the Comparison Servers).

Authoritative Size. The size of the file found on the authoritative server.

Average Active Usage. The running average of connections that are active in the Connector Pool, since the pool was last shrunk.

Average CPU Usage. The average percentage that the CPU has been busy since the server was started.

Average Create Time. The average method response time for creations in milliseconds.

Average Drain Size. The average number of objects discarded in each drain. Applies to entity and stateless beans.

Average Execution Time. The average amount of time, in milliseconds, of all invocations of a servlet to date.

Average Garbage Collection Duration. The average duration of a garbage collection call.

Average Heap Size after Garbage Collection. The average dynamic storage for a procedure after inactive data is deleted.

Average Invalidation Time. The average time required to invalidate HTTP sessions.

Average Method Response Time. The average response time, in milliseconds, of all invocations of the remote interface for this bean.

Average Method Response Time for Create. The average time, in milliseconds, it takes to create a bean, including load time.

Average Method Response Time for Remove. The average time, in milliseconds, for a beanRemove call, including the time at the database.

Average Pool Size. The average number of objects in the pool. Applies to entity and stateless beans.

Average Remove Time. The average response time, in milliseconds, for removes.

Average Response Time. The average elapsed time between entering a request and receiving a response.

Average Sessions Lifetime. The average lifetime of invalidated HTTP sessions.

Average Time between Garbage Collection Calls. The average time (in seconds) between two successive garbage collection calls.

Average Time In Use. The average time of the connection pool that is in use.

Average Time Wait for Lock. The average time that a thread waits for a lock.

Average Use Time. The average time, in milliseconds, a connection is used by a request.

Average Wait Time. The average time, in milliseconds, a request waits for a connection.

Average Waiting Threads. The average number of threads concurrently waiting for a connection.

B

Baseline Definition. The threshold to which a server group's average response time is compared.

Baseline Indicator Settings. The percentage above the baseline that you determine to be slow or very slow. "Slow response" means the present response time is between 26% and 50% of the baseline; "very slow response" means the present response time exceeds 50% of the baseline. When the response time reaches Indicator 1, an orange indicator displays on the Application Overview page; a red indicator means the response time has exceeded Indicator 2 and the system is very unhealthy.

Baseline Response Time. The historical response time, displayed on the Application Overview page, to which the current response time is compared.

Baseline Response Time Sample Duration. The number of days used to determine the average Baseline Response Time.

Bytes Received. The number of bytes transferred to the server from all clients.

Bytes Sent. The number of bytes sent from the server to all clients.

Bytes Threshold Time. The amount of time in the threshold condition since the last reset.

C

Cache Discards. 1. The number of session objects that have been forced out of the cache. Applicable only for persistent sessions.

2. The total number of statements discarded because the statement cache is at its maximum size.

Cancel Request. A method for terminating application requests from the system. Cancel Request terminates the request by throwing a run-time exception. All necessary clean-up will occur accordingly.

Capacitive Increment. The initial capacity configured for the Connector connection pool.

Change Priority. A feature that lets you raise or lower the priority of a thread, by selecting a different priority number. Priority 1 is the lowest and priority 10 is the highest.

Change Thread Status. A feature that lets you freeze the execution of a thread so you can investigate the problem further, and then re-activate it when the problem is resolved.

CICS Transaction Gateway. This page lists all the CTGs that the selected J2EE server has contacted.

Class. A collection of related data and methods (operations).

Class Path. The pathname where the Class is stored.

Client ID. The client ID for the connection/ durable subscriber.

Client Request. The request by a client for a particular server resource. This resource is often a Web page or a Java application.

Client Request Start. The start date and time for the current request.

Community. A string that is part of the SNMP protocol.

Comparison Date/Time Stamp. The date/time of the comparison.

Comparison Only. The file only exists on the Comparison Server.

Comparison Servers. The servers whose installed binaries are compared to those on the authoritative server.

Component ID. An ID assigned by the system for identification.

Composite Request Transactions. The requests that conform to an Enterprise Application Intergration architecture. The requests for web-enabled Legacy applications. It allows the user to monitor transactions that utilize resources on more than one server.

Concurrent Actives. The average level as a function of time of bean instances of the home that are in the ready state (active beans). A measure of server activity.

Concurrent Lives. 1. The average number of concurrently live beans.

2. The average level, as a function of time, of bean objects that exist in runtime, whether the objects are active or pooled (instantiated but not destroyed). A measure of how many resources the home interface consumed.

Concurrent Requests. 1. The number of requests that are concurrently processed by the ORB.

2. The number of requests that are concurrently processed by servlets.

Concurrent Waiters. The average number of threads concurrently waiting for a connection.

Condition. A user-defined criteria that is part of a trap definition.

Configuration Name. The name of the configuration you apply to the Data Collector.

Configuration Profile. This parameter provides the name of the Configuration Repository of the Kernel.

Connected Kernel. The IP address and port number for the Kernel.

Connection Delay Time (ms). The average time (in milliseconds) necessary to get a connection from the database. This is how long it takes to get a physical connection from the database. It is calculated as summary time to connect divided by the number of connections.

Connection factory. A connection factory is an object whose sole purpose is to create connection objects. When an application needs a connection, it asks the connection factory to "manufacture" a connection object.

Connection Factory Name. The configured Connection Factory Name for the Connection Factory using this Connector connection pool.

Connection Pool Faults. The number of faults (e.g. time-out) in a connection pool.

Connector Pool Name. 1. The configured Logical Name for a Connection Factory (using this Connector connection pool).

2. The name of the connection pool a SQL statement belongs to.

3. The name of the connection pool for a leaked connection.

Container Thread Pool. The current number of threads in a container.

Context Root. The context root (context path) for a Web application.

Contrast Options. A second data set used for the purpose of comparative analysis.

Cookies & Attributes. The name and contents of the cookies associated with a session.

CPU Speed. How fast the CPU processes the runtime environment comparison.

CPU Utilization (% , Last Hour). The percentage of CPU being utilized in last hour.

Created Sessions. The number of sessions that were created.

CRS. Collapsible Request String.

Current Total CPU Time. The total CPU time used to process the current (active) request.

Current Total Elapsed Time. The total time that has elapsed since a request began executing.

D

Daemon Thread Group. A thread group which contains the program that runs unattended to perform continuous or periodic functions, such as network control.

Data Collector. A component of the Application Monitor. Software that runs alongside an application server and captures information regarding the internal workings of the application server.

Data Collector Controller. Controls the behavior of a Data Collector, including the monitoring level, filter list, and enable or disable status.

Data Collector Listen Port. The port that clients of the Data Collector use to communicate with the Data Collector.

Data Collector Uptime. The amount of time that has passed since the Data Collector started running.

Data Grouping. Aggregates a data set based on a selected time interval, i.e., month, date of the month, day of the week, and hour of the day.

Data Interval. Part of the user-specified definition of a report. The distance between points on the X axis of a report.

- Database Connection Pool.**
1. A group of database connections. A new request is assigned a free connection from the pool. Upon completion of the request, the system returns the connection to the pool.
 2. An indicator that displays the number of connections in use and the total number of connections in the pool, for a selected Application Server.

Database Connection Pool Name. The name of the Database Connection Pool.

Database Name. The name of the database that the connection is associated with.

Date Range. The start and end dates for the report.

Decomposition Report. A report that provides a breakdown of the Trend Report by a user-selected criteria.

Default Data Collector Configuration. The configuration assigned to new Data Collectors to capture information regarding the applications running inside the application server.

Default Monitoring Level. The monitoring level used by all servers when they initially connect to the Application Monitor as well as application servers (or groups of application servers) whose monitoring level is not explicitly set. The default monitoring level for all platforms except z/OS is L2 (Problem Determination Mode). For the z/OS platform, the default monitoring level is L1 (Production Mode).

Destination Name. The destination for the consumer.

Device Host Name. The name or address of the network management machine being sent SNMP messages.

Drains from Pool. The number of times the daemon found the pool was idle and attempted to clean it. Applies to entity and stateless beans.

DRS. Distinguishable Request String.

Durable. Determines whether the consumer is durable.

Durable Subscribers. With a Durable JMS Subscriber, messages are persisted by the JMS system when the subscriber is not available, normally in a database store. When the durable subscriber becomes available, the JMS server will provide them with the messages that the subscriber missed due to its unavailability.

E

EAI. Enterprise Application Integration. It refers to the plans, methods, and tools aimed at modernizing, consolidating, and coordinating the computer applications in an enterprise. EAI involves integrating an enterprise's new and existing applications.

EAR File. Enterprise Archive File. The number of Enterprise Archive files on the application server.

EJB. Enterprise Java Bean. Component architecture for the development and deployment of object-oriented, distributed, enterprise-level applications. Applications written using the EJB architecture are scalable, transactional, and secure.

EJB Activity. The amount of EJB calls made for the last hour with a 5 minute refresh rate.

- EJB Coverage.**
1. The distribution of EJB invocations in the last hour, by EJB Home name.
 2. The graphical representations of the most frequently accessed EJBs.

EJB Home. The name of the Enterprise Java Bean method.

EJB Name. The name of an EJB component.

EJB Role. The list of EJB Roles associated with the method separated by a semicolon ";" up to 256 characters.

EJB Type. The bean's type (CMP entity bean, BMP entity bean, stateless session bean, and stateful session bean).

EJB Volume. The number of times that EJB methods were invoked on the Application Server.

ejbActivate Average Execution Time. The average execution time of the method `ejbActivate`.

ejbActivate Invocations. The number of `ejbActivate` invocations.

ejbActivate Maximum Execution Time. The longest execution time of the method `ejbActivate`.

ejbLoad Average Execution Time. The average execution time of the method `ejbLoad`.

ejbLoad Invocations. The number of `ejbLoad` invocations.

ejbLoad Maximum Execution Time. The longest execution time of the method `ejbLoad`.

ejbPassivate Average Execution Time. The average execution time of the method `ejbPassivate`.

ejbPassivate Invocations. The number of invocations of the method `ejbPassivate`.

ejbPassivate Maximum Execution Time. The longest execution time of the method `ejbPassivate`.

ejbStore Average Execution Time. The average execution time of the method `ejbStore`.

ejbStore Invocations. The number of invocations of the method `ejbStore`.

ejbStore Maximum Execution Time. The longest execution time of the method `ejbStore`.

Entity Bean. An enterprise bean that represents persistent data maintained in a database. An entity bean can manage its own persistence or it can delegate this function to its container.

Enterprise Overview. Feature that displays the availability, aggregated by server groups, of all applications running on the application servers in a server group.

Errors. The number of errors encountered by a servlet.

Exclude (Classname). A list of classes that will not be monitored unless they are part of the Exclude Override (Classname).

Exclude Override (Classname). A subset of classes in the Exclude (Classname) that will be monitored.

Execution Time High. The amount of time, in milliseconds, of the single longest invocation of the servlet.

Execution Time Low. The amount of time, in milliseconds, of the single shortest invocation of the servlet.

Execution Time Total. The amount of time, in milliseconds, of all invocations of the servlet.

Existing Sessions. The number of communication sessions that exist at the end of the interval.

External Read Size. The size of the session data read from the persistent store. Applicable only for (serialized) persistent sessions.

External Read Time. The time (in milliseconds) taken when reading the session data from a persistence store. For multi-row, the metrics are for the attribute; for single row, the metrics are for the whole session. Applicable only for persistence sessions. This metric is not available for applications that do not serialize data.

External Write Size. The size of session data written to persistent store. Applicable only for (serialized) persistent sessions.

External Write Time. The time (in milliseconds) taken in writing the session data from persistent store. Applicable only for (serialized) persistent sessions.

F

Failures to Reconnect. The number of cases when a connection pool attempted to refresh a connection to a database and failed. Failure may happen because of database unavailability or a broken connection to the database.

Faults. The number of faults (e.g. time-out) in the connection pool.

File Name Match. The file names only matched. They are unlikely to be the same.

File Name/Path/Size Match. Files on comparison servers whose file name, path and size, but not timestamp matched a file on the authoritative server; these files are likely to be the same.

Finalized Sessions. The number of sessions that were finalized.

First Join Time. The time a component first joined the Kernel.

Fixed Date. An interval, specified by a start and end date, for which average response times are calculated (for each five minute period of the day) and used as the baseline against which current response times are compared on the Overview pages.

Fixed Response Time. A response time against which all your current response times on the Application Overview will be compared.

Force GC. Force Garbage Collection. When this option is enabled, the JVM will perform a garbage collection before taking a heap dump.

Free Memory. 1. The free memory in JVM runtime.

2. The snapshot of free memory, in KB.

Free Pool Size. The number of free connections in the pool.

Full Match. Files that are likely to be identical to each other based on matching file name, path, size and file system timestamp.

Full Pathname / Size Match. Files that are likely to be identical to each other based on matching pathname and file size.

G

Garbage Collection. Java automatically reclaims any memory that is no longer needed for reuse through a process called Garbage Collection. An object is considered garbage when there are no longer references to it stored in variables, the fields of any objects, or the elements of an array.

Garbage Collection Delay. The amount of time the system should wait prior to taking a second heap snapshot.

Gets Found. The number of times a retrieve found an object available in the pool. Applies to entity and stateless beans.

Gets from Pool. The number of calls retrieving an object from the pool. Applies to entity and stateless beans.

Global before Completion Duration. The average duration of before_completion for global transactions.

Global Commit Duration. The average duration of commits for global transactions.

Global Prepare Duration. The average duration of prepares for global transactions.

Global Transaction Duration. The average duration of global transactions.

Global Transactions. The number of global transactions run through and initiated by the server instance during the interval.

Global Transactions Begun. The number of global transactions begun on the server.

Global Transactions Committed. The number of global transactions committed.

Global Transactions Involved. The number of global transactions involved on the server.

Global Transactions Rolled Back. The number of global transactions rolled back.

Global Transactions Timeout. The number of global transactions timed out.

Group Name. 1. The name of the group.

2. All servers that belong to the group will display on the Server Statistics Overview page.

Group Overview. This page provides a high-level overview of activity for each server in the group.

H

Heap. A heap is an area of pre-reserved computer main storage (memory) that a program process can use to store data in some variable amount that won't be known until the program is running.

Heap Size. The amount of memory allocated to JVM.

I

Idle Time. The time that a request has been idling, plus any unaccounted CPU time not captured by the Application Monitor.

Initial Capacity. The initial capacity configured for a Connector Connection Pool.

Installed Binary. A file deployed to a server. In a server farm, it is important that all the files are the same version.

Instance Name. The name of the WebSphere server instance.

Interceptors. A callback code that is executed when an ORB request enters or exits the process space.

Interrupted. The system stopped the thread.

Interval Start/End. The snapshot start time and end time.

Invalidated Sessions. The number of sessions that were invalidated.

Invalidated via Time-Out. The number of sessions that are invalidated via timeout.

Invocations. The number of times the method was invoked during the interval.

IP Address. The IP address of the application server.

J

Java Policy. The pathname where the Java Policy is stored.

JDBC Connection Pools. The number of JDBC connections available on the selected application server.

JDBC Driver Version. The version of the JDBC driver, in the format of concatenating the Driver class name with 'major: XX, minor: YY'.

JDBC Operation Timer and JDBC Operation Time. The amount of time, in milliseconds, spent executing in the JDBC driver.

JNDI Name. The configured JNDI Name for a Connection Factory. The name of the Connection Factory using a Connector Connection Pool.

JSP. Java Servlet Page. A server-side technology. Java server pages are an extension to the Java servlet technology. JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the design and display of the page.

JSP Coverage. The distribution of servlet/JSP requests in the last hour, by servlet/JSP name.

JVM. Java Virtual Machine. A self-contained operating environment that executes pre-compiled Java byte code.

JVM CPU Delta. 1. The amount of CPU time that a JVM used since the last page refresh.

2. The system highlights this number on the Server Statistics Overview page when the JVM CPU Delta exceeds the threshold value.

JVM CPU Usage. The current CPU utilization of the JVM space itself.

JVM CPU%. 1. The percentage of time that the JVM platform was using CPU.

2. The system highlights this number on the Server Statistics Overview page when the JVM CPU % exceeds the threshold value.

JVM Heap Size. The size of the heap that is available to the JVM.

JVM ID. The JVMID of the server.

JVM Memory Usage. 1. The amount of memory, in MB, used by the JVM of an application server.

2. The system highlights this number on the Server Statistics Overview page when the JVM memory usage exceeds the threshold value.

JVM Memory Utilization (% , Last Hour). The percentage of JVM Memory being utilized in last hour.

JVMPI. Java Virtual Machine Profiler Interface. A two-way function call interface between the Java Virtual Machine and an in-process profiler agent.

JVM / Region CPU Delta. The difference between the current JVM / Region CPU and its last refreshed data.

JVM / Region CPU %. The utilization percentage of JVM / Region CPU.

JVM Thread. The basic unit of program execution in the Java Virtual Machine. A process can have several threads running concurrently, each performing a different job. When a thread has finished its job, it is suspended or destroyed.

K

Kernel. A component of the Application Monitor that acts as a directory service that keeps track of which components have joined or left the network.

Kernel Codebase. The URL where Application Monitor components download binaries from the Kernel.

L

L1 (Production Mode). This monitoring level provides Availability Management, System Resources, and basic request data. Use this level for servers with high volume transactions, stable operations, and simple transactions.

L2 (Problem Determination Mode). This monitoring level provides Production level monitoring of advanced request data, including CPU information. The JVMPI is enabled on the corresponding JVMs. Use this level for high volume transactions in an environment that is occasionally unstable, with simple to complex transactions.

L3 (Tracing Mode). This is the most powerful monitoring level. Therefore, only this level utilizes all the reporting elements available. For example, in the Tracing mode, the Server Activity Display shows additional data for the following columns: Accumulated CPU and Idle Time. In addition, on the Request Detail page, the Method Trace function is available. Use this level to get diagnostics and detailed workload characterization.

Last Access Time. The last time a client sent a request associated with a session.

Last Contract Renewal Time. The most recent renewal time of the contract with the Kernel.

Last Known Action. The name of the last action accessed by the current request.

Last Known Class Name. The name of the last class accessed by the current request.

Last Known Method. The name of the last method accessed by the current request.

Last Known SQL Statement. The last SQL statement accessed by the current request.

Library Path. The pathname where the library is stored.

Listen Address. The address on which a server listens for connections.

Listen Port. The port on which a server listens for connections.

Live Sessions. The number of live sessions concurrently in cache.

Load Timestamp. The timestamp when a servlet was loaded.

Local Active Sessions. The number of active local communication sessions attached and active within the server instance, during the interval.

Local Before Completion Duration. The average duration of `before_completion` for global transactions.

Local Bytes Received. The number of bytes transferred to the server from all locally attached clients.

Local Bytes Sent. The number of bytes transferred from the server to all locally attached clients.

Local Commit Duration. The average duration of commit for local transactions.

Local Existing Sessions. The number of existing local communication sessions attached and active within the server instance during the interval.

Local Transaction Duration. The average duration of local transactions.

Local Transactions. The number of local transactions initiated by the server instance during the interval.

Local Transactions Begun. The number of local transactions begun on the server.

Local Transactions Committed. The number of local transactions committed.

Local Transactions Involved. The total number of global transactions involved at the server (begun and imported).

Local Transactions Rolled Back. The number of local transactions rolled back.

Local Transactions Timeout. The number of local transactions timed out.

Log File Name. The Log File used by the Resource Adapter for a Connector Connection Pool.

Logging Enabled. The Log File used by the Resource Adapter for a Connector Connection Pool.

M

Managed Space. This is a term we use to describe the entire scope of what WSAM can monitor. Since WSAM can monitor servers and application servers, along with applications and J2EE components like EJBs, the managed spaces has many dimensions.

Max Capacity. The maximum capacity configured for a Connector Connection Pool.

Max Inactive Interval. The maximum time interval, in seconds, that a servlet container will keep a session open between client accesses.

Maximum Active Sessions. The maximum number of active HTTP sessions during an interval.

Maximum Inactive Interval. The maximum time interval, in seconds, that the servlet container will keep a session open between client accesses.

Maximum Live Sessions. The maximum number of live HTTP sessions during an interval.

Maximum Method Records. The maximum number of method entry/exit records maintained by the Application Monitor for a request. The records will be over written when they reach this value starting with the oldest. The default value is 10,000.

Max Priority. The highest rank assigned to a thread that determine its precedence in processing a request.

Maximum Response Time. The maximum response time, measured in milliseconds.

MD5. A unique numeric signature that is different for files whose contents are different, even if their creation dates and file names coincide.

Memory Leak. A memory leak is the gradual loss of available computer memory when a program (an application or part of the operating system) repeatedly fails to return memory that it has obtained for temporary use. As a result, the available memory for that application or that part of the operating system becomes exhausted and the program can no longer function. For a program that is frequently opened or called or that runs continuously, even a very small memory leak can eventually cause the program or the system to terminate. A memory leak is the result of a program bug.

Memory Leak Candidate. Java classes and objects that are likely to be causing a memory leak.

Memory Leak Confirmation. The Memory Leak Confirmation Report helps you detect memory leaks in your system. Try various comparison metrics until you determine the cause of the leak. If there is over 24 hours of data available, your report will show the last 48 hours. Otherwise, the report will display the last 60 minutes of data.

Memory Usage. The amount of memory being used by the JVM process.

Message Back Out Count. The number of backed out messages that failed to be delivered to the bean's `onMessage` method. Applies to Message Driven beans.

Message Count. The number of messages delivered to the bean's `onMessage` method. Applies to Message Driven beans.

Message Dispatcher. The Message Dispatcher sends out emails of performance reports and trap results, as well as SNMP messages.

Messages Threshold Time. The amount of time in the threshold condition since the last reset.

- Method.** 1. A function defined in a class. A class can contain data and methods. Methods are operations that are performed on data.
2. The type of HTTP request with valid values of Get or Post.
 3. The number of associated methods.

- Method Signature.** 1. Methods may have the same name but accept different arguments. An example of a uniquely "callable" method would be classname+methodname+methodsignature.
2. The name of the method including its signature (only the leftmost 512 characters are recorded).

Method Trace. A Method Trace is the path of execution for a request. The trace includes entry and exit for methods in the thread, as well as the entry and exit for any embedded methods.

Method Trace Data. Each Method Trace contains entry and exit records including the Method Name, Date, Time, Elapsed Time, and CPU Time.

- Metric.** 1. The item you want to measure: Throughput per Second, Throughput per Minute, Throughput per Hour, Response Time, or CPU Time.
2. The item you want to measure: Pool Size, Concurrent Waiters, Average Wait Time, Faults, Percentage Pool Usage, Physical Connections, Connection Handles, JVM Free Memory, and JVM Memory Used.

Minimum Active Sessions. The minimum number of active HTTP sessions during an interval.

Minimum Life Sessions. The minimum number of live HTTP sessions during an interval.

Minimum Response Time. The minimum response time, in milliseconds.

MIPS. Million Instructions per Second. This is an estimated computation to give an indication of the platform CPU power. This computation is based on an empirical formula derived from the SRM (System Resources Manager) service units/second factor.

MOD. Monitoring on Demand.

Monitoring Level. In the Application Monitor, the user has the ability to select between three levels of monitoring for a server or set of servers: L1 (Production mode), L2 (Problem Determination mode), and L3 (Tracing mode.)

MQBACK. A MQ API to back out a MQ transaction.

MQBEGIN. A MQ API to begin a MQ transaction.

MQCLOSE. A MQ API to close a queue.

MQCOMIT. A MQ API to commit a transaction.

MQCONN(X) Average Response Time. A MQ API to make queue manager connection.

MQDISC. A MQ API to disconnect a queue manager connection.

MQGET Average Response Time. The average response time of a MQ API to get a message from a queue.

MQINQ. A MQ API to inquire a queue attributes.

MQOPEN. A MQ API to open a queue.

MQPUT(1) Average Response Time. A MQ API to put a message in a queue.

MQSET. A MQ API to set a queue attribute.

Stored Procedure. A block of procedural constructs and embedded SQL statements that is stored in a database and that can be called by name. Stored procedures allow an application program to be run in two parts, one on the client and the other on the server, so that one call can produce several accesses to the database.

N

No Local. The noLocal Boolean for the durable subscriber.

No Room for New Session. The number of times that a request for a new session can not be handled because it would exceed the maximum session count.

Number of Activates. The number of times beans were activated (applies to Entity and stateful session beans).

Number of Activations. The number of beans made active.

Number of Active Connections . The current total active connections.

Number of Active Connections High. The peak number of active connections in a Connector Pool since the pool was instantiated.

Number of Active Servers. The current total number of alive servers in a cluster.

Number of Active Transactions. The number of active transactions on a server.

Number of Allocates. The total number of connections allocated.

Number of Beans in Use. The number of beans currently in use during the session (active or ready state).

Number of Bytes Current®. 1. The current number of bytes stored in the destination.

2. The current number of bytes stored on the JMS server.

3. The current number of bytes received by the durable subscriber.

Number of Bytes High. The peak number of bytes stored in the destination/JMS server since the last reset.

- Number of Bytes Pending.** 1. The number of bytes pending (uncommitted and unacknowledged) by the consumer/durable subscriber/producer.
2. The number of bytes pending (uncommitted and unacknowledged) stored on the JMS server or in the destination.
 3. The number of bytes pending (uncommitted or unacknowledged) for the session.

- Number of Bytes Received.** 1. The number of bytes received by the consumer or the session since the last reset.
2. The number of bytes received on the JMS server since the last reset.
 3. The number of bytes received in the destination since the last reset.

Number of Bytes Sent. The number of bytes sent by the producer or the session since the last reset.

Number of Cache Accesses. The number of times the cache has been accessed.

Number of Cache Hits. The number of times a bean is looked up and successfully found in the cache.

Number of Cached Beans. The number of beans currently cached.

Number of Connection Consumers Current . The current number of connection consumers for the session pool.

Number of Connection Consumers High. The peak number of simultaneous connection consumers for the session pool.

Number of Connection Consumers Total. The total number of connection consumers made by the session pool since the last reset.

Number of Connections Created Total. The total number of Connector connections created in this Connector Pool since the pool was instantiated.

Number of Connections Destroyed Total. The total number of Connector connections destroyed in this Connector Pool since the pool was instantiated.

Number of Connections Matched Total. The total number of times a request for a Connector connections was satisfied via the use of an existing created connection since the pool was instantiated.

Number of Connections Rejected Total. The total number of rejected requests for a Connector connections in this Connector Pool since the pool was instantiated.

Number of Connections Total. The total number of JDBC connections in the JDBCConnectionPoolRuntimeMBean since the pool was instantiated.

- Number of Consumers Current.** 1. The current number of consumers accessing the destination.
2. The current number of consumers for the session.

- Number of Consumers High.** 1. The peak number of consumers accessing the destination since the last reset.
2. The peak number of consumers for the session since the last reset.

- Number of Consumers Total.** 1. The total number of consumers accessing the destination since the last reset.
2. The total number of consumers instantiated by the session since the last reset.

- Number of Creates.** 1. The number of times beans were created.
2. The total number of connections created.
 3. The number of create calls.

Number of Destinations Current. The current number of destinations for the JMS server.

Number of Destinations High. The peak number of destinations on the JMS server since the last reset.

Number of Destinations Total. The number of destinations instantiated on the JMS server since the last reset.

- Number of Destroys.** 1. The number of times bean objects were freed.
2. The total number of connections destroyed.

Number of Errors. The total number of errors in a servlet/JSP.

Number of Foreign Fragments Dropped. The number of fragments that originated in foreign domains/cluster that use the same multicast address.

Number of Fragments Received. The total number of multicast messages received on this server from the cluster.

Number of Fragments Sent. The total number of multicast fragments sent from a server into a cluster.

Number of Free Connections Current. The current total free connections.

Number of Free Connections High. The peak number of free connections in a Connector Pool since the pool was instantiated.

Number of Garbage Collection Calls. The number of garbage collection calls.

Number of Idle Beans. The number of idle beans in a pool that are available for use.

Number of instantiations. The number of times the system creates the bean objects.

Number of Invalid Login Attempts Total. The cumulative number of invalid logins attempted on the server.

Number of Invalid Login Users High. The peak number of users with outstanding invalid login attempts for the server.

Number of Invocation Total. The total number of servlet invocations.

Number of Leaked Connections. A connection that was checked out from the connection pool but was not returned to the pool by calling close.

Number of loaded servlets. The number of servlets that were loaded.

Number of Loads. The number of times the system loaded bean data.

Number of Lock Entries. The number of entries that are currently locked.

Number of Lock Managers Accesses. The number of times the Lock Manager is accessed. It applies to the beans that have exclusive locking specified.

Number of Locked Users Current. The number of currently locked users on the server.

Number of Login Attempts While Locked Total. The cumulative number of invalid logins attempted on this server while the user was locked.

Number of Managed Connections. The number of ManagedConnection objects in use for a particular EIS product name.

Number of Managed Connections Allocated. The total number of connections allocated.

Number of Managed Connections Created. The total number of connections created.

Number of Managed Connections Destroyed. The total number of connections destroyed.

Number of Managed Connections Freed. The total number of connections freed.

Number of Messages Current. 1. The current number of messages in the destination.

2. The number of messages still available by the durable subscriber.

3. The current number of messages stored on the JMS server.

Number of Messages High. The peak number of messages in the destination since the last reset.

Number of Messages Pending. 1. The number of messages pending (uncommitted and unacknowledged) by the consumer/durable subscriber/producer. Pending messages are over and above the current number of messages. A pending message is one that has either been sent in a transaction and not committed, or that has been received and committed or acknowledged.

2. The number of messages pending stored on the JMS server.

3. The number of messages pending for the session.

Number of Messages Received. The number of messages received by the consumer/ the session or received on the destination since the last reset.

Number of Messages Sent. The number of messages sent by the producer/ session since the last reset.

Number of Multicast Messages Lost. The total number of in-coming multicast messages that were lost according to the server.

Number of Objects Allocated. The number of objects allocated.

Number of Objects Freed. The number of objects freed.

Number of Objects in Heap. The number of instance data in storage.

Number of Objects Moved. The number of objects moved.

Number of Online / Total. The total number of CPU currently enabled.

Number of Open Sessions Current. The current total number of open sessions in the component.

Number of Open Sessions High. The peak number of the total number of open sessions in the server.

Number of Open Sockets Current. The current number of sockets registered for socket mixing on the server.

Number of Open Sockets Total. The total number of registrations for socket mixing on the sever.

Number of Optimization. The total number of global transactions converted to single phase for optimization.

Number of Passivates. 1. The number of times beans were passivated (applies to Entity and stateful session beans).

2. The number of times the system passivated (removed from memory) a bean instance.

Number of Passivations. The number of beans made passive.

Number of Pending Requests Current. The number of waiting requests in the queue.

Number of Pending Requests Oldest Time. The time that the longest waiting request was placed in the queue.

Number of Persistence Loads. The number of times bean data was loaded from persistent storage. This applies to entity beans.

Number of Persistence Stores. The number of times bean data was stored in persistent storage. This applies to entity beans.

Number of Primary. The number of objects that the local server hosts as primaries.

Number of Ready Beans or Concurrent Actives. The number of bean instances in ready state or method-ready state.

Number of Reload Total. The total number of servlets that were reloaded.

Number of Reloads. The number of servlets that were reloaded.

Number of Removes. 1. The number of times beans were removed.

2. The number of remove calls.

Number of Resend Requests. The number of state-delta messages that had to be resent because a receiving server in the cluster missed a message.

Number of Restarts. The total number of restarts for this server since the cluster was last activated.

Number of Returns. The total number of connections freed.

Number of Second Active Transactions. The total number of seconds for all committed transactions.

Number of Serviced Requests Total Time. The number of requests which have been processed by the queue.

Number of Session Pools Current. The current number of session pools instantiated on the JMS server.

Number of Session Pools High. The peak number of session pools instantiated on the JMS server since the last reset.

Number of Session Pools Total. The number of session pools instantiated on the JMS server since the last reset.

Number of Sessions Current. The current number of sessions for the connection.

Number of Sessions High. The peak number of sessions for the connection since the last reset.

Number of Sessions Opened Total. The total number of open sessions in this web application component.

Number of Sessions Total. The number of sessions on the connection since the last reset.

Number of Stores. The number of times the system wrote bean data to the database.

Number of Threads Dead. The number of threads that died.

Number of Threads Started. The number of threads started.

Number of Time-outs Total. The total number of transactions that have timed out.

Number of Total Connections. The total number of JDBC connections in the JDBCConnectionPoolRuntime MBean since the pool was instantiated.

Number of Transactions. The total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.

Number of Transactions Abandoned. The number of transaction that were abandoned.

Number of Transactions Committed Total. The total number of committed transactions.

Number of Transactions Heuristic. The number of transactions that completed with a heuristic status.

Number of Transactions Rolled Back App. The number of transactions that were rolled back due to an application error.

Number of Transactions Rolled Back Resource. The number of transactions that were rolled back due to a resource error.

Number of Transactions Rolled Back System. The number of transactions that were rolled back due to an internal system error.

Number of Transactions Rolled Back Total. The total number of transactions rolled back.

Number of Transactions Timed Out Total. The number of transactions that were rolled back due to a timeout expiration.

Number of Unlocked Users Total. The number of times a user was unlocked on a server.

Number of User Lockout Total. The cumulative number of user lockouts done on a server.

Number of Waiters Total. The number of times a thread requested and had to wait for a bean from the pool.

Number of Waits for Lock. The number of times that a thread waits for a lock.

Number Waiting for Connections. The current total number of threads waiting for a connection.

Number Waiting for Connections High. The peak number of threads waiting for a connection in the JDBCConnectionPoolRuntimeMBean. The count starts at zero each time the JDBCConnectionPoolRuntimeMBean is instantiated.

O

Object. An instance of a class.

ORB. Object Request Brokers. An Object Request Broker (ORB) manages the interaction between clients and servers, using the Internet InterORB Protocol (IIOP). It enables clients to make requests and receive responses from servers in a network-distributed environment.

ORS. Original Request String.

Override Monitoring Level. The selected monitoring level will override the current monitoring level (until the next scheduled monitoring level change or monitoring level override.)

P

Per Method Concurrent Requests. The number of concurrent calls to invoke the same method.

Percent CPU Usage. The average percentage the CPU has been busy since the last query.

Percent Maxed. 1. The average percent of the time that all connections are in use.

2. The average percent of the time that all threads are in use.

Percent of Total Number. The number of Java objects belonging to the same Java class and the total number of Java objects in the heap.

Percent Time Max in Use. 1. The average percentage of time that all connections are in use.

2. The percentage of time that the maximum configured threads are in use. If this value is consistently in the double-digits, then the Web container could be a bottleneck and the maximum number of threads available to the Web Container should be increased. See WebSphere documentation.

Percent Used. The average percentage of a pool that is in use.

Plan Name. The DB2® plan name used by a connection.

Platform. The application server product name.

Platform CPU Delta. 1. The amount of CPU time that the operating system used since the last refresh. (This feature does not apply to z/OS Platform)

2. The system highlights this number on the Server Statistics Overview page when the amount of CPU time that the operating system used since the last page refresh exceeds the threshold value.

Platform CPU% Utilization. The percent of the total CPU being utilized by the server platform.

PMI Polling Frequency. 1. Performance Monitoring Infrastructure Polling Frequency.

2. The number of times the system resources request information from the PMI in seconds.

Pool Max Capacity. The maximum capacity of the servlet for single thread model servlets.

Pool Size. 1. The average pool size.

2. The average number of threads in the pool.

Pool State. The pool state as one of "Active" or "Suspended".

Port Number. The port number of the machine being sent SNMP messages.

Portal. A single, secure point of access to diverse information, applications, and people that can be customized and personalized.

Portlet. A reusable web module that runs on a portal server. Portlets have predefined roles such as retrieving news headlines, searching a database, or displaying a calendar.

Prepared Stmt Cache Discard Count. The total number of statements discarded because the statement cache is at its maximum size.

Priority. A number assigned to the JVM thread.

Processing Time. The time (in milliseconds) it takes a registered portable interceptor to run.

Publish Server. Accepts data from a Data Collector and aggregates it based on different needs.

Q

R

Recent Activity. A diagnostic tool that provides server activity analysis reports regarding memory.

Recent Requests. A feature that describes requests that have recently completed.

Recycle Total. The total number of Connector connections that have been recycled in this Connector Pool since the pool was instantiated.

Reentrant. A bean's reentrance policy (reentrant or not reentrant within transaction).

Reference Lookup Time. The amount of time (in milliseconds) taken to look up an object reference before method dispatch can be carried out.

Registered EJBs. The number of registered EJBs on an application server.

Registered Servlets. The number of registered servlets on an application server.

Remote Active Sessions. The number of active remote communication sessions attached and active within a server instance, during the interval.

Remote Bytes Received. The number of bytes transferred to the server from all remotely attached clients.

Remote Bytes Sent. The number of bytes transferred from the server to all remotely attached clients.

Remote Existing Sessions. The number of existing remote communication sessions at the end of the interval.

Remote Host. The host name of the client initialing the request.

Remote IP. The IP address of the client initialing the request.

Request. Request by a client for a particular server resource. This resource is often a Web page or a Java application.

Request Name. 1. The name of the request submitted to the server.

2. Allows you to assign alternate request identifiers that are more meaningful and appropriate to the chosen programming model of the application.

Request Object. The J2EE server converts an HTTP request to an HTTP request object and delivers it to the Web component identified by the request URL. The Web component fills in an HTTP response object, which the server converts to an HTTP response and sends to the client.

Request Object Attributes. The attributes bound to a request object.

Request Sampling Rate. The percentage of requests that will be stored in the database for reporting and analysis.

Request Type. The type of request, such as EJB, JSP, or servlet.

Request URL. The URL associated with a request.

Requests. The total number of times the servlet or JSP was requested during the interval.

Resident Time. The time a request has been active and served.

Resource. 1. The resource selected for a trap, i.e., Occurrence, CPU time, Resident time, Wait time,

SQL Resident time, HTTP request parameters, or SQL statements.

2. The full name of an EJB.

3. The full name of a servlet or JSP on an application server.

Resource Adapter Link Ref. The Resource Adapter Link Reference for cases where a Connection Factory refers to an existing Resource Adapter deployment.

Response Time (ms). The response time (in milliseconds) of a request.

Returns Discarded. The number of times a returning object was discarded because the pool was full. This applies to entity and stateless beans.

Returns to Pool. The number of calls returning an object to the pool. This applies to entity and stateless beans.

RMI. Remote Method Invocation. A standard from Sun for distributed objects written in Java. RMI is a remote procedure call, which allows Java objects (software components) stored in the network to be run remotely.

Role. The administrator assigns a role to each user. The system default roles are Administrator, Operator, and User. The administrator can create custom roles to suit the needs of their specific environment.

Rolling Date. One of three options for specifying the baseline average response time on the Systems Overview pages. The number of days over which the average response will be calculated. The response times on the Systems Overview pages will be compared to these baselines.

Run-time Environment. The specifics regarding the set up and installation of a server. The Application Monitor provides details for three environments: System, Java, and application server.

Run-time Exception. The exception generated by an application during the normal operation of the Java Virtual Machine.

Runnable. The thread is active or executing.

S

Sample End Time. The time that the last sample arrived.

Sample Start Time. The time that the system received the first sample.

Sample Sum. The total number of samples collected for a report period.

Sampling Frequency. The percentage of requests that will be stored in the database for reporting and analysis.

Schedule. A set of definitions of when a Data Collector will switch monitoring levels.

Secondary Distribution Names. The names of the remote servers (e.g. myserver) of which the local server is hosting secondary objects. The name is appended with a number to indicate the number of secondaries hosted on behalf of that server.

Security Information. The pathname where the Security Policy file is stored.

Selector. 1. The selector associated with a consumer.

2. A selector for a durable subscriber.

Serializable Session Object Size. The average size of session objects at session level, including only serializable attributes in the cache.

Server Activity Display. Tracks transactions and requests and provides detailed thread data for an application server at a specific point in time.

Server Name. 1. The name of the server where the system captured data.

2. The combination of the admin server name and the Application Server name and the process ID, or in the case of z/OS platform, the name of the Sysplex node, the Application Server name, and the address space ID of the server region.

Server Names. The names of the servers in a cluster.

Server Overview. This page displays comprehensive server information, activity, statistics, and resource data for a selected server.

Server Region Name. The name of the Server Region which belongs to a server instance.

Server Resource Trap. A trap on system resource activity, as opposed to a user's application behavior.

Server Scope. The server(s) on which a report is generated.

Server Session Usage. The percentage of ServerSession pool in use. This applies to Message Driven beans.

Servlet. A Java application that runs in a Web server or application server and provides server-side processing, typically to access a database or perform e-commerce processing. Servlets provide a Java-based component-based, platform-independent method for building Web-based applications. It is a Java-based replacement for CGI scripts, Active Server Pages (ASPs) and proprietary plugins written in C and C++ for specific Web servers (ISAPI, NSAPI).

Servlet Name. The name of the servlet or JSP on an application server.

Servlet Volume. The number of times that servlet requests were sent to the application server.

Servlet/JSP Activity. The number of servlet/JSP calls made in the last hour, with a 5 minute refresh rate.

Servlet/JSP Coverage. The graphical representation of the most frequently accessed servlet/JSPs.

Session Attributes. The attributes bound to a session object.

Session Create Time. The time the server created a session.

Session Created. The number of HTTP sessions created.

Session ID. The ID associated with an HTTP session object.

Session Invalidate Time. The average time from when a session is invalidated until it is finished.

Session Lifetime. The average session life time.

Session Object. The session object is used to share information for one user across multiple pages while visiting a Web site. In other words, a session object is a way of retaining state for a normally stateless HTTP Web site. The J2EE container creates the session object when a client makes a request to the server. When the same client makes another request, the server finds the session object associated with that client and uses it.

Session Invalidated. The number of HTTP sessions invalidated.

Shrink Count Down Time. The amount of time left (in minutes) until an attempt to shrink the pool will be made.

Shrink Period Minutes. The Shrink Period (in minutes) of a Connector connection pool.

Shrinking Enabled. The shrinking of a Connector connection pool is enabled.

Size of Live Objects on Heap. The size of the Instance Data that is currently in storage.

Size Percent of Total Size. The total size of the Java objects on the heap size.

Snapshot Date. The date when the currently displayed data was frozen.

Snapshot Time. The time when the currently displayed data was frozen.

SNMP. Simple Network Management Protocol. A set of protocols for managing complex networks.

Source Info. An informative string about a component's source.

SQL Call. The SQL operation performed on a Table.

SQL Statement. The SQL statement that is currently being processed by the connection.

SSL Listen Address. Secured Socket Layer. The address on which the server is listening for connections.

SSL Listening Port. Secured Socket Layer. The secure port the application server uses to listen for requests.

Stack Trace. Displays a list of method calls starting with the method where the Stack Trace printed in a Last in First Out order. For each method, the class name, method name, and (optionally) a line number are displayed.

Start Date/Time. An ID assigned to a thread. The ID cannot be modified.

State. The state of a messaging bridge.

Stateful Session Bean. A session bean with a conversational state.

Stateless Session Bean. A session bean with no conversational state. All instances of a stateless session bean are identical.

Status. 1. A string representation of a transaction's status.
2. A component's status.

Subscription Name. The subscription name for a durable subscriber.

Suspended. A user paused the thread and can re-active it when ready.

System Paging Rate. A paging file is a space on a hard disk used as the virtual memory extension of a computer's real memory (RAM). Having a paging file allows a computer's operating system to pretend that it has more RAM than it actually does. The least recently used files in RAM can be swapped out to the hard disk until they are needed later so that new files can be loaded into RAM. In larger operating systems, the units that are moved are called pages and the swapping process is called paging. Paging rate is referring to the rate of the swapping process in kilobytes per second.

System Resources. Displays the summary for all system resources usage information with a 5 minute refresh rate.

System Resources Comparison. A feature that lets you compare all the servers in a group by a selected resource.

System Resources Polling Frequency. Set how often the Managing Server requests system resources information from your application servers. The default setting is 60 seconds.

T

Table Name. The name of the table affected by a SQL call.

Target Type. The metric used in a trap, e.g., DB Pool Size or CPU Time.

Thread. A thread allows multiple streams of execution concurrently and independently in the same program.

Thread Create. The total number of threads created.

Thread Destroy. The total number of threads destroyed.

Thread Dump. Detailed information of memory allocation of threads in a JVM.

Thread ID. ID assigned to a thread by the JVM when the thread is created. The ID cannot be modified.

Thread Pool. A pool of threads available for servicing client requests. The J2EE application server pre-creates a collection of threads. This collection is the pool. As new requests arrive to the server, it assigns a free thread to a request. When the request completes, the thread is returned to the pool.

Thread Stack. A list of methods currently being executed in a thread. In a thread, method A invokes method B that invokes method C, the stack is A->B->C. When C finishes, it becomes A->B.

Thread Status. Suspend status denotes that an operator suspended a thread, while Active status denotes an executing thread. To return a thread to active status, select Resume.

Thread Type. The types of thread, such as JSP, EJB or Servlet.

Thread's Priority. The priority of an active thread.

Threshold. A value against which server activity is compared. The system will send alerts when the actual value exceeds the threshold.

Throughput (response / min, Last Hour). The amount of transactions / requests being transmitted in a given period of time, with the response time per minute of a server to process a transaction in last hour.

Time Since Last Activated. The time difference, in milliseconds, of the previous and current access time stamps. This does not include sessions timed out.

Top CPU Intensive Methods Report. A report that displays the most popular unique methods that, during the report period, took the most cumulative CPU time and the sum total CPU time. Displays up to 100 records.

Top CPU Intensive Requests Report. A report that displays unique requests that, during the report period, took the most cumulative CPU time and the sum total CPU time. Displays up to 100 records.

Top Method Used Report. A report that displays the most popular unique methods used during the report period, and how often each request was used. Displays up to 100 records.

Top Request Used Report. A report that displays the most popular unique requests used during the report period, and how often each request was used. Displays up to 100 records.

Top Slowest Methods Report. A report that displays the top unique methods that took the longest time to

complete and the average completion time. It displays up to 100 records.

Top Slowest Requests Report. A report that displays unique requests that took the longest time to complete, as well as the average completion time. Displays up to 100 records.

Top SQL Intensive Methods Report. A report that displays unique methods that made the highest sum total of SQL calls during the report period. Displays up to 100 records.

Top SQL Intensive Requests Report. A report that displays unique requests that made the highest sum total of SQL calls during the report period. Displays up to 100 records.

Top SQL Used Report. A report that displays the five SQL call types that were most often called, as well as the number of calls during the report period.

Top Tables Used Report. A report that displays the tables that were called most often, as well as the number of times each table was called during the report period. Displays up to 100 records.

Total CPU%. 1. The percentage of time that the entire platform was using CPU.
2. The system highlights this number on the Server Statistics Overview page when the total CPU usage exceeds the threshold value.

Total GC Time. The total time of a routine that searches memory to reclaim space from program segments or inactive data.

Total Memory. The total memory in JVM runtime.

Total Method Calls. 1. The number of calls to a bean's remote methods.
2. The total number of methods being processed. A measure of server activity.

Total Method Count. The total number of methods being processed by the selected request.

Total Requests. 1. The total number of requests sent to ORB.
2. The total number of requests a servlet processed.
3. The total number of times the Servlet or JSP services made a request during the interval.

Total Resident Time. The total amount of time, in milliseconds, since the start of the request.

Total Sessions. The total number of HTTP sessions tracked by the server at the interval. Includes both active and inactive sessions.

Total SQL Used. The SQL call types that were most often called, as well as the number of calls.

Total Thread Count. The total number of active requests being serviced by the selected application server.

Total Volume. 1. The number of completed requests.

2. The system highlights this number on the Server Statistics Overview page when the total number of completed requests exceeds the threshold value.

Transaction Failure Rate. The percentage of transactions handled by the application server that did not successfully complete.

Transaction Policy. The bean method's transaction policy: TX_NOT_SUPPORTED; TX_BEAN_MANAGED; TX_REQUIRED; TX_SUPPORTS; TX_REQUIRES_NEW; TX_MANDATORY; TX_NEVER.

Transaction Server Name. 1. The name of the Sysplex.

2. The name of the WebSphere Server.

Transaction Supported. The transaction support level for the Resource Adapter for a Connector connection pool.

Transaction Volume. The number of times that transactions were executed on an application server.

Trap. A set of conditions, thresholds, and criterion set by the user, which, when met, trigger actions.

Trap Action History. Whenever a trap triggers an action, a record will be placed in this page.

Trap Condition. The user-defined criteria that is part of a trap definition.

Trap Type. A way of categorizing two different types of metrics used to define traps application traps or server resource traps.

Trend Report. A report that displays the results of the defined data set. To view the detailed report broken down by different criteria, choose an option from the Additional Detail drop-down menu, and then click on the data points.

U

UUID. Universally Unique Identifier. An identifier for each symbol in an activity diagram.

Unique Request. All the instances of a specific request string.

Uptime. The amount of time, in seconds, the JVM has been running.

Used Memory. The used memory in the JVM runtime.

User Name. The database user name that is used for creating a connection.

V

Volume Delta. 1. The number of completed requests since the screen refreshed.

2. The system highlights this number on the Server Statistics Overview page when the number of completed requests since the last refresh exceeds the threshold value.

Volume Throughput. The amount of data being processed in a specified amount of time.

W

Wait Seconds High Count. The number of seconds the longest waiter for a connection waited.

Waiting Condition. The thread is waiting on a condition variable.

Waiting Monitor. The thread is waiting on a monitor.

Web Container. Handles requests for servlets, JSP files, and other types of server-side include coding. The Web container creates servlet instances, loads and unloads servlets, creates and manages requests and response

objects, and performs other tasks for managing servlets effectively.

Web Server Overview. Displays information about the performance of Web servers.

WLM. Workload Manager.

WLM Associated Service Class. This page offers a way to view selected data from the Workload Manager for z/OS and OS/390, for the address space associated with a particular server, as well as its associated service class data and service class period data.

WLM Associated Service Class Period. This page displays the response time distribution detail and delay detail information about each subsystem work manager.

Worst Performers. The 5 worst performing EJBs and Servlets. Worst Performing is defined as the slowest response time.

X

Y

Z

Notices

This information was developed for products and services offered in the U.S.A. IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION •AS IS• WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including

this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

P.O. Box 12195, Dept. TL3B/B503/B313

3039 Cornwallis Rd.

Research Triangle Park, NC 27709-2195

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE: This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to Cyanea Systems Corp., for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM or Cyanea, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM or Cyanea Systems Corp. for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's and Cyanea's application programming interfaces.

Koshna SNMP Agent Toolkit

“Portions of this program include material copyrighted © by Koshna Software Technologies Pvt Ltd 2000-20002”

Apache Software Foundation

Log4j, XML Parser, II8N Tag Libraries, Jakarta REGEXP, and BCEL are subject to the Apache Software Version 1.1, see below.

=====

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the

Apache Software Foundation (<http://www.apache.org/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED

WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

iText

“Report format conversion engine Copyright © 1999 - 2003 Bruno Lowagie and others. The source code version of the report format conversion engine is available under the terms of the Mozilla Public License, v. 1.1, from www.lowagie.com/iText/download.html . Any terms which differ from the terms of Mozilla Public License, v. 1.1, are offered by Cyanea Systems Corp. alone, and not by Bruno Lowagie or any other contributor to the report format conversion engine source code.”

Trademarks

The following terms are trademarks of the IBM Corporation or its subsidiaries in the United States or other countries or both:

Table 26. Trademarks

IBM	MVS
WebSphere	WebSphere Studio
z/OS	DB2
S/390	OS/390
AIX	AIX-5L
Tivoli Enterprise Console	Tivoli
USS	TSO
RACF	TME

The following terms are trademarks of other companies:

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and JavaScript are registered trademarks and Sun Solaris and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Pentium is a registered trademark of Intel Corporation in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

HP-UX is a trademark of Hewlett-Packard Corporation in the United States, other countries, or both.

RedHat is a trademark of RedHat Corporation in the United States, other countries, or both.

Linux is a trademark of Linux Torvalds in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

Index

A

- Activate
 - Thread 60
 - Trap 99
- activating a thread 60
- Active Request Data
 - Filter 56
 - Sort 57
- Active Requests 56
- Active Status
 - Thread 60
- Active Thread
 - View 82
- Application Name
 - Request Mapper 156
- Application Trap
 - Create 96
 - Define 96
 - Set 96
- Apply
 - Data Collector Configuration 19
- Assign
 - Role 11
- Authorization
 - Composite Transactions 148

C

- Cancel
 - Request 60
 - Thread 83
- canceling a request 60
- Change
 - Thread Status 59
 - Thread's Priority 61
- CICS Region 142
- CICS Transaction 106
- CICS Transactions
 - Composite Transactions 142
- Composite Method Trace
 - View 146
- Composite Request Space
 - Define 140
- Composite Stack Trace
 - View 147
- Composite Transaction Indicator
 - Use 144
- Composite Transactions
 - Authorization 146
 - In-Flight Request Search 144
 - Performance Analysis & Reporting 145
- Configure
 - Data Collector 17
 - Server Statistics Overview 50
- Create
 - Application Trap 96
 - Data Collector Configuration 18
 - Memory Analysis Report 74, 81
 - Memory Leak Confirmation Report 77, 83
 - Recent Activity Report 71
 - Role 10
 - Schedule 33
 - User Account 8
- CSV
 - exporting reports 136

D

- Daily Statistics Overview
 - z/OS Platform 118
- Data Collector
 - Configure 17
 - Disable 18
 - Enable 18
 - Unconfigure 17
- Data Collector Configuration
 - Apply 19
 - Create 18
 - Delete 21
 - Modify 20
- Data Collectors 16
- Database Connection Pools 109, 113
- Date Range Settings 137
- Deactivate
 - Trap 100
- Decomposition Report 124, 125, 128, 129
- Default Request Mapping Behavior
 - Request Mapper 157
- Define
 - Application Trap 96
 - Composite Request Space 140
 - Server Availability Analysis Report 129
 - Server Resource Trap 97
- Delete
 - Data Collector Configuration 21
 - Role 11
 - Trap 101
- Delete Old Daily Statistics Snapshots
 - z/OS Platform 118
- Delete Schedule
 - MOD 35
- Disable
 - Data Collector 18
- Distributed Program Link 142
- DPL 142
- Duplicate
 - Role 10
 - Trap 101
- Duplicate Schedule
 - MOD 35

E

- EAI 140
- EJB 109, 113
- Email
 - PDF 66, 135
 - Report 135
- Enable
 - Data Collector 18
- Enterprise Application Integration 140
- Enterprise Overview 37
- Export
 - File 67
 - reports 136

F

- File
 - Export 67
 - File Name Match 88
 - File Name/Path/Size Match 88
- Filter

- Active Request Data 57
- Recent Request Data 58
- Thread Data 57, 71
- Full Match 88

G

- Group Overview 38

H

- Heap Analysis 75, 82

I

- In-Flight Request Search
 - Composite Transactions 144
 - using method trace 63
- Installed Binary Check
 - Run 89
- Installed Binary Detail
 - View 89
- Installed Binary Files 89
- Interface MappedRequest
 - Request Mapper 158

J

- JCA Connection Pools 110
- JTA Transactions 110
- JVM Thread Display 81
- JVM/System 111

M

- Manage
 - MOD 30
 - Traps and Alerts 94
- Managed Space 140
- Memory Analysis 73
- Memory Analysis Report
 - Create 74, 81
- Memory Leak 77, 83
- Memory Leak Confirmation Report
 - Create 77, 83
- Method Analysis Report
 - Define 125
 - defining 125
- Method Report 125
- method trace
 - about 63
 - drilldown view, detail 69
 - drilldown view, report 70
 - emailing a PDF 66
 - exporting to file 67
 - flow view 64
 - nesting summary 69
 - searching 67
 - thread status 63
 - threshold highlighting 64
 - viewing PDF 66
- Metrics
 - CICS Transaction 106
 - Database Connection Pools 109, 113
 - EJB 109, 113
 - JCA Connection Pool 110
 - JTA Transaction 110
 - JVM/System 111
 - ORB Detail/Interceptor 111
 - Queue 107
 - Queue Manager 106
 - Server 114

- Server Regions 115
- Servlet and Session Manager 114
- Session Manager 112
- SQL 107
- Thread Pool 112
- Web Applications 111, 115
- MOD
 - Delete Schedule 35
 - Duplicate Schedule 35
 - Manage 30
 - Modify Schedule 34
- Modify
 - Data Collector Configuration 20
 - Role 10
 - Trap 101
- Modify Schedule
 - MOD 34
- Monitoring Level
 - Override 34
 - View 31
- Multiple Hops 143

N

- Name Trap 99

O

- ORB Detail/Interceptor 111
- Override
 - Monitoring Level 34

P

- PAR File
 - Export 136
- PAR PDF File
 - View 136
- PDF
 - Email 136
 - method trace 66
 - viewing method trace 66
 - viewing reports 136
- PDF File
 - View 66
- Performance Analysis & Reporting
 - Composite Transactions 145
 - method trace features 68
- PMI Data 108
- Portal Details 43
- Portal Overview 3, 42

Q

- Queue 107
- Queue Manager 106

R

- Recent Activity Report
 - Create 71
- Recent Request Data
 - Filter 58
- Recent Requests 57
- Remove Server
 - Server Statistics Overview 49
- Report
 - Email 135
 - Modify 134
- Reports
 - date range settings 137
 - deleting 135

- emailing 135
- exporting CSV file 136
- modifying 134
- running 134
- viewing 131
- viewing PDF file 136
- Request
 - Cancel 60
 - Search 53
- request
 - canceling 60
 - viewing object 61
- Request Analysis Report
 - Define 124
 - defining 124
- Request Detail 58
 - View 58, 146
- Request Mapper
 - Interface MappedRequest 158
- Request Object Session Object
 - View 61
- Request Report Detail 124
- Role
 - Assign 11
 - Create 10
 - Delete 11
 - Duplicate 10
 - Modify 10
- Role Configuration 9
- Run
 - Installed Binary Check 89

S

- Sample Request Mapper
 - Request Mapper 157
- Schedule
 - Apply 33
 - Create 33
 - Delete 35
 - Duplicate 35
 - Modify 34
- Search
 - Request 53
- Self Diagnosis 26
 - Archive Agent 27
 - Data Collector Controller 27
- Self-Diagnosis
 - Email Engine 28
- Self-Diagnosis
 - Kernel 26
 - Publish Server 27
- Server 114
- server activity 55
- Server Activity Display
 - about 55
- Server Availability
 - Detail 48
- Server Availability Analysis Report
 - defining 129
- Server Overview 39
- Server Regions 115
- Server Resource Trap
 - Define 97
 - Set 97
- Server Resource Trap Alert
 - Set 97
- Server Statistics Configuration 49
- Server Statistics Overview
 - Configure 50
 - Display Groups 49
 - Display Servers 49
 - Remove Server 49
- Servlet and Session Manager 114

- session
 - viewing object 61
- Session Manager 112
- Set
 - Application Trap 96
 - Server Resource Trap 97
 - Server Resource Trap Alert 97
 - Severity Level 98
 - Trap Alert 98
- Severity Level
 - Set 98
- SMF Data 108
- Sort
 - Active Request Data 57
 - Thread Data 57
- SQL 107
- SQL Analysis Report
 - defining 125
- SQL Report 125
- Stack Trace
 - View 62
- stack trace
 - viewing 62
- suspending a thread 59
- System Resource Analysis Report
 - Define 128, 129
 - defining 128, 129
- System Resources Overview 104
 - z/OS Platform 105

T

- Thread
 - Activate 60
 - Cancel 83
 - Suspend 59
- thread
 - activating 60
 - changing priority 61
 - status in method trace 63
 - suspending 59
- Thread Data
 - filter 57, 71
 - Sort 57
- Thread Dump 84
- Thread Pools 112
- Thread Status
 - Change 59
- Thread's Priority
 - Change 61
- Top Report
 - Define 127
 - defining 127
 - modifying 135
- Trace Report 124
- Trap
 - Activate 99
 - Deactivate 100
 - Delete 101
 - Duplicate 101
 - Modify 101
- Trap Action History
 - View 102
- Trap Alert
 - Set 98
- Traps
 - viewing method trace 63, 68
- Trend Report 124, 125, 128, 129

U

- Unconfigure
 - Data Collector 17
- Use

- Composite Transaction Indicator 144
- User Account
 - Create 8
- User Audit Trail 153

V

- View
 - Active Thread 82
 - Composite Method Trace 146
 - Composite Stack Trace 147
 - Installed Binary Detail 89
 - Monitoring Level 31
 - PDF File 66
 - Request Detail 58, 146
 - Request Object Session Object 61
 - Stack Trace 62
 - Trap Action History 102

W

- Web Applications 111, 115
- Web Server Details 41
- Web Server Overview 40
- WebSphere - PMI 108
- WLM Associated Service Class Period Detail 45, 46
- WLM Associated Service Class Summary 44

Z

- z/OS Platform
 - Daily Statistics Overview 118
 - Delete Old Daily Statistics Snapshots 118
 - System Resources Overview 105

Bibliography

- *WebSphere Studio Application Monitor
User's Guide (SC31-6865-00)*
- *WebSphere Studio Application Monitor
Operator's Guide (SC31-6866-00)*
- *WebSphere Studio Application Monitor
Installation and Configuration Guide (SC31-6867-00)*
- *WebSphere Studio Application Monitor
CICS Data Collector Product Guide (SC31-6868-00)*
- *WebSphere Studio Application Monitor
IMS Data Collector Product Guide (SC31-6869-00)*
- *WebSphere Studio Application Monitor
Messages and Codes (SC31-6864-00)*
- *WebSphere Studio Application Monitor
Program Directory for
IBM WebSphere Studio Application Monitor (GI10-3349-00)*
- *WebSphere Studio Application Monitor
Program Directory for the
CICS Data Collector (GI10-3350-00)*
- *WebSphere Studio Application Monitor
Program Directory for the
IMS Data Collector (GI10-3351-00)*

Readers' Comments—We'd Like to Hear from You

WebSphere Studio Application Monitor
WebSphere Studio Application Monitor User's Guide
3.1

Publication No. SC31-6865-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your response. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

_____ Name	_____ Address
_____ Company or Organization	_____
_____ Phone No.	_____

SC31-6865-00



Cut or Fold
Along Line

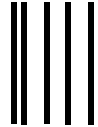
IBM Corporation
P.O. Box 12195, Dept. TL3B/B503/B313
3039 Cornwallis Rd.
Research Triangle Park, NC
U.S.A.27709-2195



Fold and Tape

Please do not staple

Fold and Tape



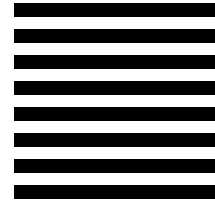
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

SC31-6865-00



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your response. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

_____ Name	_____ Address
_____ Company or Organization	_____
_____ Phone No.	_____

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your response. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

_____ Name	_____ Address
_____ Company or Organization	_____
_____ Phone No.	_____



Part Number: 5697-J18

SC31-6865-00



(1P) P/N: 5697-J18

