IBM Tivoli Composite Application Manager for Application Diagnostics

Version 7.1.0.4

User Guide



IBM Tivoli Composite Application Manager for Application Diagnostics

Version 7.1.0.4

User Guide



Note

Before using this information and the product it supports, read the information in “Notices” on page 797.

Contents

About this guide ix
Who should read this guide ix
Publications ix
ITCAM for Applications library for Agents for WebSphere Applications, J2EE, and HTTP Servers. ix
Related publications x
Accessing terminology online xi
Accessing publications online xi
Ordering publications xi
Accessibility xi
Tivoli technical training xii
Supporting information xii
Conventions used in this guide. xii
Typeface conventions xii
Operating-system-dependent variables and paths xiii
Tivoli command syntax xiii

Part 1. Part 1: Introduction to the agents 1

Chapter 1. Overview of the agents . . . 3
Agent functionality 3
User interfaces. 4
Components used by the Tivoli Enterprise Portal user interface 5
The agents 5
IBM Tivoli Monitoring components 7
Components used by the MSVE user interface . . . 8
Tivoli Enterprise Portal interoperation with MSVE 10
What is new 11
Getting Started with ITCAM for Application Diagnostics 12

Chapter 2. Scenarios. 13
Scenario 1: Diagnosing a memory leak 13
Scenario 2: Diagnosing hanging transactions . . . 15
Scenario 3: Diagnosing a WebSphere server shutdown 19
Scenario 4: Determining if the WebSphere cluster needs to be load balanced 21
Scenario 5: Determining the cause of high response times 23
Scenario 6: Determining the cause of connection problems 29
Scenario 7: Determining if the Garbage Collection policy needs to be adjusted 30
Scenario 8: Troubleshooting application response time in an XD cell 32
Scenario 9: Ensuring that jobs processed by Compute Grid do not execute for longer than one hour. 35

Part 2. Part 2: Using the agents. . . 39

Chapter 3. ITCAM for Application Diagnostics - Managing Server Visualization Engine 41
Using IBM Support Assistant to diagnose problems 41
Access the Managing Server Visualization Engine from Tivoli Enterprise Portal 42
Account management 46
User Scenarios 46
User profiles 47
Role configuration 49
Server management 52
User Scenarios 52
Server groups 53
Data Collector Configuration 56
Data Collector Profiles. 65
Monitoring on Demand (TM) 77
User Scenarios 78
MOD Console 78
Schedule Management. 79
Creating a schedule. 79
Applying a schedule 80
Overriding a monitoring level 81
Modifying a schedule 81
Deleting a schedule. 82
Duplicating a schedule 82
Managing server. 83
System properties 83
Self-diagnosis. 86
Systems overview 89
User Scenarios 89
Enterprise Overview 90
Group Overview 90
Server Overview. 91
Portal Overview 92
Viewing the Portal Page Summary. 93
Viewing the Portlet Summary 94
WLM associated service class summary 95
WLM associated service class period detail . . . 95
Viewing a WLM enclave 96
Server statistics overview. 97
User scenarios 97
Configuring the Server Statistics Overview page 98
Viewing the Server Statistics Overview 99
Recent Activity Display 100
User Scenarios 100
Creating a Recent Activity report. 100
System resources 101
User Scenarios 101
Viewing the System Resources Browser. . . . 102
Resources Performance Metrics 103
SMF data. 107
User Scenarios 107
Viewing SMF data. 107
Alerts and Events 108

© Copyright IBM Corp. 2006, 2015

Escalating alerts and events to the Problem Center 109
Problem Center 110
Viewing the details of a problem 111
Adding a problem manually 113
Closing a problem 114
In-flight request search 114
User Scenarios 115
Searching for an Application Request 115
Sorting search results 116
Server Activity Display 116
User Scenarios 117
Server Activity Display - active requests . . . 117
Server Activity Display - recent requests . . . 119
Server Activity Display - lock contentions . . . 121
Viewing request detail 122
Suspending a thread 123
Activating a thread 123
Canceling a request 124
Changing the priority of a thread. 124
Viewing a Stack Trace 125
Viewing a Method/Component Trace - flow view 125
Viewing a request object and session object . . 126
Searching a Method/Component Trace 126
Sending a PDF file by email: Server Activity Display (SAD) 127
Viewing a PDF file: Server Activity Display (SAD) 128
Exporting to a file: Server Activity Display (SAD) 128
Web Session Browser 129
Viewing the Web Session Browser 129
Memory diagnosis. 130
User Scenarios 130
Heap Dump Management 130
Memory Analysis 134
Heap Analysis 135
Memory Leak 135
JVM thread display 139
User Scenarios 140
Changing the priority of a JVM thread 140
Viewing a stack trace 140
Canceling a thread 141
Viewing a thread dump 142
Trap and alert management 142
User Scenario 143
Setting an Application trap 144
Setting an Application trap using the Resident Time - Misbehaving Transaction target type . . 146
Setting a Server Resource trap 148
Activating a trap 150
Deactivating a trap 151
Modifying a trap 152
Duplicating a trap 153
Deleting a trap 153
Viewing the trap action history 154
Setting alert actions and data actions 154
Software consistency check 156
User Scenarios 156
Installed Binary Comparison 156
Installed Binary Check 158
Runtime Environment Comparison 159
Runtime Environment Check 159
Performance analysis and reporting 160
User Scenarios 160
Defining reports 160
Report management 175
Method Profiling 180
Daily Statistics 182
Viewing the Daily Statistics Overview 182
Deleting Daily Statistics 183
Custom requests 183
Types of requests 183
Creating custom requests 184
Viewing custom requests 186
Composite requests 187
User Scenarios 187
The scope of composite requests 188
Composite requests involving CICS and IMS systems 190
Locate, view, and analyze composite requests 192
Viewing composite requests 193
Audit trails 200
User Scenarios 200
Accessing the user audit trail 200
Request Mapper 200
Purpose 200
Data used by the Request Mapper 201
Default request mapping behavior 202
Configuring a Request Mapper 203
Glossary 207

Chapter 4. ITCAM Agent for WebSphere Applications 247

Workspaces for ITCAM Agent for WebSphere Applications. 249
Organization of the predefined workspaces . . 249
Summary workspaces 256
Summary Workspace Views 258
WebSphere agent summary workspace 259
WebSphere agent summary status workspace 259
Application Server Summary and Business Process Manager Summary workspaces . . . 260
Resources and Applications workspaces . . . 263
Situation Mapping and Summary Workspaces 263
Summary Workspaces error messages 266
Configuration workspaces 267
Workspace link to managing server visualization engine. 271
Workspace link to ITCAM for SOA 280
Alarm Manager workspace. 281
Allocation Failures workspace. 282
Selected Application - Application Trend at L1 workspace 282
Selected Application - Application Trend at L2/L3 workspace 283
Application Health workspace. 283
Application Registry workspace 285
Business Process Manager workspace 285
Cache Analysis workspace 286
Client Communications workspace 287

Container Object Pools workspace 288
Container Transactions workspace 289
Data sources workspace 290
DB Connection Pools workspace 292
DCS Stacks workspace 294
Destinations workspace 295
Durable Subscriptions workspace. 296
EJB Containers workspace 297
Enterprise Java Beans workspace 299
Garbage Collections - Selected Allocation Failure workspace 300
Garbage Collection Analysis workspace . . . 300
High Availability Manager workspace 302
IMAP/POP workspace 303
J2C Connection Pools workspace 304
JMS Summary workspace 306
JVM Stack Trend workspace 307
Log Analysis workspace. 308
Lotus Workplace Server workspace 309
Messages Queues workspace 309
Messaging Engine Communications workspace 310
Messaging workspace 311
OS Stack workspace 312
Pool Analysis workspace 313
Portal Pages Summary workspace 314
Portal Summary workspace 315
Portlet Summary and Selected Portlet - History workspaces 316
Request Analysis workspace 317
Request Baseline workspace 319
EJB Tier Analysis workspace 320
Application Configuration workspace 321
Backend Tier Analysis workspace 322
Application Health History workspace 323
Web Tier Analysis workspace 323
Selected Datasources - Datasource Trend workspace 324
Selected Request - Datasources workspace. . . 325
Selected Request - JMS Queues workspace . . 325
Selected Request - Portal Processing workspace 326
Selected Request - Resource Adapters workspace 326
Selected Request - Service Components workspace 327
Service Component Elements workspace . . . 327
Selected Request - History workspace 328
Service Components workspace 329
Servlets/JSPs - Selected Web Application workspace 330
Scheduler workspace 331
Sessions workspace 332
Thread Pools workspace. 333
Thread Pool Trend workspace 334
Web Applications workspace 335
Web Services workspace. 337
WebSphere Agent workspace 338
WebSphere application server workspace . . . 339
WebSphere ESB Server workspace 340
WebSphere Portal Server workspace. 341
WMQ Client Link Communications workspace 342
WMQ Link Communications workspace . . . 343
Workload Management workspace 344
Workplace Mail workspace 345
Region workspaces in a z/OS environment . . 346
Attributes for ITCAM Agent for WebSphere Applications. 349
Attribute groups used by the predefined workspaces 349
Alarm Manager attributes 351
Allocation Failure attributes 353
Application Health Status attributes 355
Application Monitoring Configuration attributes 357
Application server status attributes 360
Application server attributes 362
Baseline attributes 365
Client Communications attributes 367
Container Object Pools attributes 374
Container Transactions attributes 376
Data sources attributes 380
DB Connection Pools attributes 383
DC Messages attributes 386
DCS Stack attributes 388
Durable Subscriptions attributes 391
Dynamic Cache attributes 393
Dynamic Cache Templates attributes 395
EJB Containers attributes 399
Enterprise Java Beans attributes 402
Garbage Collection Analysis attributes 407
Garbage Collection Cycle attributes 409
High Availability Manager attributes 412
J2C Connection Pools attributes 414
JMS Summary attributes. 418
Log Analysis attributes 420
Listener Port attributes 422
Messaging Engine Communications attributes 423
Messaging Engines attributes 425
Portal Page Summary attributes 426
Portal Summary attributes 428
Portlet Summary attributes 430
Queue attributes 431
Remote Configuration Requests attributes . . . 434
Request Analysis attributes 435
Requests Monitoring Configuration attributes 441
Request Times and Rates attributes 444
Selected Request attributes 445
Servlet Sessions attributes 448
Servlets JSPs attributes 452
Scheduler attributes 454
Service Component Elements attributes. . . . 456
Service Components attributes. 458
Thread Pools attributes 460
Topic Spaces attributes 462
Web Applications attributes 465
Web Services attributes 467
Web Services Gate Way attributes 470
WebSphere Agent Events attributes 471
WMQ Client Link Communications attributes 472
WMQ Link Communications attributes. . . . 474
Workload Management Client attributes . . . 477
Workload Management Server attributes . . . 479
Workplace Mail IMAP/POP attributes 481
Workplace Mail Queues attributes 483

Workplace Mail Service attributes 485
Agent Configuration attributes 487
KYN Command attributes 488
Server Configuration attributes 488
Situations for ITCAM Agent for WebSphere Applications 488
Predefined situations-descriptions and formulas (that run automatically) 489
Predefined situations descriptions and formulas (that run manually) 493
Take Action commands for ITCAM Agent for WebSphere Applications 500
Add_XD_Cell: Add an XD Cell to a WebSphere agent 500
Enable_Auto_Threshold: set threshold parameters 500
Override_Auto_Threshold: override threshold values 503
Remove_WebSphere_SubNode: Remove an inactive WebSphere application server 503
Set_Application_Monitoring: Set monitoring . . 504
Set_Completion_Thresholds: Set completion thresholds 504
Set_Request_Sampling_Rate: Set the sampling rate for request data 505
Start_Baselining: Start the baselining process 505
Start_GC_Monitoring: Begin reporting garbage-collection data 506
Start_Request_Monitoring : Begin reporting request data 506
Start_Resource_Monitoring: Begin reporting PMI data 507
Start_WebSphere_Server: Start a WebSphere application server 507
Stop_Baselining: stop the baselining process . . 508
Stop_GC_Monitoring: Stop reporting garbage-collection data 508
Stop_Request_Monitoring: Stop reporting request data 508
Stop_Resource_Monitoring: Stop reporting PMI data 508
Stop_WebSphere_Server: Stop a WebSphere application server 509
Update_Baseline: trigger a baseline update . . 509
Threshold calculation detail 510
WebSphere XD Overview for ITCAM Agent for WebSphere Applications 511
ITCAM for Application Diagnostics - WebSphere XD Cell workspaces 512
WebSphere XD Attributes 526
ITCAM for Application Diagnostics - WebSphere XD Take Actions 538
ITCAM for Application Diagnostics - XD Agent situations 539
WebSphere Extreme Scale overview for ITCAM Agent for WebSphere Applications 544
WebSphere Extreme Scale prerequisites 544
WebSphere Extreme Scale Workspaces 544
WebSphere Extreme Scale Attributes 565
Situations for WebSphere Extreme Scale . . . 580
WebSphere Extreme Scale Take Action Commands 581
WebSphere PMI Attribute Mapping 583
Data sources for workspaces 601
Glossary for ITCAM Agent for WebSphere Applications 614

Chapter 5. ITCAM Agent for J2EE . . . 623

Workspaces for ITCAM Agent for J2EE 625
Organization of the predefined workspaces . . 625
Configuration workspaces 630
Allocation Failures workspace 634
Application Health Summary workspace . . . 635
Application Registry workspace 636
BEA WebLogic Application Server workspace 637
Data sources workspace 637
DB Connection Pools workspace 638
DC Message Events workspace 639
EJB Components workspace 640
Enterprise Java Beans workspace 641
EJB Modules workspace 642
EJBs - Selected Enterprise Application workspace 643
Garbage Collection Analysis workspace . . . 643
Garbage Collections - Selected Allocation Failure workspace 644
J2EE Agent workspace 645
J2SE Application workspace 645
JBoss App Server workspace 646
JCA Connection Pools workspace. 647
JDBC Connection Pools workspace 648
JMS Sessions workspace 649
JMS Summary workspace 649
JTA Resources workspace 650
JTA Summary workspace 651
JVM Statistics workspace 651
Log Analysis workspace 652
Oracle App Server workspace 653
Request Analysis workspace 654
Selected Request - Baseline workspace 655
Selected Application - Application Tier Analysis workspace 655
Selected Application - Configuration workspace 656
Selected Application - Backend Tier Analysis workspace 657
Selected Application - Health History workspace 658
Selected Application - Client Tier Analysis workspace 658
Selected Request - data sources workspace . . 659
Selected Request - JMS Queues workspace . . 660
Selected Request - Resource Adapters workspace 660
SAP NetWeaver Server workspace 661
Servlets/JSPs - Selected Enterprise Application workspace 661
Servlets/JSPs - Selected Web Application workspace 662
Tomcat Server workspace 663
Web Applications workspace 663
Web Container workspace 664
WebSphere App Server CE workspace 665

Attributes for ITCAM Agent for J2EE 666
Attribute groups used by the predefined workspaces 666
Allocation Failure - J2EE attributes 668
Application Health Status attributes 669
Application Monitoring Configuration attributes 672
Application Server Status - J2EE attributes. . . 674
Application Server - J2EE attributes 675
Baseline attributes 677
Data sources - J2EE attributes 679
DB Connection Pools - NetWeaver attributes 682
DC Messages - J2EE attributes. 683
Enterprise Java Bean Components - WebLogic attributes 684
Enterprise Java Bean Modules - J2EE attributes 685
Enterprise Java Bean Service - NetWeaver attributes 687
Enterprise Java Beans - WebLogic attributes . . 689
Garbage Collection Analysis - J2EE attributes 692
Garbage Collection Cycle - J2EE attributes. . . 694
J2EE Agent Events attributes 696
J2EE Connector Connection Pools - WebLogic attributes 697
JCA Connection Pools - J2EE attributes 699
JDBC Connection Pools - WebLogic attributes 701
JDK - Operation System attributes 703
JDK - Memory attributes 705
JDK - JVM attributes 706
JDK - Threading attributes 708
JMS Sessions - WebLogic attributes 709
JMS Summary - J2EE attributes 711
Java Transaction Service - WebLogic attributes 713
JTA Resources - J2EE attributes 716
JTA Summary - NetWeaver attributes 717
Log Analysis - J2EE attributes 718
Request Analysis - J2EE attributes 720
Requests Monitoring Configuration attributes 725
Request Times and Rates - J2EE attributes . . . 727
Selected Request - J2EE attributes 729
Servlets JSPs - J2EE attributes 731
Servlets and JSPs - WebLogic attributes. . . . 733
Web Container - NetWeaver attributes 734
Web Applications - J2EE attributes 735
Web Applications - WebLogic attributes . . . 737
Situations for ITCAM Agent for J2EE 738
Take Action commands for ITCAM Agent for J2EE 742
Enable_Auto_Threshold: set threshold parameters 742
Override_Auto_Threshold: override threshold values 744
Recycle_Application_Server: Recycle a J2EE application 745
Remove_J2EE_Application: Remove a J2EE Application 745
Remove_J2EE_SubNode: Remove an inactive J2EE application server 746
Set_Application_Monitoring: Set monitoring . . 746
Set_Completion_Thresholds: Set completion thresholds 747
Set_Request_Sampling_Rate: Set the sampling rate for request data 747
Start_Application_Server: Start a J2EE application server 747
Start_Baselining: Start the baselining process 747
Start_GC_Monitoring: Begin reporting garbage-collection data 749
Start_Request_Monitoring : Begin reporting request data 749
Start_Resource_Monitoring: Begin reporting PMI data 749
Stop_Application_Server: Stop a J2EE application server 749
Stop_Baselining: stop the baselining process . . 750
Stop_GC_Monitoring: Stop reporting garbage-collection data 750
Stop_Request_Monitoring: Stop reporting request data 750
Stop_Resource_Monitoring: Stop reporting PMI data 750
Update_Baseline: trigger a baseline update . . 750
Threshold calculation detail 751
Glossary for ITCAM Agent for J2EE 753

Chapter 6. ITCAM Agent for HTTP Servers 761

Workspaces for ITCAM Agent for HTTP Servers 761
Organization of the predefined workspaces . . 761
Apache web server workspace 762
Apache websites workspace 763
Sun Java System web server workspace . . . 764
Sun websites workspace. 764
web server Agent workspace 765
Attributes for ITCAM Agent for HTTP Servers . . 766
Attribute groups used by the predefined workspaces 766
Apache web server attributes 767
Apache websites attributes 768
Sun web server attributes 770
Sun websites attributes 773
HTTP Servers Agent Events attributes 776
Web Servers Status attributes 777
Situations for ITCAM Agent for HTTP Servers . . 778
Take Action commands for ITCAM Agent for HTTP Servers 783
Glossary for ITCAM Agent for HTTP Servers. . . 784

Part 3. Appendixes 787

Appendix. Accessibility 789

Index 791

Trademarks 795

Notices 797

About this guide

This book provides a user guide for ITCAM for Application Diagnostics 7.2.

Who should read this guide

This user guide is intended for users of ITCAM for Application Diagnostics.

Publications

This section lists publications in the product library and related documents. It also describes how to access Tivoli® publications online and how to order Tivoli publications.

ITCAM for Applications library for Agents for WebSphere Applications, J2EE, and HTTP Servers

The following publications are included in the ITCAM for Applications library, available in the: ITCAM for Applications Information Center

- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers User's Guide*

  Provides the user overview, user scenarios, and Helps for agents for WebSphere® Applications, J2EE, and HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers Planning an Installation*

  Provides the user with a first reference point for installation or upgrade of agents for WebSphere Applications, J2EE, and HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agent for WebSphere Applications Installation and Configuration Guide*

  Provides installation instructions for setting up and configuring ITCAM Agent for WebSphere Applications on distributed systems.
- ITCAM Agent for J2EE Applications Installation and Configuration Guides:
  - *IBM Tivoli Composite Application Manager: Agent for J2EE Data Collector Installation and Configuration Guide*
  - *IBM Tivoli Composite Application Manager: Agent for J2EE Monitoring Agent Installation and Configuration Guide*

  Provides installation instructions for setting up and configuring ITCAM Agent for J2EE.
- *IBM Tivoli Composite Application Manager: Agent for HTTP Servers Installation and Configuration Guide*

  Provides installation instructions for setting up and configuring ITCAM Agent for HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers Troubleshooting Guide*

  Provides instructions on problem determination and troubleshooting for agents for WebSphere Applications, J2EE, and HTTP Servers.
- *IBM® Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers: Messaging Guide*

© Copyright IBM Corp. 2006, 2015

Provides information about system messages received when installing and using agents for WebSphere Applications, J2EE, and HTTP Servers.

- *IBM Tivoli Composite Application Manager: Agent for WebSphere Applications Reporting Guide*

  Provides information about installing Agent for WebSphere Applications Reports and creating pre-defined and ad-hoc reports.

Related publications

The following documentation also provides useful information:

- IBM Tivoli Documentation Central:

  Information about IBM Tivoli Documentation is provided on the following website:

  http://www.ibm.com/tivoli/documentation.html

- IBM WebSphere Application Server:

  Information about IBM WebSphere Application Server is provided on the following website:

  http://www.ibm.com/software/webservers/appserv/was/library

- ITCAM for Application Diagnostics library:

  Information about ITCAM for Application Diagnostics Managing Server is provided on the following website:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/index.jsp?topic=%2Fcom.ibm.itcamfad.doc_7101%2Fic-homepage.html

- IBM DB2®:

  Information about IBM DB2 is provided on the following website:

  http://www.ibm.com/software/data/sw-library/

- Tivoli Data Warehouse

  Information about Tivoli Data Warehouse is provided on the following website:

  https://www.ibm.com/developerworks/wikis/display/tivolidoccentral/Tivoli+Data+Warehouse

- IBM Tivoli Monitoring

  Information about IBM Tivoli Monitoring is provided on the following website:

  http://submit.boulder.ibm.com/tividd/td/EnterpriseConsole3.9.html

- IBM Tivoli Information Center:

  Information about IBM Tivoli products is provided on the following website:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.tivoli.tdwi.doc/toc.xml

- IBM Tivoli Composite Application Manager for WebSphere Application Server 7.2 support for WebSphere Application Server 8.5: Installation and User Guide:

  The guide is available in the Integrated Service Management (ISM) library on the following website:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.tivoli.tdwi.doc/toc.xml

- ITCAM Diagnostics Tool Installation Guide:

  The guide is available from the ITCAM for Applications Diagnostics beta. For more information about how to access the beta site, see the following website:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.tivoli.tdwi.doc/toc.xml

Accessing terminology online

The IBM Terminology website consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology .

Accessing publications online

The documentation CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center website. Access the Tivoli software information center by viewing the Tivoli software library at the following web address:

http://www.ibm.com/software/tivoli/library/

Tip: If you print PDF documents on other than letter-sized paper, set the option in the **File > Print** window that enables Adobe Reader to print letter-sized pages on your local paper.

The IBM Software Support website provides the latest information about known product limitations and workarounds in the form of technotes for your product. You can view this information at the following website:

http://www.ibm.com/software/support

Ordering publications

You can order many Tivoli publications online at the following website:

http://www.elink.ibmlink.ibm.com/public/applications/ publications/cgibin/ pbi.cgi

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to the following website:

   http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/ pbi.cgi
2. Select your country from the list and click **Go**.
3. Click **About this site** in the main panel to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate most features of the graphical user interface. For more information, see“Accessibility,” on page 789.

Tivoli technical training

For Tivoli technical training information, see the following IBM Tivoli Education website:

http://www.ibm.com/software/tivoli/education/

Supporting information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use various ways to contact IBM Software Support.

Conventions used in this guide

This guide uses several conventions for special terms and actions, and operating-system-dependent commands and paths.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, lists, items inside lists, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip**, and **Operating system considerations**)
- Keywords and parameters in text

Italic

- Words defined in text
- Emphasis of words (for example, "Use the word *that* to introduce a restrictive clause.")
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Code and other examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating-system-dependent variables and paths

The publications in this library use the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace **$***variable* with **%***variable***%** for environment variables and replace each forward slash (**/**) with a backslash (\) in directory paths. The names of environment variables are not always the same in Windows and UNIX. For example, %TEMP% in Windows is equivalent to $tmp in UNIX.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Tivoli command syntax

The following special characters define Tivoli command syntax:

[ ] Identifies elements that are optional. Required elements do not have brackets around them.

... Indicates that you can specify multiple values for the previous element. Separate multiple values by a space, unless otherwise directed by command information.

If the ellipsis for an element follows a closing bracket, use the syntax within the brackets to specify multiple values. For example, to specify two administrators for the option [**–a** *admin*]..., use **–a admin1 –a admin2**.

If the ellipsis for an element is within the brackets, use the syntax of the last element to specify multiple values. For example, to specify two hosts for the option [**–h** *host*...], use **–h host1 host2**.

| Indicates mutually exclusive information. You can use the element on either side of the vertical bar.

{ } Delimits a set of mutually exclusive elements when a command requires one of them. Brackets ([ ]) are around elements that are optional.

In addition to the special characters, Tivoli command syntax uses the typeface conventions described in "Typeface conventions" on page xii. The following examples illustrate the typeface conventions used in Tivoli command syntax:

- **wcrtpr** [**–a** *admin*]... [**–s** *region*] [**–m** *resource*]... *name*

  The *name* argument is the only required element for the **wcrtpr** command. The brackets around the options indicate they are optional. The ellipses after the **–a** *admin resource* option means that you can specify multiple administrators multiple times. The ellipses after the **–m** *resource* option means that you can specify multiple resources multiple times.

- **wchkdb** [**–o** *outfile*] [**–u**] [**–x**] {**–f** *infile* | **–i** | *object*...}

  The **–f**, **–i**, and *object* elements are mutually exclusive. Braces that surround elements indicate that you are including a required element. If you specify the *object* argument, you can specify more than one object.

Part 1. Part 1: Introduction to the agents

© Copyright IBM Corp. 2006, 2015

Chapter 1. Overview of the agents

IBM Tivoli Composite Application Manager (ITCAM) agents for WebSphere Applications, J2EE, and HTTP Servers are a monitoring, diagnostics, and management technology for WebSphere, J2EE, and HTTP servers in a distributed environment. They also provide enhanced support for monitoring Virtual Enterprise and Compute Grid products from the WebSphere XD suite. They help to maintain and improve the availability and performance of on-demand applications in your environment. It helps you to quickly locate in real time the source of bottlenecks in application code, server resources, and external system dependencies.

The agents can monitor application servers and HTTP servers at different levels. Monitoring applications incurs an unavoidable cost in terms of processing time. To minimize this cost, multiple monitoring levels are available. A minimum amount of information is collected during standard operations. As problems are encountered, the level of information that is collected can be gradually increased until the problem is located and solved.

Agent functionality

The agents can perform monitoring, diagnostics, and management functions for WebSphere, J2EE, and HTTP servers.

Monitoring

The agents can monitor the following WebSphere, J2EE, and HTTP servers:

- WebSphere servers
  - WebSphere Application Server (Network Deployment)
  - WebSphere Application Server (Extended Deployment)
  - WebSphere Process Server
  - WebSphere Portal Server
  - WebSphere ESB Server
  - WebSphere Business Process Manager Server
  - WebSphere Extreme Scale
  - WebSphere Virtual Enterprise
  - WebSphere Compute Grid
- J2EE servers
  - SAP Net Weaver
  - Oracle Application Server
  - JBoss Application Server
  - Apache Tomcat
  - Oracle/BEA WebLogic Server
  - WebSphere Application Server Community Edition
  - Java™ 2, Standard Edition
  - WebLogic Portal Server
  - Sun Java System Application Server Enterprise Edition

© Copyright IBM Corp. 2006, 2015

- HTTP servers
  - Apache Web Server
  - IBM HTTP Server
  - Sun Java System Web Server

Diagnosing

Use the agents to diagnose the following problems in your on-demand application environment:

- Hanging requests
- Lock contention problems
- Malfunctioning applications in a server farm
- Memory problems relating to garbage collection and JVM (Java Virtual Machine) heap size

Managing

Use the agents to perform the following management functions in your on-demand application environment:

- Start and stop monitored servers
- Manage servers using groups
- Configure data collector
- Use roles to restrict access to features
- Use server groups to grant access to servers
- Adjust the monitoring level at specific times based on the current work load of the server

User interfaces

Functions of the agents can be broadly divided into two areas: monitoring and diagnostics. Each of these functions uses different combinations of components. Each function also uses a different user interface. The two user interfaces are the Tivoli Enterprise Portal and the Managing Server Visualization Engine (MSVE).

The Tivoli Enterprise Portal user interface

The Tivoli Enterprise Portal is part of the IBM Tivoli Monitoring architecture. The Tivoli Enterprise Portal is the user interface into your ITCAM for Application Diagnostics environment site and possibly other IBM Tivoli enterprise applications if they are installed in your environment. For further information about the Tivoli Monitoring architecture, see “Components used by the Tivoli Enterprise Portal user interface” on page 5.

Using the Tivoli Enterprise Portal interface, you can easily monitor the health and availability of production applications and application servers. You can also quickly identify and isolate availability and response time problems. The Tivoli Enterprise Portal provides monitoring information, such as memory usage, response time, pool analysis, and data source analysis. The Tivoli Enterprise Portal enables you to *drill down* from server level metrics to specific application and resource level metrics.

MSVE user interface

The MSVE user interface provides users with management and monitoring functions for application servers. The MSVE also provides a diagnostic function. Here are some of the diagnostic activities you can perform in MSVE:

- Detect transactions failing
- Detect memory leaks
- Examine detailed method traces, which help to detect application code hotspots
- Generate reports to analyze historical information, such as application performance and OS performance

Components used by the Tivoli Enterprise Portal user interface

A number of components work together to collect, analyze, and display monitoring data in the Tivoli Enterprise Portal. These components are:

1. The Agents
2. IBM Tivoli Monitoring components

The agents

There are separate agents for WebSphere, J2EE, and HTTP servers. The agents consist of the following components:

- Tivoli Enterprise Monitoring Agent (referred to as the monitoring agent)
- Data Collector

Important: The exception is the HTTP servers agent. The HTTP servers agent does not contain a data collector. Only the WebSphere and J2EE Agents contain data collector.

The Monitoring Agent: The monitoring agent is a component of the IBM Tivoli Monitoring architecture. The purpose of the monitoring agent is to route information to the Tivoli Enterprise Monitoring Server (monitoring server) where the information is processed and presented in the Tivoli Enterprise Portal.

The Data Collector: The purpose of the data collector is to collect and route data to the monitoring agent. It is not intended to analyze or interpret data.

On each WebSphere and J2EE server that you monitor, an agent is installed; so there is a monitoring agent and a data collector running on each server you are monitoring.

For example, if you are monitoring a Tomcat server, a J2EE Agent is installed on this server. If you are monitoring a WebSphere Portal server, a WebSphere Agent is installed.

The WebSphere agent

The WebSphere agent consists of a monitoring agent and a data collector. The WebSphere monitoring agent works with the WebSphere data collector. The WebSphere data collector collects monitoring data from WebSphere servers and communicates the data to the monitoring agent.

The WebSphere Agent collects data from five primary sources:

- Response time data for application service requests from the data collector

- Resource data from the WebSphere Performance Monitoring Infrastructure (PMI)
- WebSphere Application Server log messages
- Garbage-collector activity that is recorded in the JVM verbose GC trace
- Process data from the operating system

For information about software prerequisites, see the Software product compatibility reports website to generate a variety of reports related to product and component requirements.

Tip: ITCAM Agent for WebSphere Applications version 7.2 is a component of ITCAM for Applications version 7.2. To run a report specific to ITCAM for Applications version 7.2, specify Tivoli Composite Application Manager for Applications as the product name and 7.2 as the version.

To view the system requirements for ITCAM Agent for WebSphere Applications version 7.2, see the Detailed system requirements report.

The J2EE agent

The J2EE agent is composed of a monitoring agent and a data collector. The J2EE agent works with the J2EE data collector to collect performance data from J2EE application servers. The J2EE data collector collects monitoring data from J2EE servers and communicates the data to the monitoring agent. The J2EE agent collects data from five main sources:

- Response time data for application service requests from the data collector
- J2EE application server log messages
- Garbage collection activity that is recorded in the JVM verbose GC trace
- Resource data from the application server mbean
- Process data from the operating system

For information about software prerequisites, see J2EE agent V7.1.

The HTTP agent

The HTTP agent is composed of a monitoring agent only. The monitoring agent can monitor the following HTTP servers:

- Apache Web Servers
- IBM HTTP Web Servers
- Sun Web Servers

The HTTP agent collects performance data about the web servers and websites in the following ways:

Apache Server and HTTP Server: The agent modifies the Apache server and IBM HTTP server configuration files to include the monitoring module. The monitoring module is loaded dynamically during web server startup. The module receives all HTTP requests and reports data to the HTTP agent. In addition, the HTTP agent parses static information from the configuration file.

Sun Web Server: The HTTP agent collects monitoring data from Sun Web servers by polling the SNMP service for web server-related statistics. It also parses web server configuration files to get information that is not provided by the SNMP service.

For information about software prerequisites, see HTTP agent V7.1.

IBM Tivoli Monitoring components

IBM Tivoli Monitoring monitors the performance and availability of distributed operating systems and applications. IBM Tivoli Monitoring products are based on a set of common service components. These service components provide security, data transfer and storage, notification mechanisms, user interface presentation, and communication services in an agent-server-client architecture. Some of these service components are shared by other products, including IBM Tivoli OMEGAMON® XE mainframe monitoring products, ITCAM for Application Diagnostics, and ITCAM for Applications.

The service components shared by ITCAM for Application Diagnostics and IBM Tivoli Monitoring are as follows:

- Tivoli Enterprise Monitoring Server (referred to as the monitoring server)
- Tivoli Enterprise Portal Server (referred to as the Portal Server)
- Tivoli Enterprise Portal
- Tivoli Enterprise Monitoring Agent (referred to as the monitoring agent)

ITCAM for Application Diagnostics uses the service components of IBM Tivoli Monitoring. The ITCAM for Application Diagnostic monitoring agents integrate with components in the IBM Tivoli Monitoring environment by retrieving data from the monitoring agents and forwarding it to the Portal Server where it is displayed in the Tivoli Enterprise Portal.

Here is some further information regarding the shared service components and how they integrate with ITCAM for Application Diagnostics:

Tivoli Enterprise Monitoring Server: The Monitoring Server performs the following functions:

- Acts as a collection and control point for alerts that are received from the monitoring agents
- Tracks the heartbeat request interval for all monitoring agents connected to it
- Stores, initiates, and tracks all situations and policies, and is the central repository for storing all active conditions on every monitoring agent
- Initiates and tracks all generated actions that start a script or program on the monitoring agent

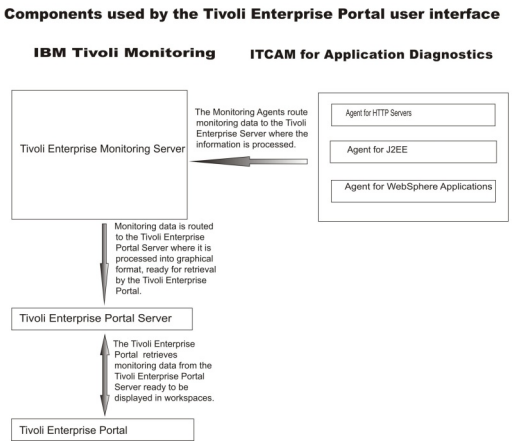
Tivoli Enterprise Portal Server: The Portal Server performs the following functions:

- Acts as a repository for all graphical presentations of monitoring data
- Provides the core presentation layer, which allows for the retrieval, manipulation, analysis, and reformatting of data
- Manages data access through user workspace consoles

Tivoli Enterprise Portal: The Tivoli Enterprise Portal is a Java user interface that connects to the monitoring server and displays monitoring data. The Tivoli Enterprise Portal can be launched from an Internet Explorer browser, or can be installed as a client application on a workstation. The Tivoli Enterprise Portal is one of the user interfaces for ITCAM for Application Diagnostics, the other user interface is the MSVE.

Tivoli Enterprise Monitoring Agent: The monitoring agents are responsible for data gathering. The monitoring agents communicate monitoring data to the Monitoring Server and the managing server. In ITCAM for Application Diagnostics, the WebSphere, J2EE, and HTTP agents contain monitoring agents.

The following diagram shows the component used by the Tivoli Enterprise Monitor:



For more information about the ITCAM for Application Diagnostics agents, see "The agents" on page 5. For more information about the IBM Tivoli Monitoring, see IBM Tivoli Monitoring Information Center

Components used by the MSVE user interface

A number of components work together to collect, analyze, and present the monitoring data in the MSVE. These components are:

- Managing Server
- Data Collector

The Managing Server is the central component of ITCAM for Application Diagnostics. For every implementation of ITCAM for Application Diagnostics, there is one Managing Server. The Managing Server is a powerful technology that provides *deep dive* functions. The user interface for the Managing Server is the MSVE. The Managing Server works with data collector, for each server being monitored there is one data collector installed. The data collector collects performance data from the application servers and HTTP servers and forwards this information to the Managing Server.

ITCAM for Application Diagnostics Managing Server provides deep dive diagnosis capabilities. The Managing Server provides the following functions:

Table 1. Managing Server Functionality

Area	What you can do
Server and Account management	• Manage servers using groups • Configure data collector • MOD management • Use roles to restrict access to features

Table 1. Managing Server Functionality (continued)

Area	What you can do
Systems Overview	• Display the availability of application servers • Provide comparisons between current response times and baseline response times • Provide application server-level statistics for quick assessment of server activity and related platform data • Provide system resources
Server Activity	• Use In-Flight Request Search and Server Activity Display – Active Requests to locate malfunctioning applications • View transactions in progress • Evaluate the current performance of your applications • Spot hanging transactions • Troubleshoot and fix hanging transactions • Solve lock contention problems • Access JVM thread data • Use memory diagnostic tools to allocate memory problems inside applications
Recent Activity	• Investigate and fix potential memory problems relating to garbage collection and the JVM heap size • Tune the JVM parameters • Find evidence of memory leaks
Performance Analysis and Reporting	• Generate reports • Analyze historical data
Problem Center	• View high priority trap alerts and Tivoli Enterprise Portal events • Use historic data to analyze performance problems found in traps and Tivoli Enterprise Portal situation events
Composite Transactions	• Monitor transactions and analyze the method flow using method trace, stack trace, and request information
Monitoring on Demand	• Create a schedule that alters the monitoring level based on a date and time when a server needs more detailed monitoring • Adjust the monitoring level at specific times based on the current work load of the server • Override the monitoring level or change the schedule for a selected server

Managing Server components

The Managing Server is J2EE application that is configured within WebSphere Application Server. The Managing Server works with a DB2 or Oracle database. It is designed for scalability and load balancing, and there are many ways to implement an installation across one or more servers. The Managing Server consists of the following components:

Kernel: The kernel controls the Managing Server. The kernel registers components as they join the Managing Server. The kernel periodically renews connections and registrations with components and data collector and collects server and component availability information.

Publish Server: The publish server receives data from the data collector and aggregates it based on different needs.

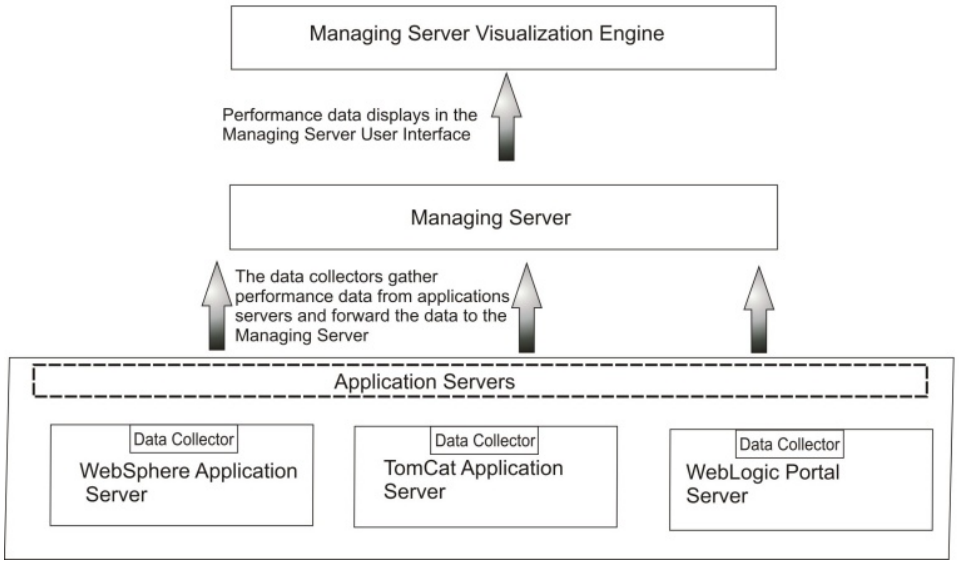
Archive Agent: The archive agent collects data from the publish server and archives it into the database for reporting.

Message Dispatcher: The message dispatcher sends out emails of performance reports and trap actions from the Performance Analysis and Reporting and the Trap and Alert Management features.

Global Publish Server (GPS): The global publish server tracks composite requests, as they move from one server to another.

The following diagram displays the component used by the MSVE:

Components used by the Managing Server Visualisation Engine



Tivoli Enterprise Portal interoperation with MSVE

The Tivoli Enterprise Portal and MSVE interoperate in a number of ways. When monitoring and troubleshooting your application environment, you can move between the Tivoli Enterprise Portal and MSVE as you can use links in the Tivoli Enterprise Portal workspaces to run in context into MSVE. In the MSVE, more detailed information is provided about individual transactions as they occur. This detailed information assists you in diagnosing problems. The Tivoli Enterprise Portal provides monitoring information, so in the Tivoli Enterprise Portal you are alerted that a problem occurred with a server, an application, or a resource. If you want to analyze problems in greater detail, you can link in context to MSVE.

Another way in which the Tivoli Enterprise Portal and the MSVE interoperate is through situations in the Tivoli Enterprise Portal and alerts in the MSVE. When a situation is triggered in the Tivoli Enterprise Portal, this is displayed as an alert in the MSVE. Alerts and Events page. All situations can be escalated to problems and then can be evaluated in the Problem Center.

For further information about configuring Single Sign on between Tivoli Enterprise Portal and MSVE, see the *ITCAM for Application Diagnostics: Installation and Configuration Guide*.

What is new

ITCAM for Application Diagnostics version 7.1 adds the following new features:

- Ability to monitor WebSphere Extended Deployment environment.
- Ability to launch in context from Tivoli Enterprise Portal workspaces into the Managing Server Visualization Engine (MSVE). This enables users to examine individual transactions in more detail.
- Rephrased data table names in Server Activity Display in MSVE makes them clearer to users.
- Improved MSVE menus to indicate if data is real-time data or historic data.
- In MSVE, ability to link from active requests to the Stack Trace page.
- New predefined situations in the Tivoli Enterprise Portal.
- Improved Summary Workspaces functionality in the Tivoli Enterprise Portal.

ITCAM for Application Diagnostics version 7.1.0.1 adds the following features:

- Integration with ITCAM for Transactions using the Transaction Tracking API (TTAPI) is improved:
  - ITCAM for Transactions displays JDBC and JNDI nested request exceptions.
  - ITCAM for Transactions displays WebSphere Portal requests.
  - ITCAM for Transactions displays WebSphere MQ version 7 JMS requests.
  - ITCAM for Transactions displays JMS messaging topology.
  - ITCAM for Transactions avoids duplicate transaction tracking in several cases, including presence of an ITCAM for Service Oriented Architecture agent on the same server as the ITCAM for Application Diagnostics agent.
- Integration with Optim™ Performance Manager is now available.
- Assignment of Managing Server components (Publish Servers) to data collector based on IP address and host name is now available. This optimizes network usage and performance on multiple-site deployments.
- ITCAM for Application Diagnostics supports IBM Tivoli Monitoring version 6.2.2 Fix Pack 2.
- Upgrading the Managing Server from the version supplied with ITCAM for WebSphere version 6.1 and ITCAM for J2EE version 6.1 is now fully automatic.

ITCAM for Application Diagnostics version 7.1.0.2 adds the following features:

- ITCAM Agent for WebSphere Applications can monitor Classic and Open Service Component Architecture (SCA) requests.
- ITCAM Agent for WebSphere Applications can monitor WebSphere Extreme Scale zones. Statistics on grids, servers, map sets, and individual shards are available, and detection of hung JVM threads.
- An installation routine is now available for WebSphere Hypervisor Edition, either stand-alone or in a WebSphere Cloud Architecture environment. You can deploy ITCAM Agent for WebSphere Applications to a template or pattern, and it is deployed automatically on the newly created virtual machines.
- ITCAM Agent for WebSphere Applications can check and compare Business Level Application (BLA) binary files on monitored servers.

- You can now upgrade monitoring of a WebSphere Application Server instance from ITCAM for WebSphere Application Server version 7.0 or version 7.1 (made available with WebSphere Application Server) to ITCAM Agent for WebSphere Applications.

Getting Started with ITCAM for Application Diagnostics

Depending on your requirements, when you use ITCAM for Application Diagnostics, you use either the MSVE or the Tivoli Enterprise Portal user interfaces, or a combination of both. For further information about using and navigating the Tivoli Enterprise Portal and the MSVE, see the *ITCAM for Application Diagnostics: User Guide.*

For further information about installing ITCAM for Application Diagnostics, see the following publications:

ITCAM for Application Diagnostics: WebSphere Monitoring Agent Installation and Configuration Guide

ITCAM for Application Diagnostics: J2EE Monitoring Agent Installation and Configuration Guide

ITCAM for Application Diagnostics: J2EE Data Collector Installation and Configuration Guide

ITCAM for Application Diagnostics: HTTP Monitoring Agent Installation and Configuration Guide

ITCAM for Application Diagnostics: Managing Server Installation and Configuration Guide

Chapter 2. Scenarios

The following scenarios describe how the agents might be used to monitor and diagnose problems in your application environment.

The following fictitious personas are used in the scenarios to reflect typical positions in your organization:

Table 2. Personas

Persona	Description
Annette – Level 2 Operator	Annette's main focus is to find which component is down, which components are affected, and the location of the problem. In addition, Annette follows procedures to correct the problem. If Annette cannot fix the problem within a time limit stipulated by her service level agreement (SLA), she produces a trouble ticket and escalates the problem.
Jim – Middleware/Application Support Subject Matter Expert (SME)	Jim's main focus is to ensure that the middleware applications he is responsible for are functional at all times. If an application goes down, then a line of business is affected and has a direct affect on how Jim's team is rated against their SLA. Jim also works with the systems monitoring and automation group to define the appropriate monitors and thresholds for his domain area of responsibility.
Dave – Application Developer	Dave's main focus is to develop in-house applications. When a problem arises in a production application, he is sent trace files so that he can analyze the problem, which he then tries to simulate in his environment.
Simon – Operating System Specialist	Simon's main focus is to work with the Systems Management team to define what the base OS services/daemons, ports, file systems, and logs are that must be monitored on every computer. He is also a recognized expert in cluster configurations.

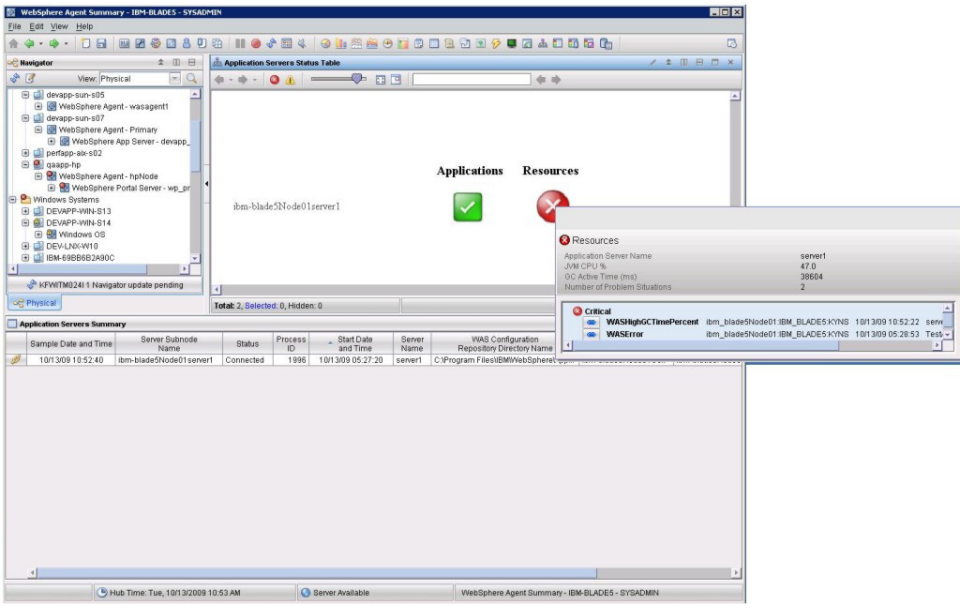
Scenario 1: Diagnosing a memory leak

Users are reporting slow response times for an application. A user contacts the help desk and raises a ticket for slow response time in relation to an application. Annette, the level 2 operator, picks up the ticket.

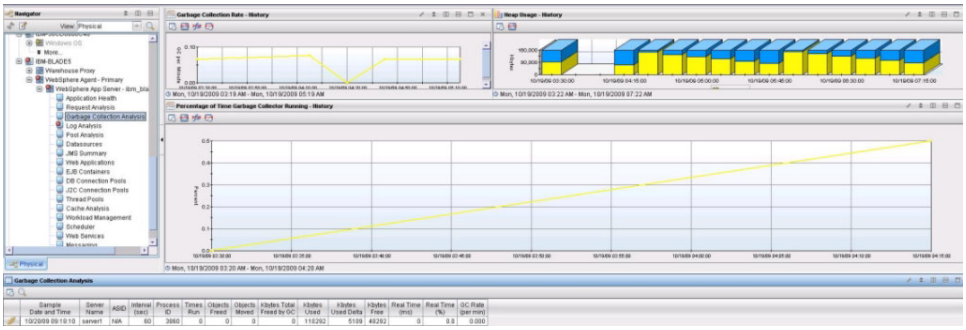
Important: This scenario assumes that you have ITCAM for Application Diagnostics, which includes the Managing Server.

1. Annette navigates to the Tivoli Enterprise Portal and notices that the **Resources** icon is displaying a Critical symbol. On the **Resources** icon flyover, the GC Active Time (ms) metric is displaying a high value and also the WASHighGCTimePercent situation has triggered. This indicates that the JVM garbage collection is taking too much time.

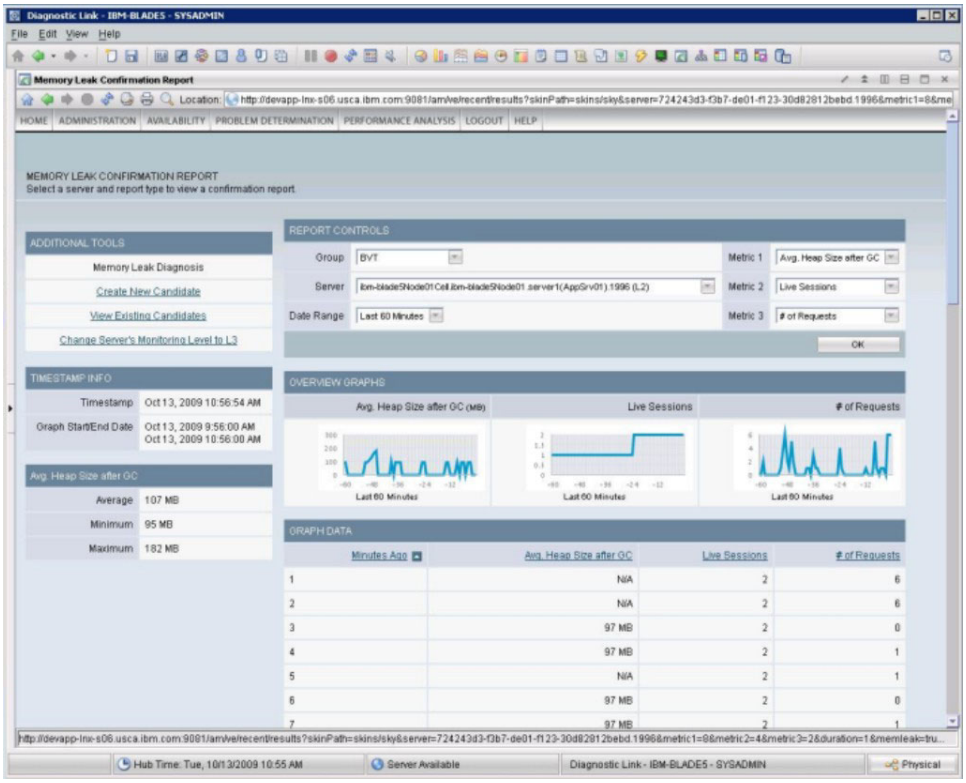
© Copyright IBM Corp. 2006, 2015



2. In the Garbage Collection Analysis workspace, Annette sees that the Percentage of Time Garbage Collector Running - History view displays an increasing trend. The increasing trend suggests that the heap is insufficient for the demand being put on it by applications. The Running - History graph, which displays the percentage of real time that the garbage collector was running during the current interval for each server region, shows an increasing trend. This trend suggests that either the heap size is insufficient for the demand that applications are putting on it, or that there is a memory leak.



3. Annette uses an external ticketing tool to route the ticket to Jim, the Middleware/Application SME, for further observance and investigation.
4. Jim notices a problem ticket from Annette involving excessive garbage collection times. Jim navigates to the Garbage Collection Analysis workspace and confirms the problem. He requires more detailed information to diagnose the cause of the problem. He clicks the **Diagnostic Memory Leak** link in the Garbage Collection Analysis workspace to open the Memory Leak Confirmation report page in MSVE.



5. From examining the Memory Leak Confirmation report page in MSVE, Jim suspects that there is a memory leak.
6. To determine if there is a memory leak, Jim sets the monitoring level to L3 and enables memory leak BCI by completing the following steps:
   - Edits the file: `$DC_HOME/runtime/appserver_version.node_name.server_name/custom/toolkit_custom.properties` file and sets the property `com.ibm.tivoli.itcam.toolkit.ai.enablememoryleakdiagnosis=true`.
   - Uncomments this line to enable Memory Leak Diagnosis (the path in the line might be slightly different):`am.camtoolkit.gpe.customxml.leak=/opt/IBM/AD710/aix533/yn/wasdc/7.2/itcamdc/etc/memory_leak_diagnosis.xml`
   - Restarts the data collector.

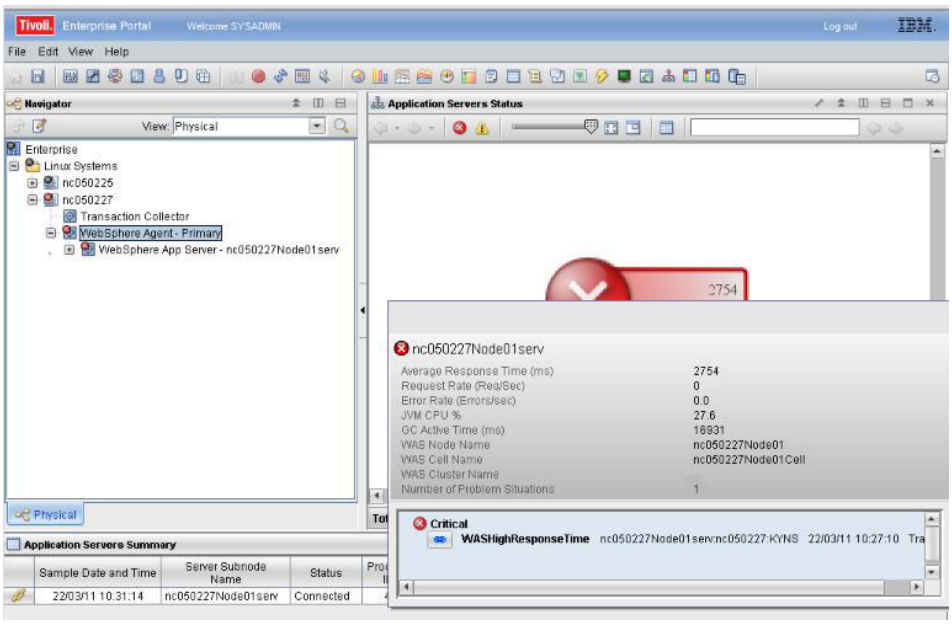
   Jim forwards the problem to Dave, the application developer. Dave works to resolve the problem. This action is outside the scope of ITCAM for Application Diagnostics.

Scenario 2: Diagnosing hanging transactions

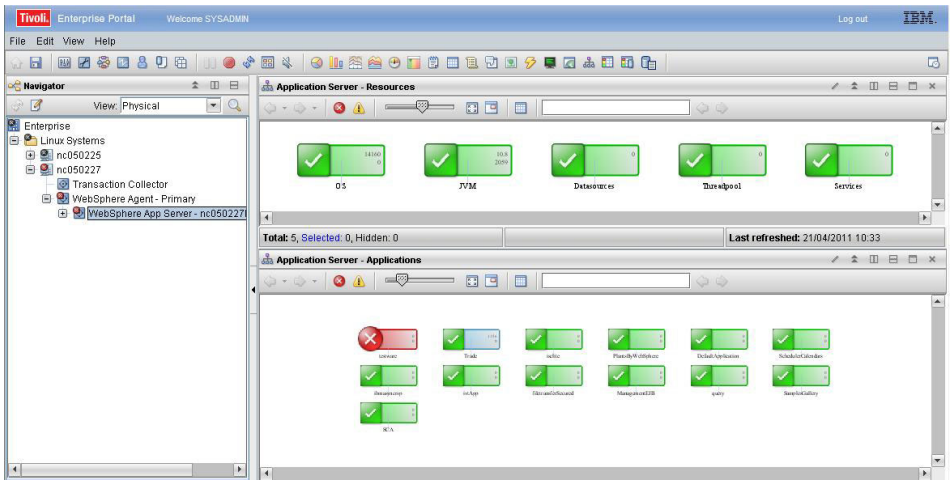
Annette, the level 2 operator, receives an email indicating that a situation triggered in the Tivoli Enterprise Portal. The situation indicates that response time is slow for an application.

Important: This scenario assumes that you have ITCAM for Application Diagnostics, which includes the Managing Server.

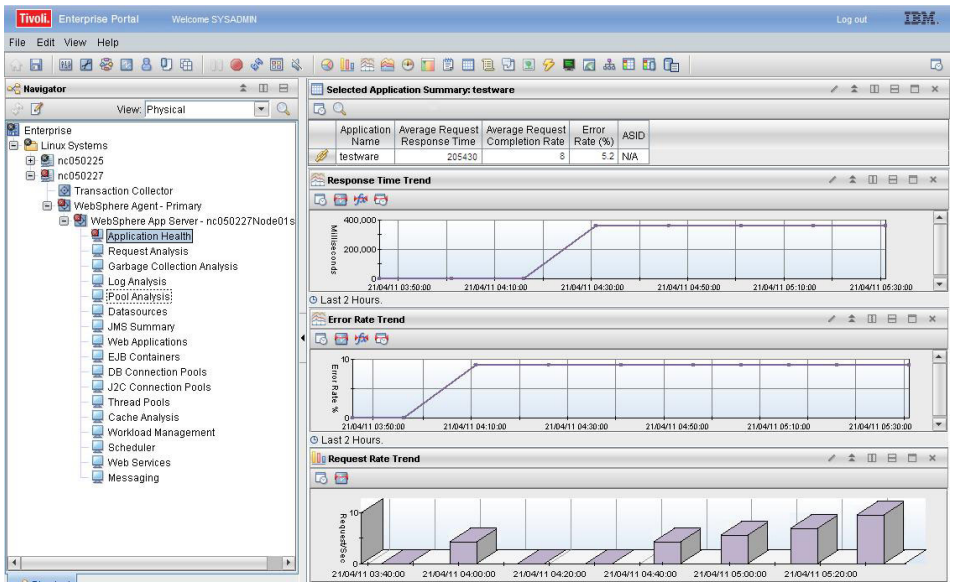
1. In the Tivoli Enterprise Portal, Annette points to the critical icon in the WebSphere Agent Summary Status workspace. Annette sees in the flyover that the WASHighResponseTime situation triggered.



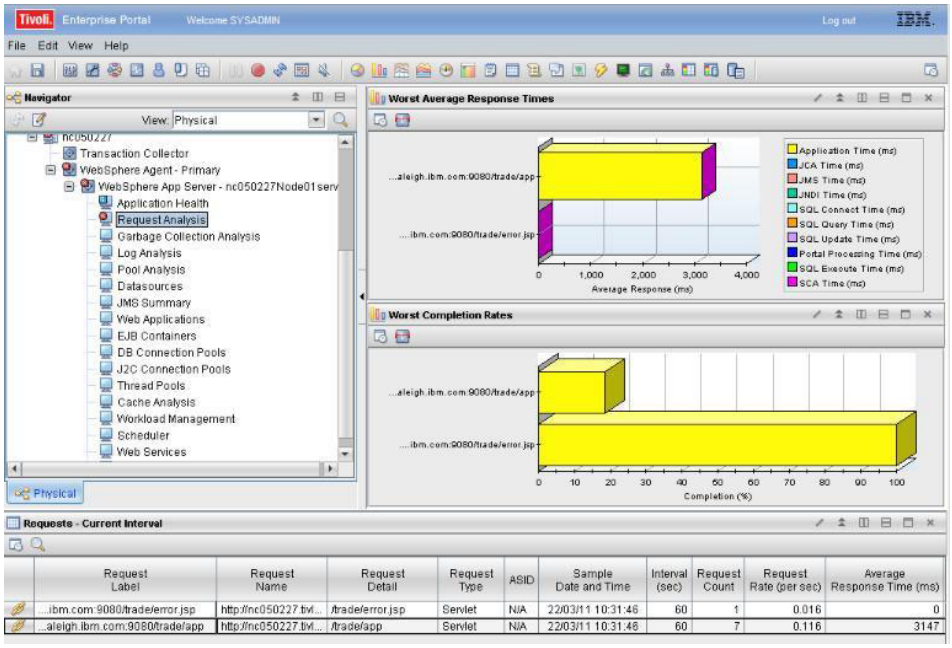
2. Annette double-clicks the critical icon and the Application Server Summary workspace opens.



3. Annette double-clicks the critical application icon and the Application Trend workspace is displayed. In this workspace, the Application Summary report displays information about response time, error rate, and request rate.

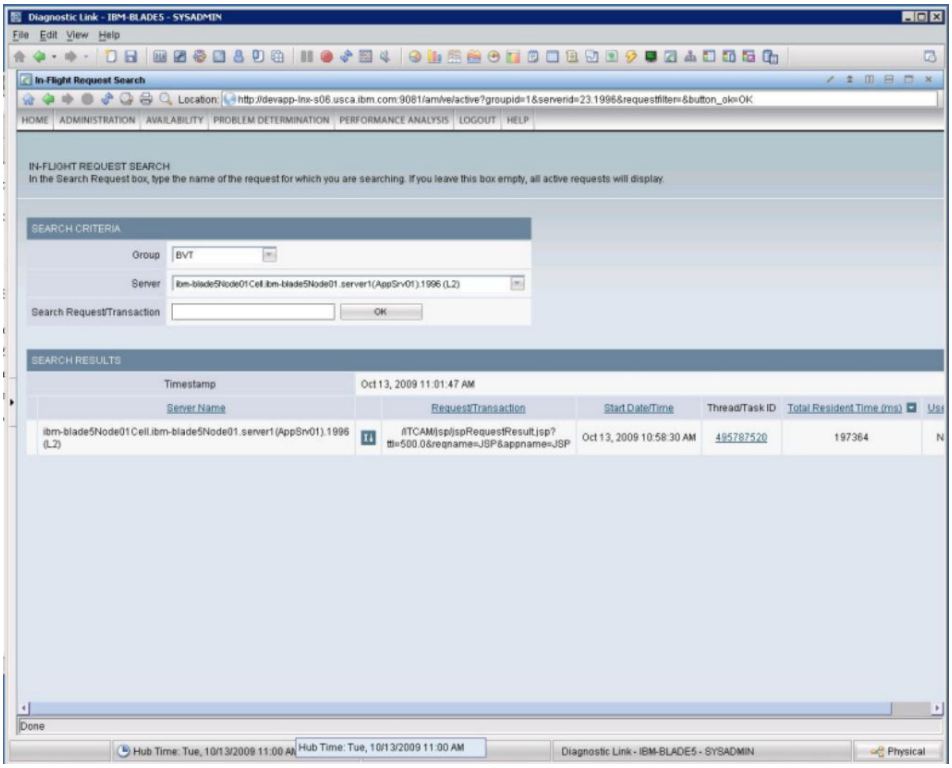


4. For a more detailed analysis of requests, Annette navigates to the Request Analysis workspace, by using the navigation tree. The Request Analysis workspace displays information about worst average request response time and worst average request completion rate. Annette observes that the average request response time is high and that the average request completion rate is low (for some of the requests).

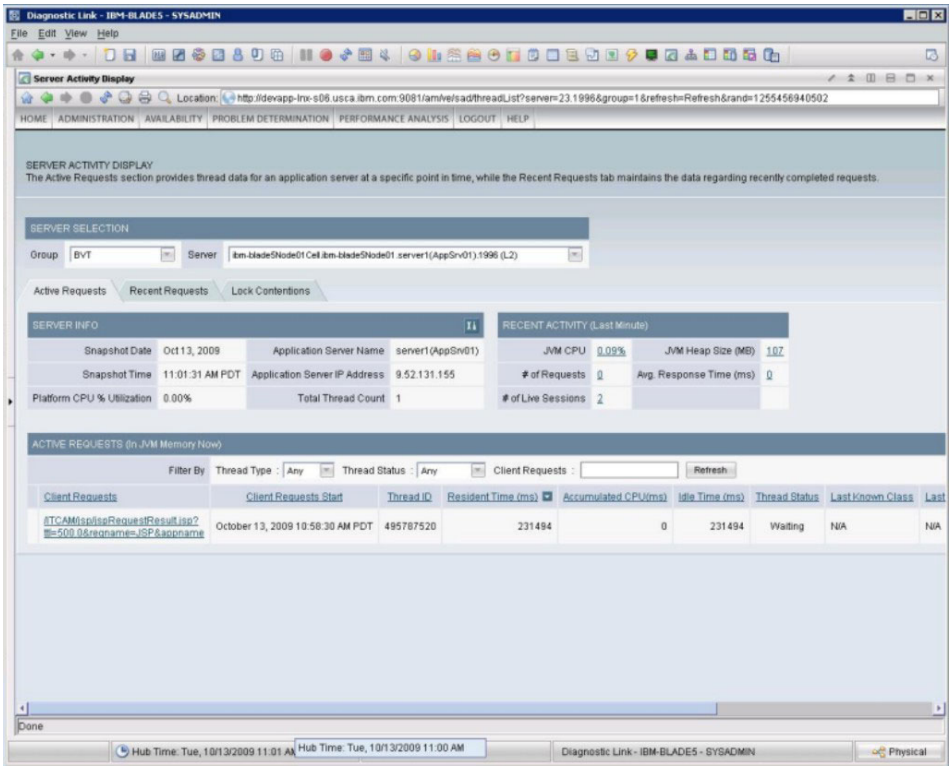


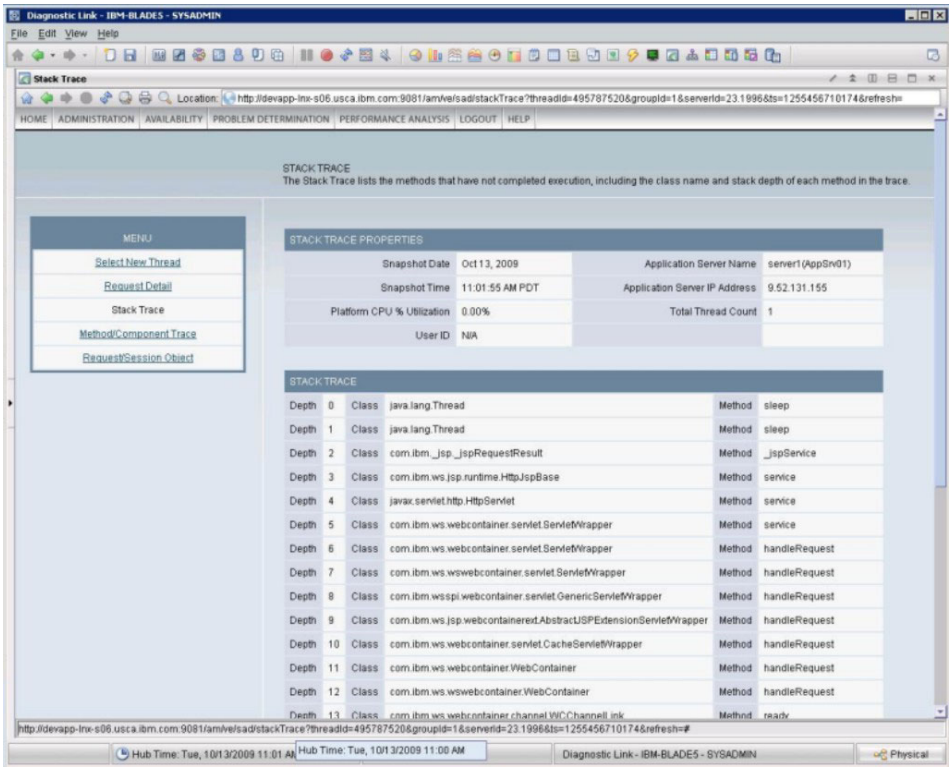
5. Annette uses an external ticketing tool to route the ticket to Jim, the Middleware/Application Support SME, for further observance and investigation.
6. Jim notices a problem ticket from Annette indicating slow response times. Jim navigates to the Request Analysis workspace in the Tivoli Enterprise Portal and confirms the slow request response time for the application.
7. To see individual hanging transactions, Jim must open MSVE. Jim right-clicks the chain icon for the request and selects the Diagnostic In-Flight Request Search link. This link displays the In-flight Request Search page in the MSVE.

Server and request information is carried over from the Request Analysis workspace.



8. From here, Jim can diagnose hanging requests and see the stack trace for that request by clicking the server activity display.



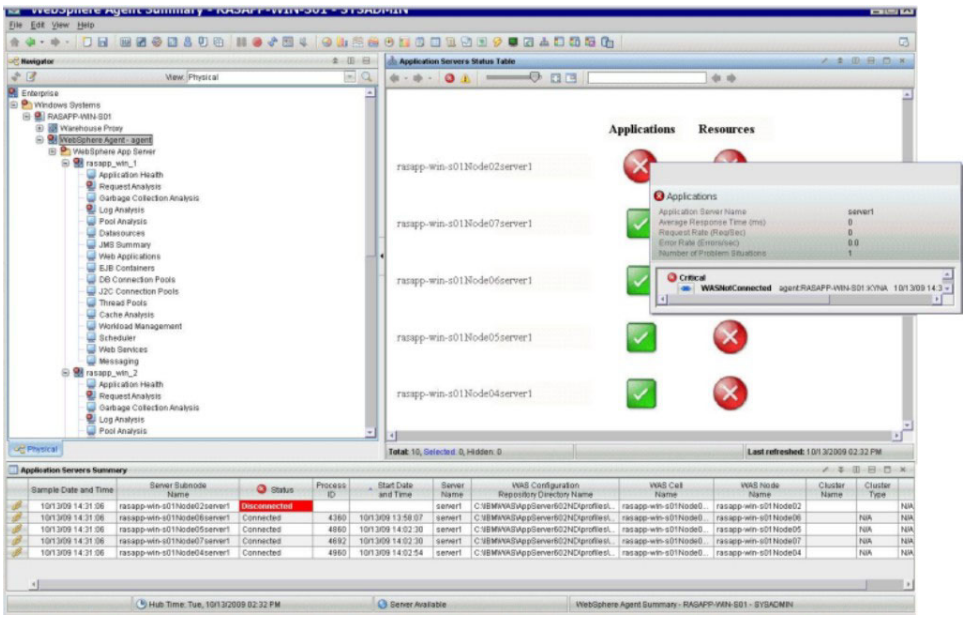


9. Jim does not have enough information in the stack trace to debug the problem, so he enables L3 monitoring. The `toolkit_custom.properties` file has already been enabled for method entry/exit trace.
10. Next, Jim enables L3 monitoring by clicking **Administration** > **Monitoring on Demand** and clicking **Schedule Change/Override**.
11. Jim returns to the In-Flight Request Search page and clicks the Thread/Task ID for the request.
12. Jim clicks the **Method/Component Trace** link to display method trace details.
13. Jim then forwards details to Dave, the application developer, by clicking **Email PDF**. Dave works to resolve the problem. This action is outside the scope of ITCAM for Application Diagnostics.
14. After collecting the data, Jim returns the data collector to L1 monitoring.

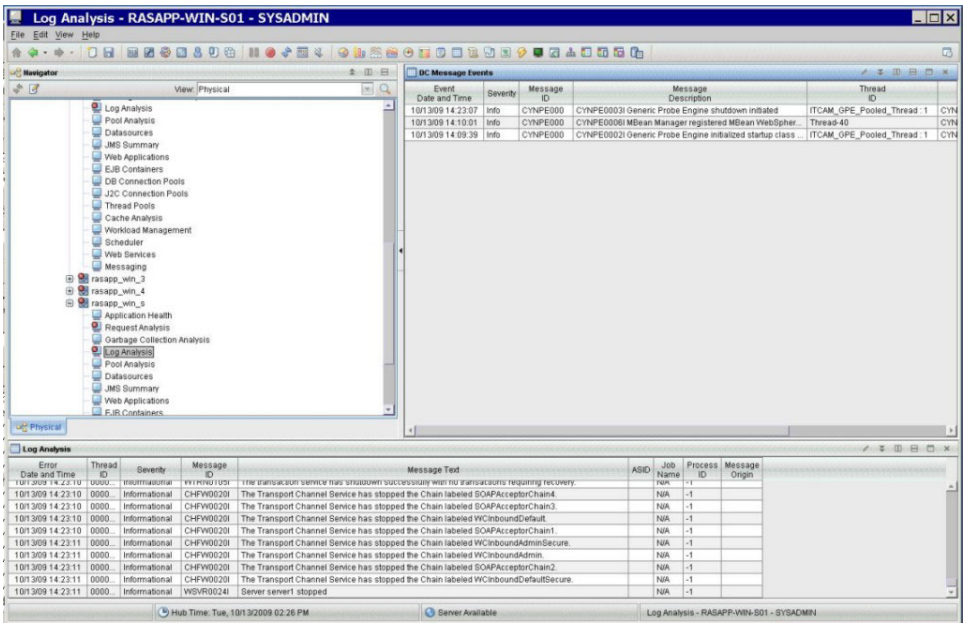
Scenario 3: Diagnosing a WebSphere server shutdown

Annette, the level 2 operator, receives a severity 1 ticket indicating that users cannot access an application.

1. Annette navigates to the Tivoli Enterprise Portal where (in the WebSphere Agent Summary Status workspace) the Applications icon is displaying a critical symbol. The flyover on the application icon shows that a WasNotConnected situation triggered. The application server summary also shows the server status as “Disconnected.”



2. Annette navigates to the Log Analysis workspace. This workspace reports application server errors and exception conditions that are recorded in the WebSphere Application Server `SystemOut.log` file log. The information in this workspace includes the exception severity of errors, and the ID and text of the associated message.
3. Annette observes that in the Log Analysis report, the Process ID value is displayed as `-1`. This value indicates that the data collector is disconnected. If a WebSphere server shutdown occurs, the connection between the data collector and Tivoli Enterprise Monitoring Agent is closed. However, the data collector continues to write to log files and Tivoli Enterprise Monitoring Agent processes these records but sets the PID value to `-1`.

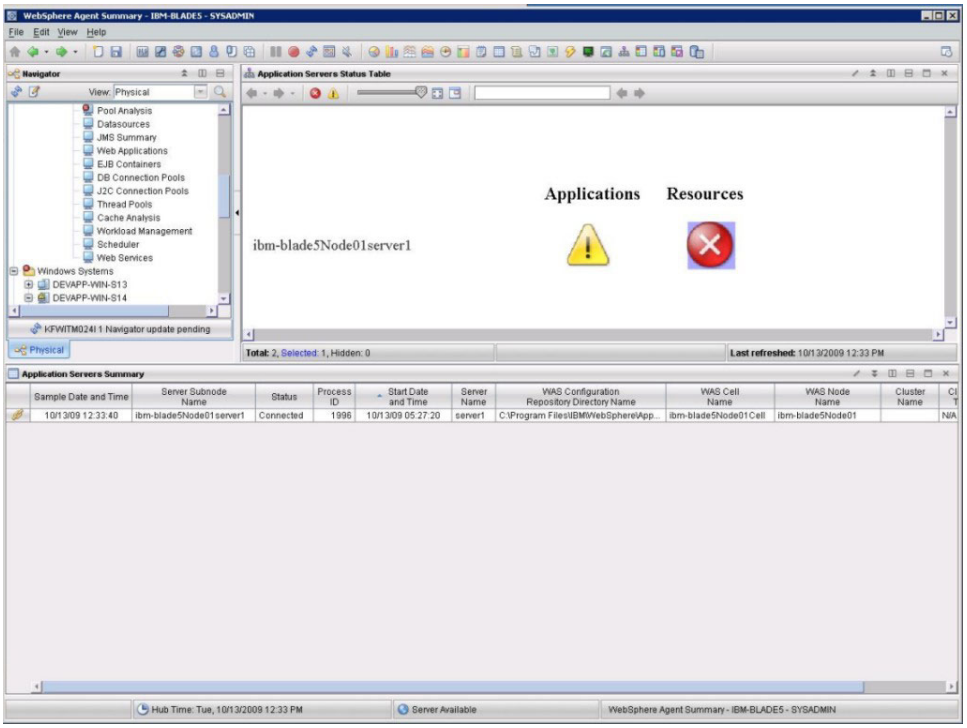


4. Annette uses an external ticketing team to forward the ticket to Jim the Middleware/Application Support SME. Jim investigates the cause of the WebSphere server shutdown and initiates a restart of the WebSphere Application Server.

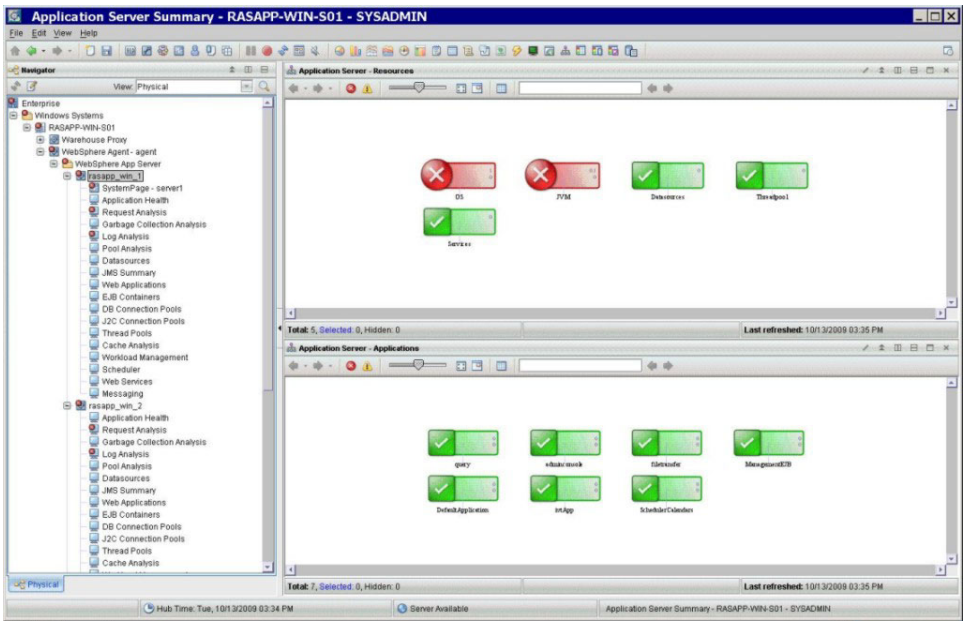
Scenario 4: Determining if the WebSphere cluster needs to be load balanced

Annette, the level 2 operator, gets a number of tickets relating to slow response time for an application. Annette receives an email indicating that the WASHighCPUPercentUsed situation triggered on the WebSphere Application Server where the application is hosted.

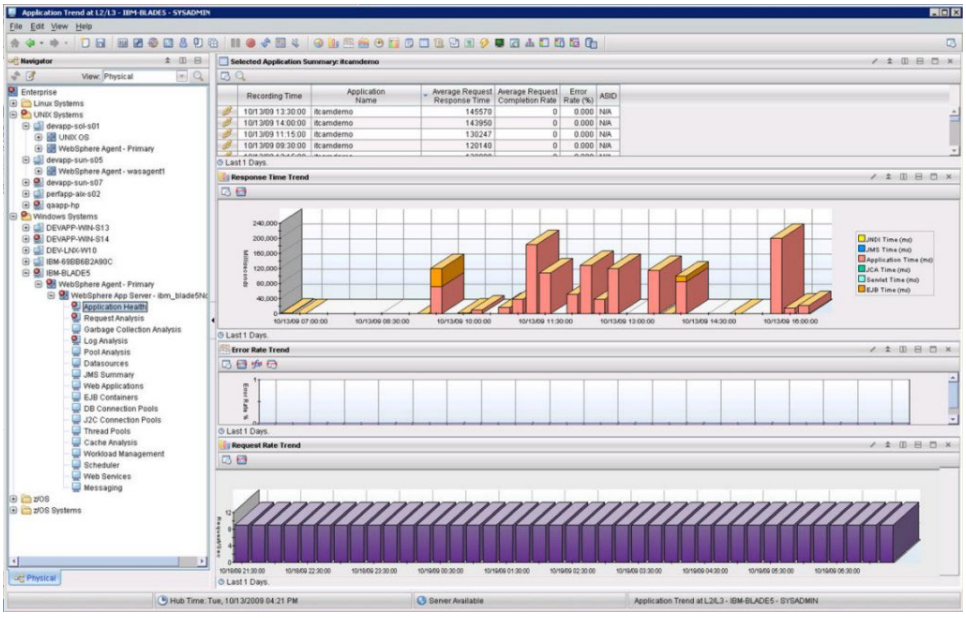
1. Annette navigates to the Tivoli Enterprise Portal and observes that on the WebSphere Agent Summary workspace the **Application** icon is displaying a warning symbol. The **Resources** icon is displaying the critical symbol. The **Resource** icon flyover is displaying high JVM CPU%.



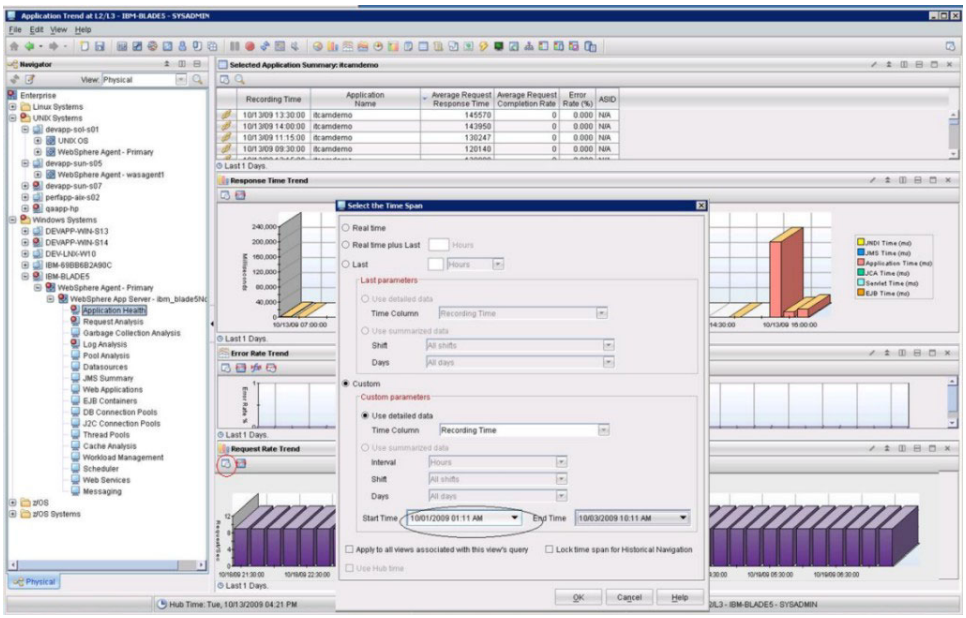
2. Annette double-clicks the **Resources** icon and sees that the **OS** icon and JVM icon are both displaying the critical symbol. The **OS** icon flyover is displaying System CPU (ms) as high. The JVM icon also displays the JVM CPU% as high.



3. Annette double-clicks the **Application** icon. The Application Trend at L2/L3 workspace is displayed. The Selected Application Summary report displays the application name, average request response time, average request completion rate, and error rate. The average request response time is high. The Request Rate Trend chart displays the number of requests that are completed per second for the application. Again, this value is displaying as high.



4. Before Annette escalates this problem, she must determine if the problem is recent or if it has been occurring for some time. Annette checks the trend by taking the following steps in the Request Rate Trend chart:
   a. In the Application Trend at L2/L3 workspace, she selects the **Specify time span for query** icon. The Select the Time Span window is displayed.
   b. In the Custom Parameters section, she enters the required values in the **Start Time** and **End Time** fields. Then, she clicks OK.



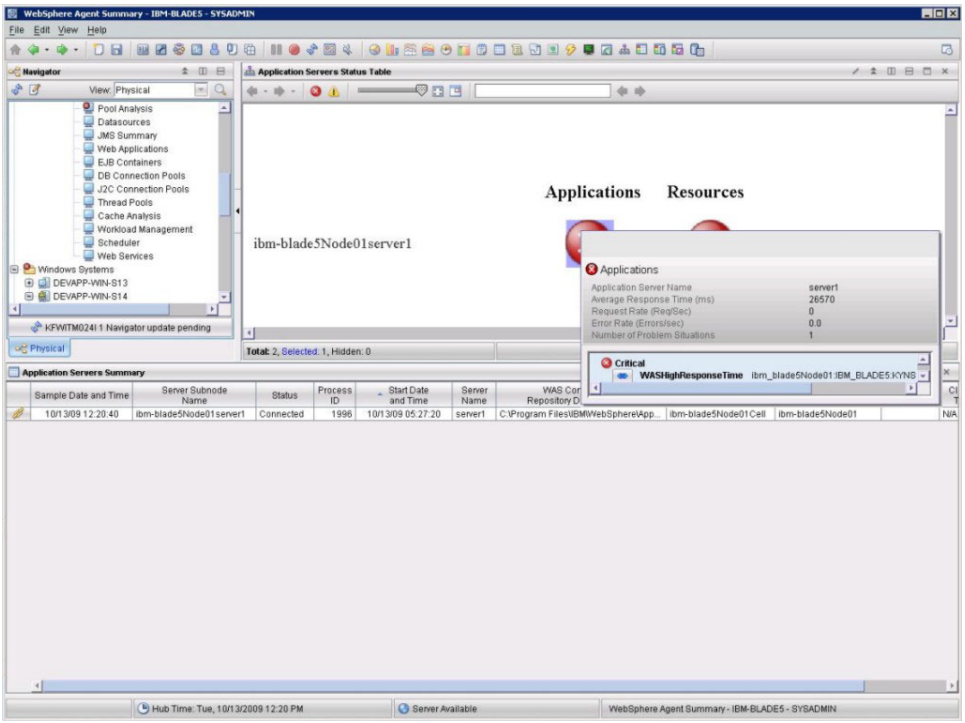
5. Annette observes that there was an increase in client requests a few days ago and that this value has remained high throughout the week. Further investigation reveals that a surge of new customers caused a large increase in new users on the system. As a result, the load on the system is high.
6. Annette uses an external ticketing tool to forward the ticket with all details to Jim, the Middleware/Application Support SME.
7. Jim immediately sees from what Annette has reported that the system is over-burdened because of a significant increase in new users, and that the number of servers that are available in the cluster must be increased. Jim forwards the ticket to Simon, the OS SME.
8. Simon must determine if the Application is running on a static or a dynamic WebSphere cluster. If the application is running on a static cluster, he adds additional application servers. If the application is running on a dynamic cluster, he increases the number of servers allowed. These actions are outside the scope of ITCAM for Application Diagnostics.

Scenario 5: Determining the cause of high response times

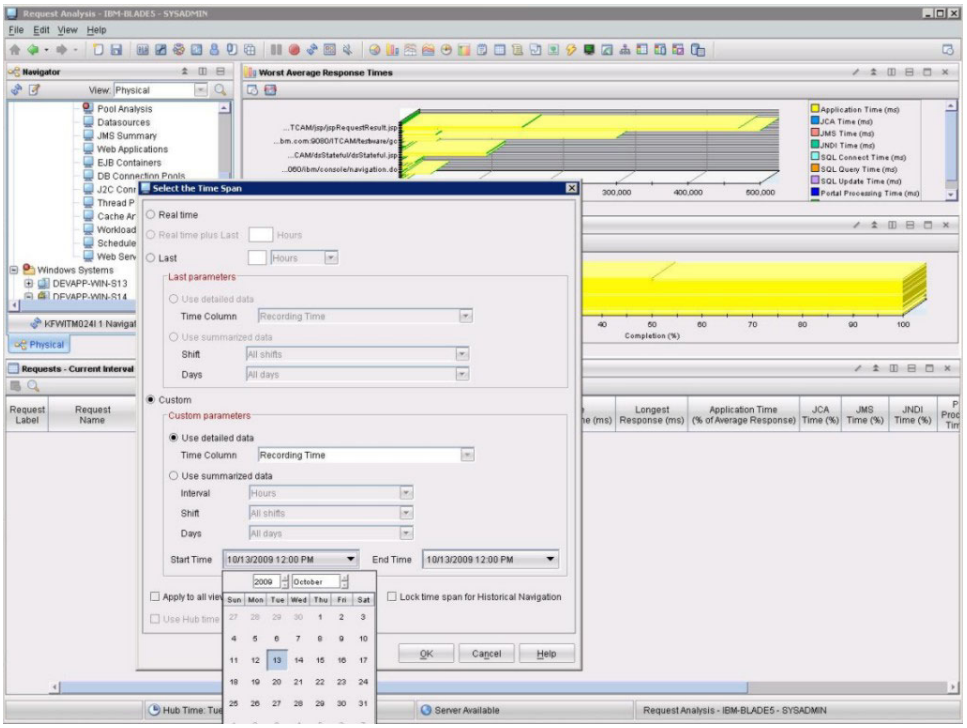
Annette, the level 2 operator, receives an email that indicates that the WASHighResponseTime situation triggered for an application.

Important: This scenario assumes that you have ITCAM for Application Diagnostics, which includes the Managing Server.

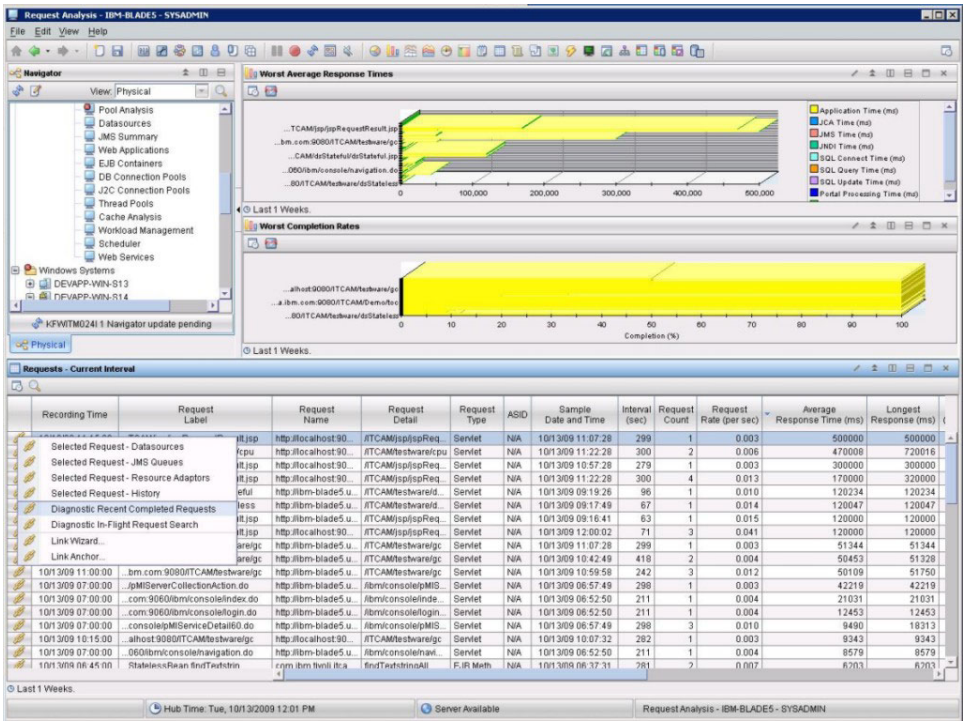
1. Annette navigates to the Tivoli Enterprise Portal and notices in the Application Server Summary workspace that the icon for the application is displaying a critical icon. The flyover for the application indicates that the Average Response time (ms) is high. Annette must determine for how long the response time has been high.



2. Annette double-clicks the **Application** icon. The Application Trend at L1 workspace is displayed. Annette requests historical data by taking the following steps:
   a. In the Requests - Current Interval View, she clicks the Specify time span for query icon. The Select the Time Span window is displayed.
   b. In the Custom Parameters section, she enters the required values in the **Start Time** and **End Time** fields and she clicks OK.
   c. She sorts by the Average Response Time column.

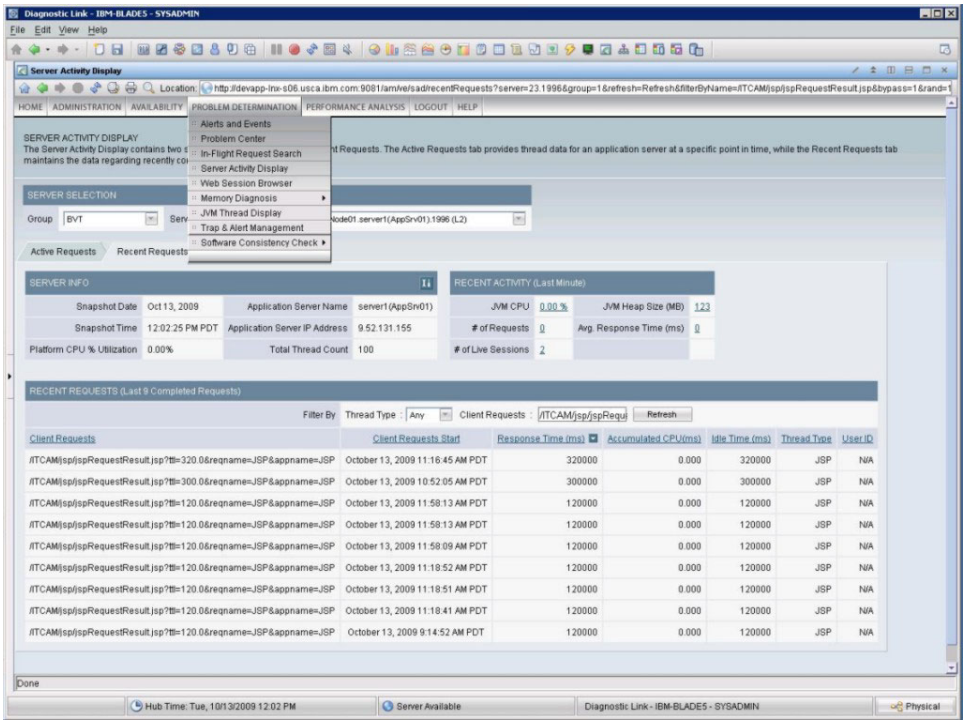


3. Annette uses an external ticketing tool to forward the trend details to Jim, the Middleware/Application Support SME.
4. Jim receives this problem ticket about high response times for a particular application. Jim navigates to the Request Analysis workspace and confirms the problem that Annette described.

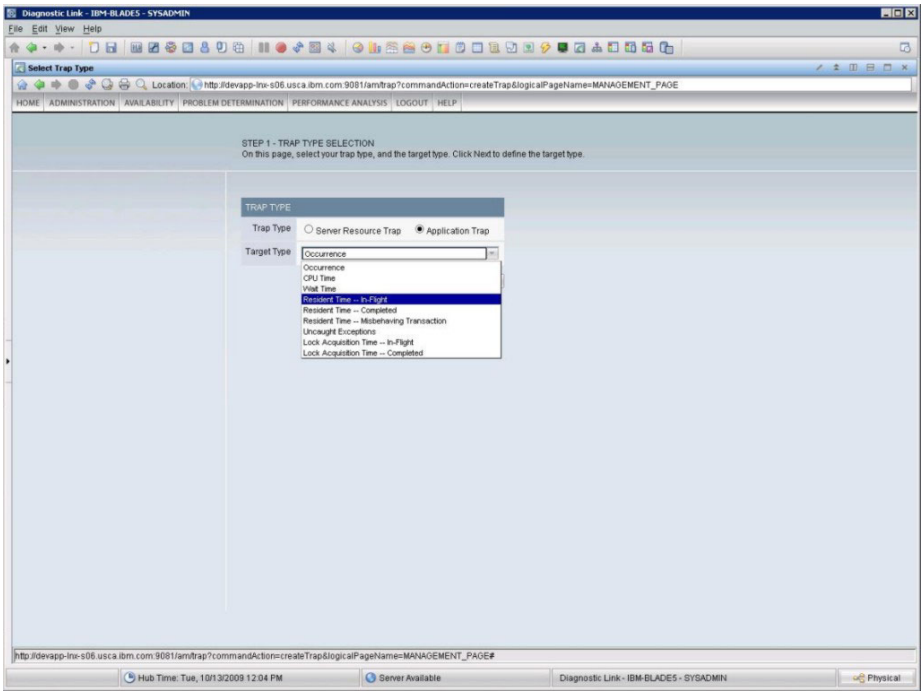


5. To further investigate the problem, Jim must open MSVE. Jim clicks the **Diagnostic Recent Completed Requests** link to open the MSVE Server Activity – Recent Requests page. Only requests that contain the URI information from

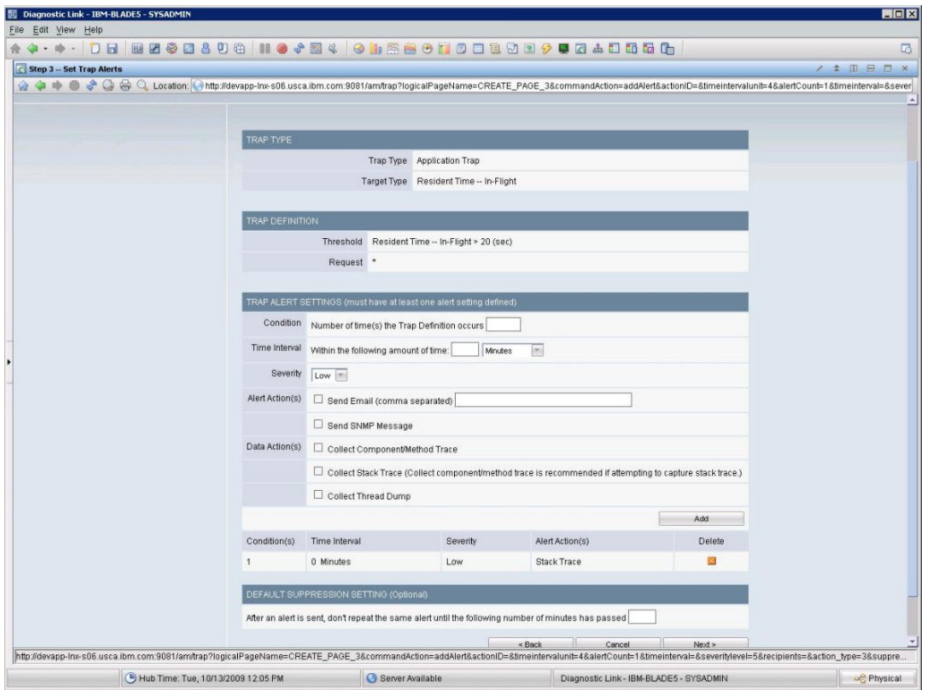
the Request Analysis workspace are displayed. Jim notices that there are a number of client requests with high response times.



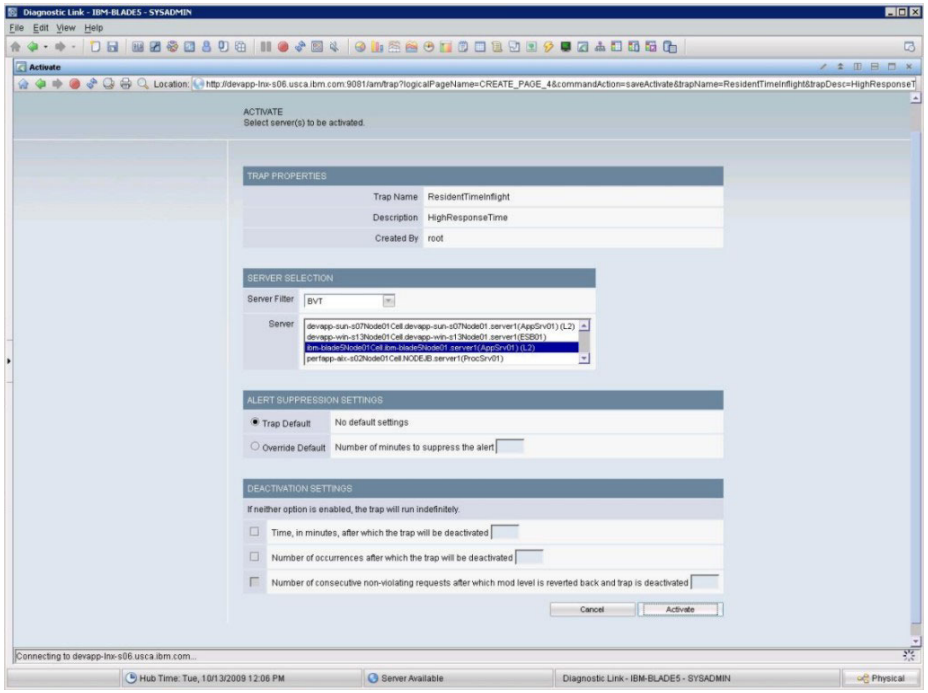
6. Jim decides to further analyze the transactions by setting a Resident Time – In-Flight trap. This trap activates the moment an in-flight request takes longer than a specified amount of time (the minimum is 15 seconds). To set up this trap, Jim must complete the following steps:
   a. Select the trap type.



   b. Set the trap alerts.

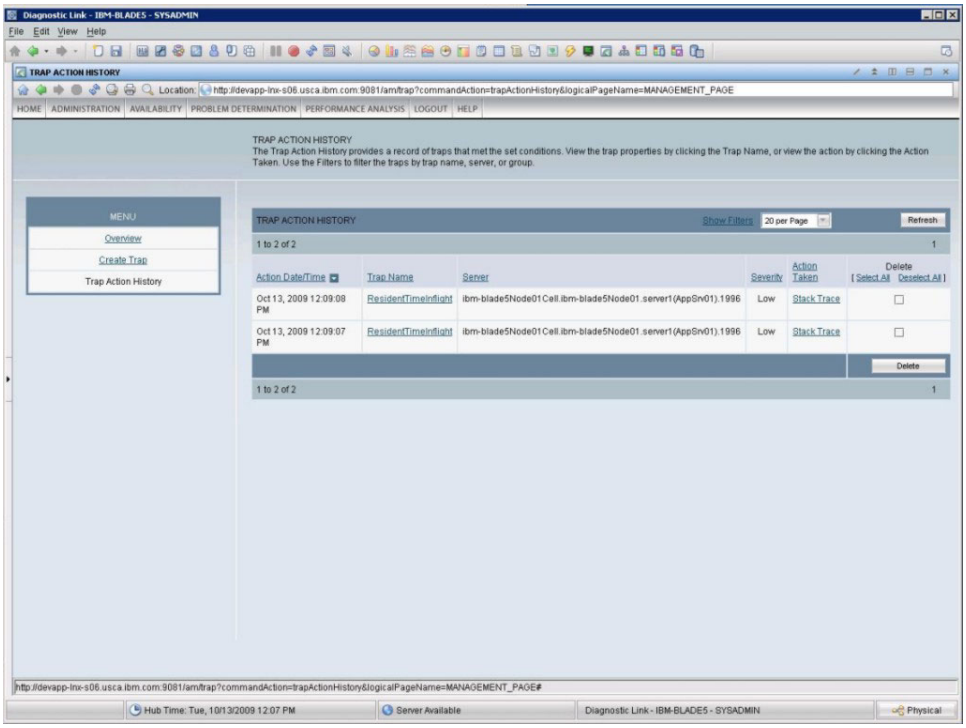


c. Activate the trap.

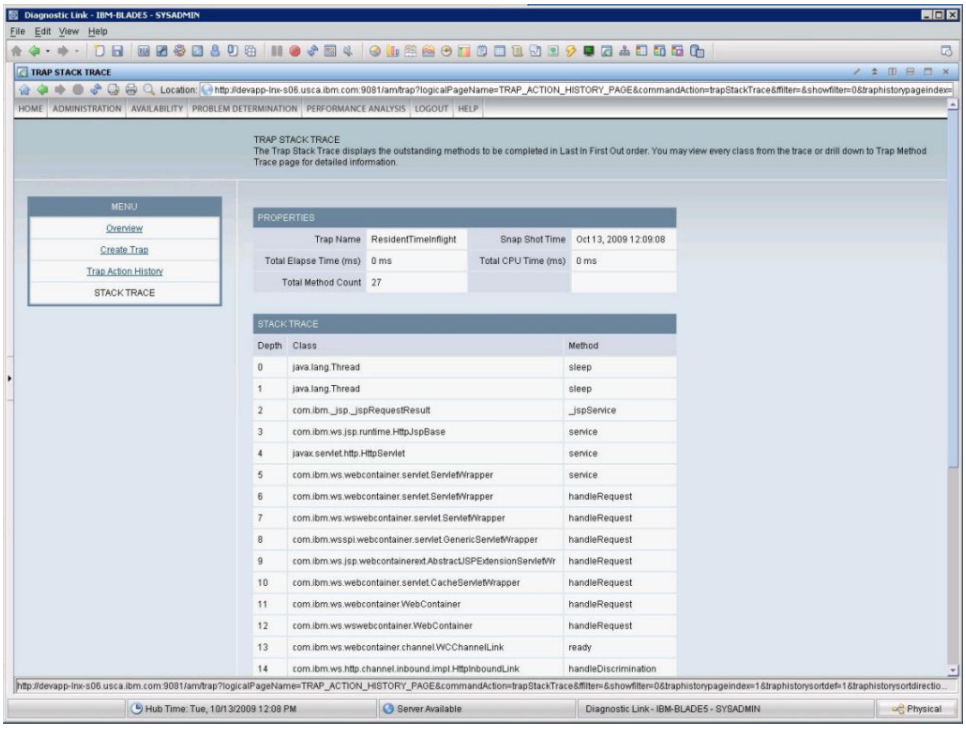


d. Jim then sets an action type of Stack Trace and waits for a problem request to trigger the trap.

7. After a while, the problem request triggers the trap.



8. This trap also produces a stack trace.

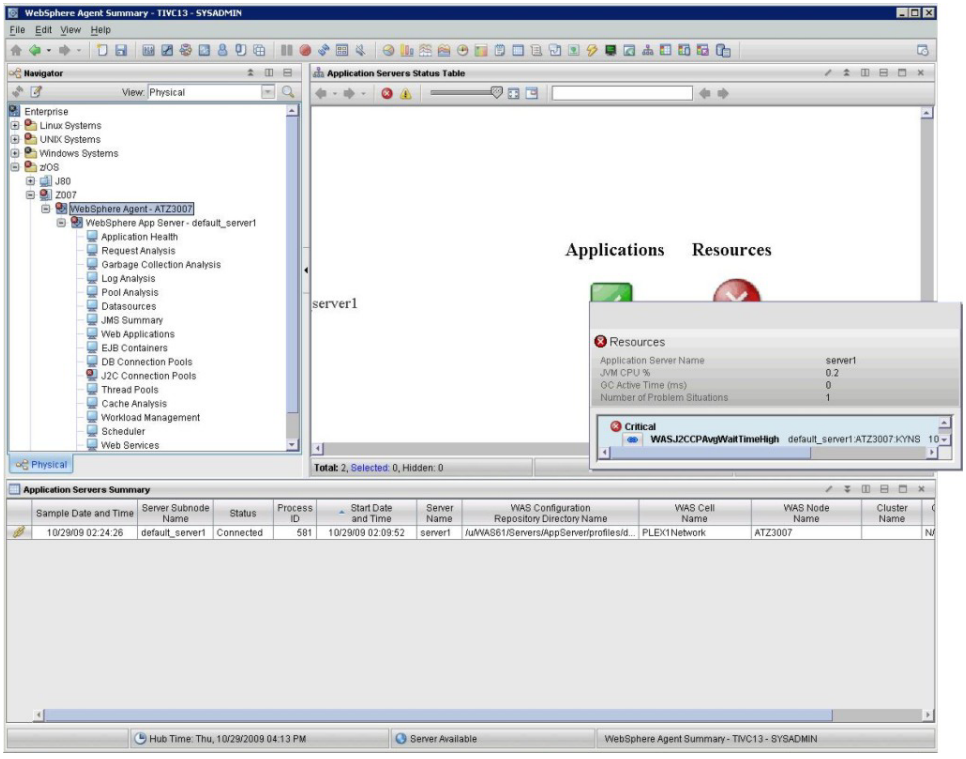


Jim forwards the trouble ticket to Dave, the application developer. Dave works to resolve the problem. This action is outside the scope of ITCAM for Application Diagnostics.

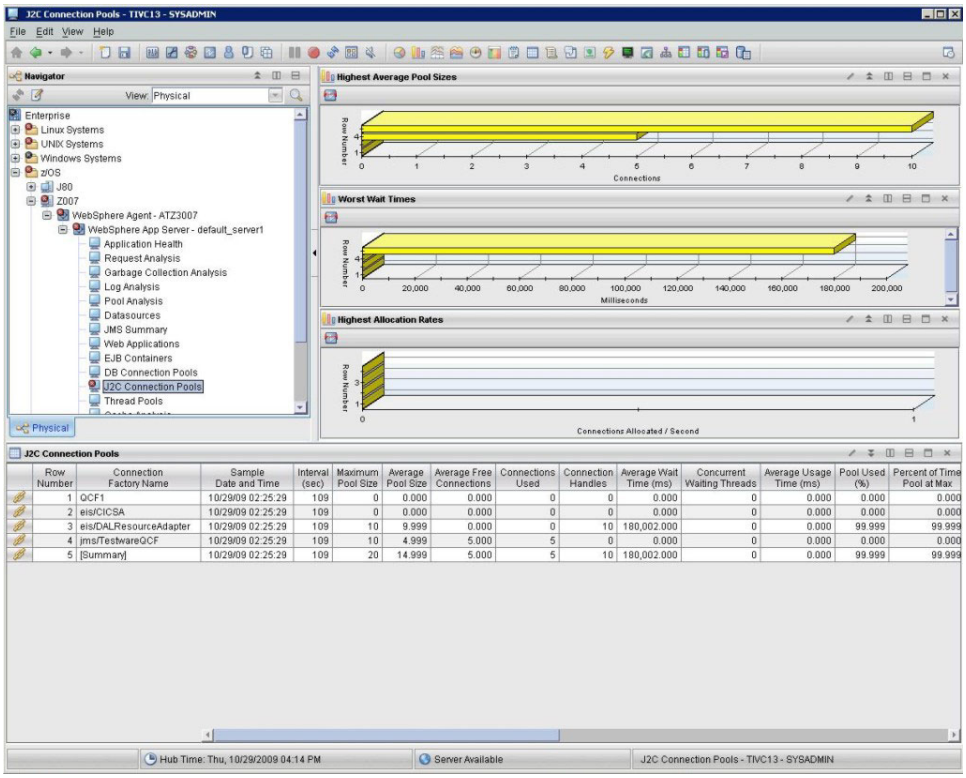
Scenario 6: Determining the cause of connection problems

Annette, the level 2 operator, receives notification that the WASJ2CCPAvgWaitTimeHigh situation triggered. This error is critical. This situation indicates that the average wait time until a connection is granted is longer than 2 seconds.

1. Annette navigates to the J2C Connections Pools workspace. This workspace reports information about resource adapters and connectors that adhere to J2EE Connector Architecture (J2C). J2C is the WebSphere Application Server implementation of the Java EE Connector Architecture (JCA). Data counters for this category contain usage information about the J2C connection pools that enable enterprise beans to connect to, and interact with, Enterprise Information Systems.



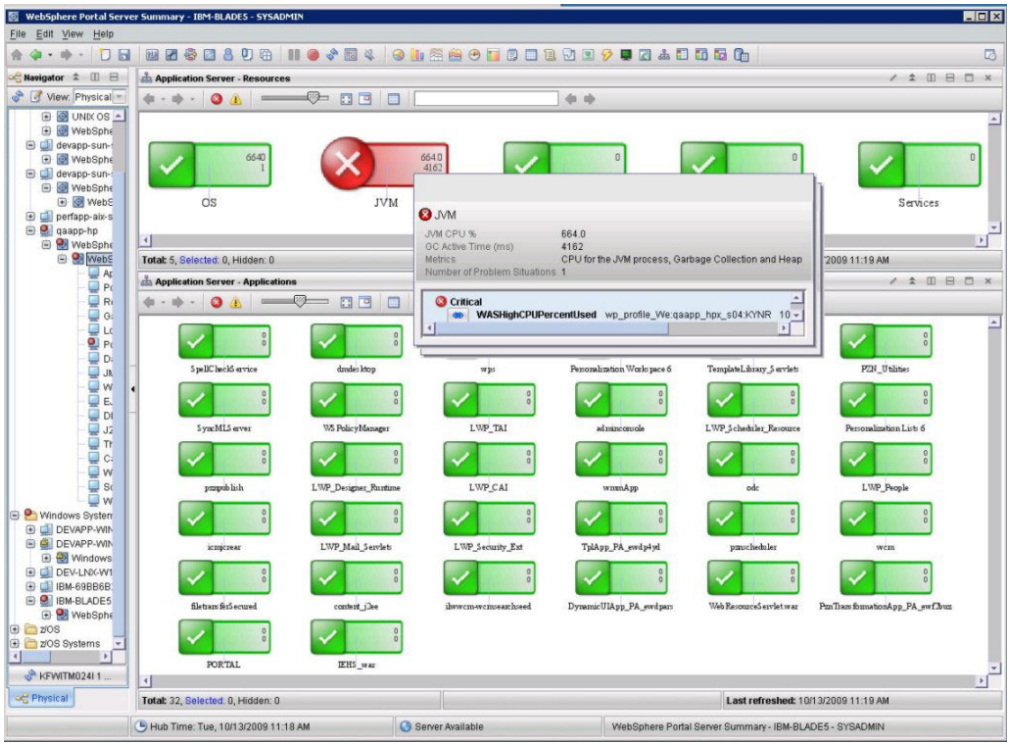
2. In this workspace, Annette observes in the Worst Wait Times view that some wait times for connections are above 2 seconds. The Highest Average Pool Sizes bar chart shows the largest average number of managed connections for each J2C connection pool. Typically, a connection takes no longer than 2 seconds.



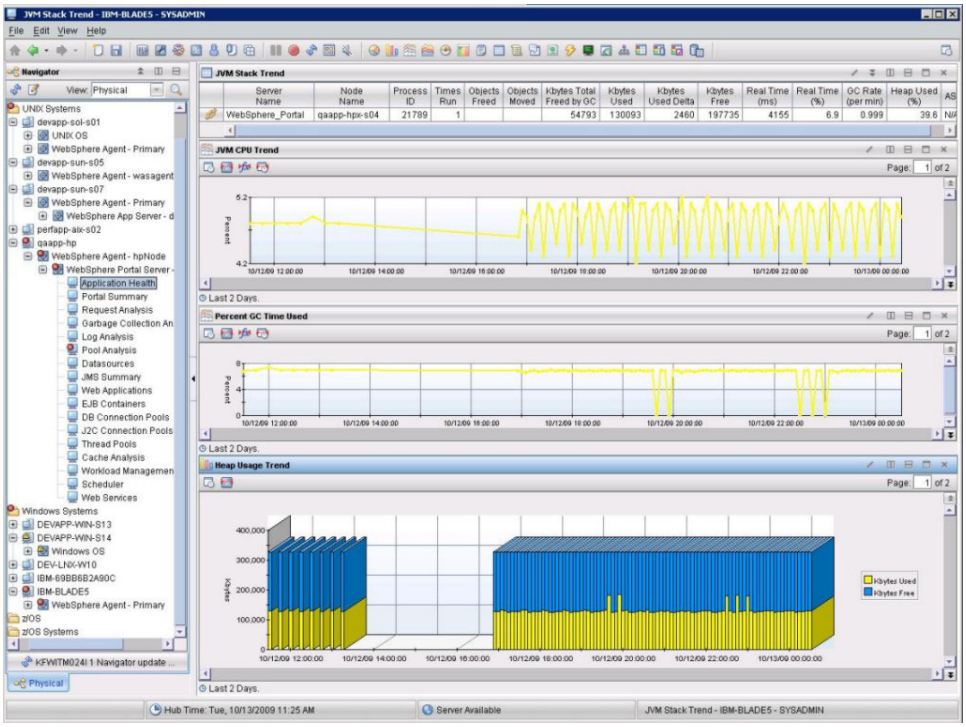
3. Annette uses an external ticketing tool to forward the ticket with details to Jim, the Middleware/Application Support SME.
4. Jim navigates to the J2C Connections Pools workspace and compares the average pools size with the maximum pool size to establish the ideal maximum value. Jim establishes that the connection pool size must be adjusted. The procedure for making this adjustment is outside the scope of ITCAM for Application Diagnostics.

Scenario 7: Determining if the Garbage Collection policy needs to be adjusted

Annette, the level 2 operator, is monitoring the Tivoli Enterprise Portal. Annette notices a critical symbol on the **JVM** icon in the Application Server Summary workspace. The flyover for **JVM** icons shows a high metric for JVM CPU% and GC Active Time (ms).



1. Annette double-clicks the **JVM** icon. The JVM Stack Trend workspace is displayed. The Percent GC Time Used view displays a high value. The heap usage trend is also high.



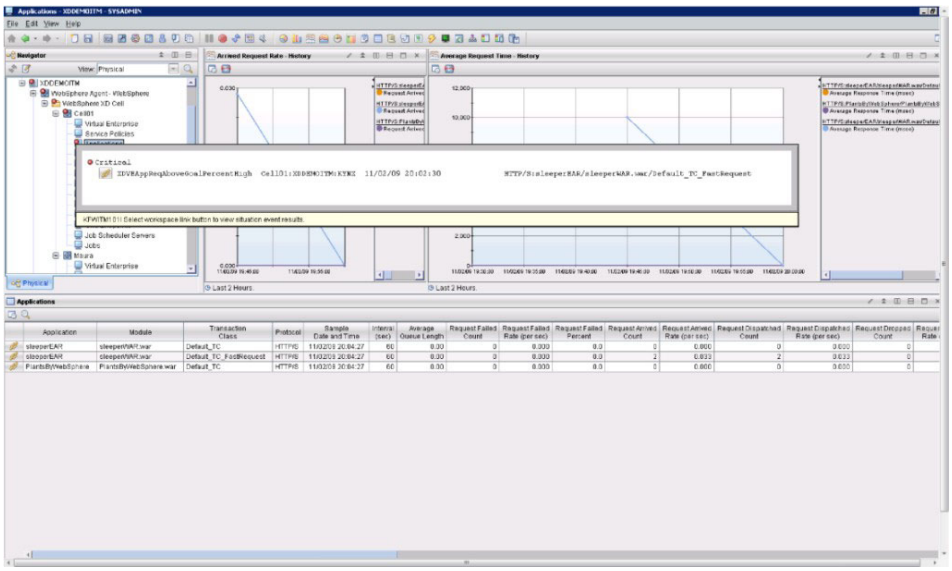
2. Annette uses an external ticketing tool to forward the ticket with details to Jim, the Middleware/Application Support SME.

3. Jim reviews the information and determines that the heap size parameters in the JVM are not set correctly. This incorrect setting affects application performance. Jim sets the appropriate GC policy.

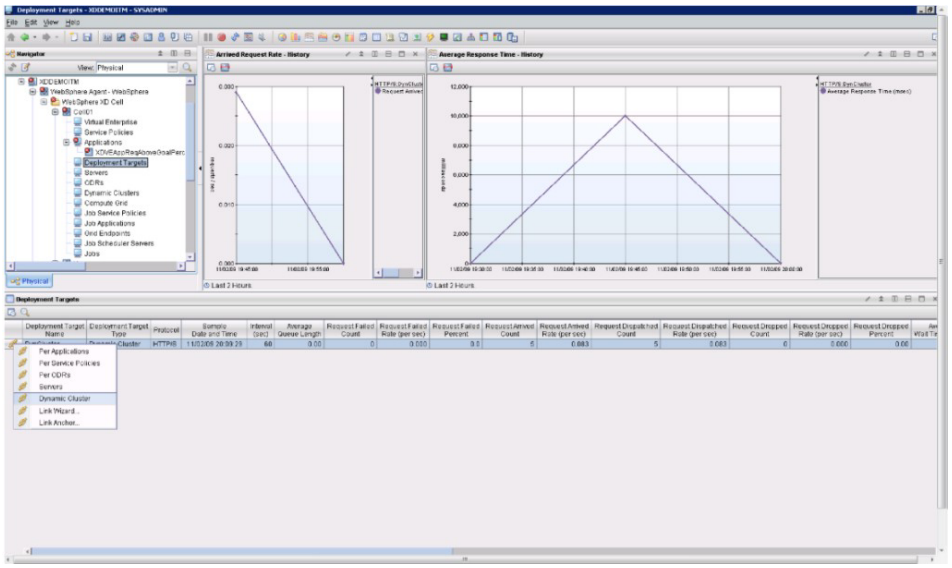
Scenario 8: Troubleshooting application response time in an XD cell

Due to external resource, the response time for one of applications in the XD cell degrades below the service policy goal. Annette, the level 2 operator, receives an email indicating that the Application Requests Above Goal situation triggered. This situation triggers when the rate of requests above goal is greater than 0.5%.

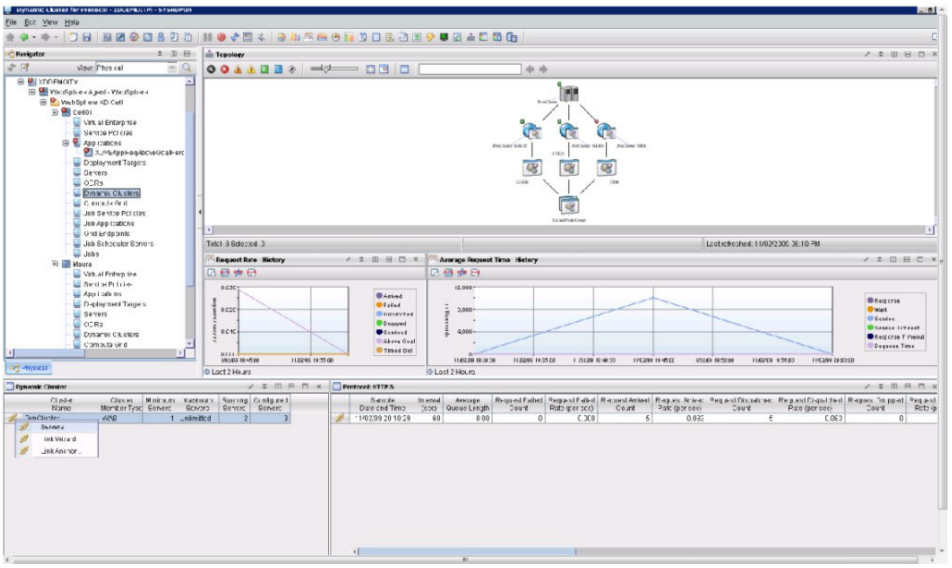
1. In the Tivoli Enterprise Portal navigation tree, Annette follows the workspace link for the triggered situation. The XDVEAppReqAboveGoalPercentHigh Event workspace is displayed.
2. Annette navigates by a link from the XDVEAppReqAboveGoalPercentHigh Event workspace to the Application workspace.



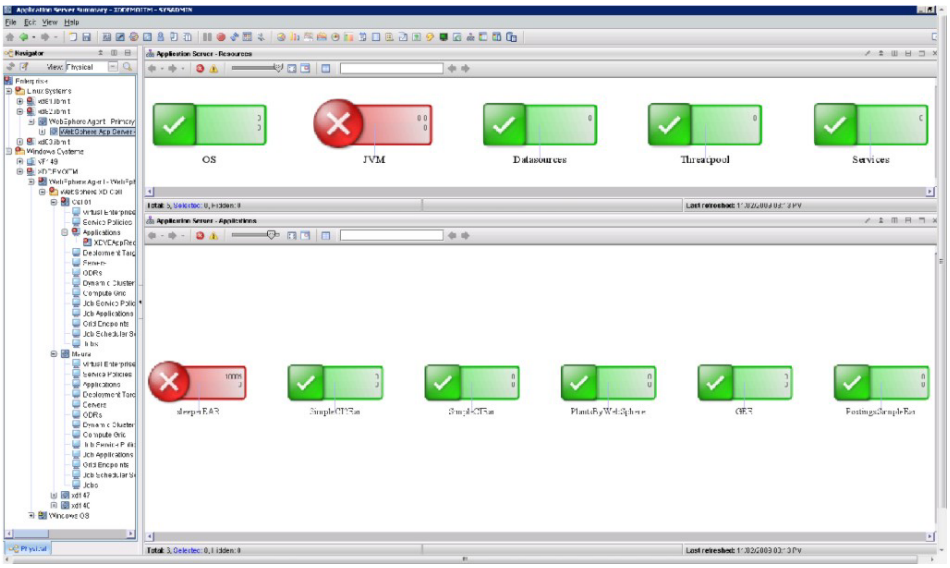
3. The Application workspace displays On Demand Router Statistics (ODR) for the selected application module, transaction, class, and protocol summarized over all ODRs in the cell.
4. From the Application workspace, Annette observes that the average overall response time is 20 seconds and the average server service time is also 20 seconds. This indicates that a problem is occurring with the handling requests by this application.
5. To see the deployment targets hosting the application and to ascertain which of the deployment targets is contributing to the slow response time, Annette navigates to the Deployment Targets workspace using the Per Deployment Targets link.



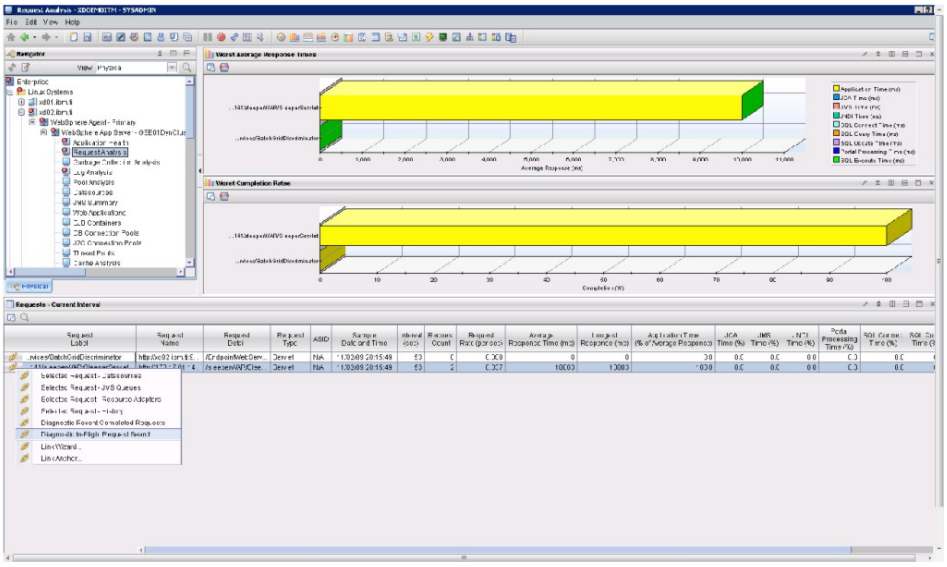
6. The Deployment Targets workspace displays a single deployment target for this application: DynCluster. Annette navigates to this dynamic cluster using the link and then to servers belonging to the dynamic cluster.
7. The servers for the dynamic cluster performance are displayed. From this view, Annette observes that both servers in the dynamic cluster have similar loading. Annette navigates to each server and observes that both have requests higher than the set goal of 10 seconds.



8. Annette navigates to Server Diagnostic for one of the servers using the link, which opens the data collector workspace for that server. Annette observes in the workspace that the sleeperEAR application is displaying a red critical icon.

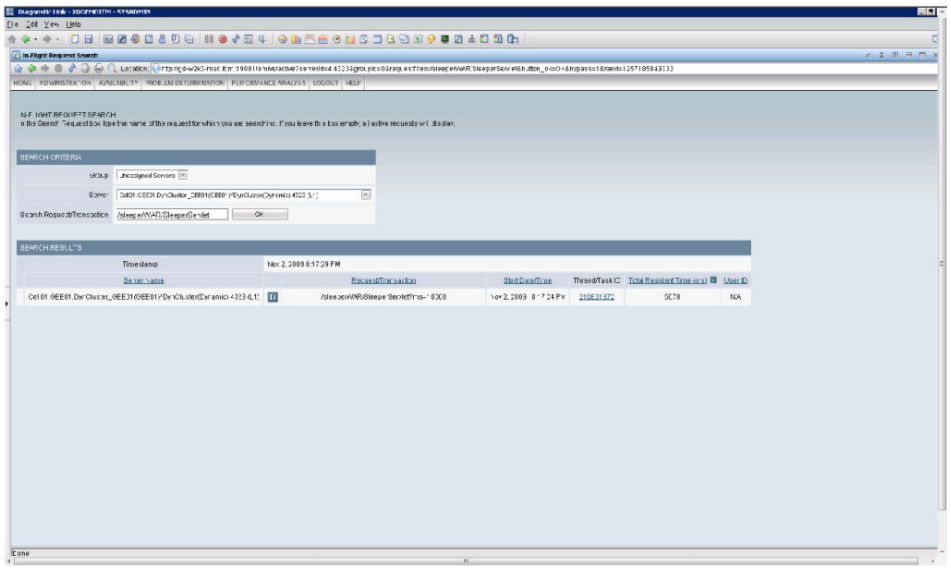


9. Annette double-clicks the application to open the Request Analysis workspace.



10. From the Request Analysis workspace, Annette navigates to the MSVE using the Diagnostic In-Flight Request Search link. From the MSVE, Annette can navigate to the thread for the selected request and view the request call stack.

    Important: This step is only available if you have ITCAM for Application Diagnostics, which includes the Managing Server.

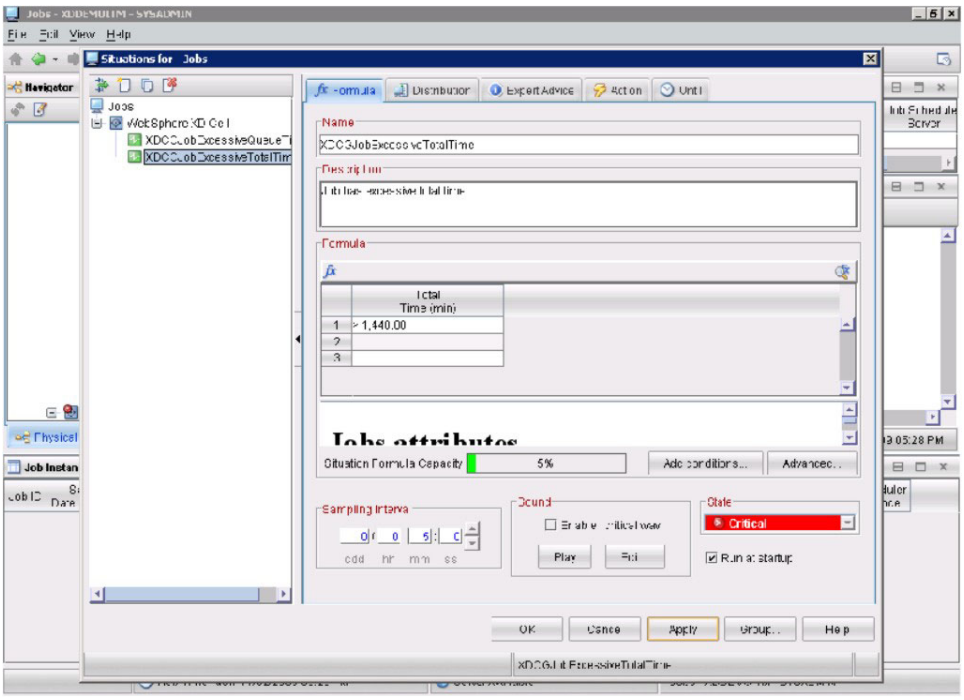


11. Annette uses an external ticketing tool to forward the ticket with details to Jim, the Middleware/Application Support SME.
12. Jim reviews the information and determines that the application waits for response from an external system. Jim researches and resolves the external system issue and restores the application response time before it affected most of the users.

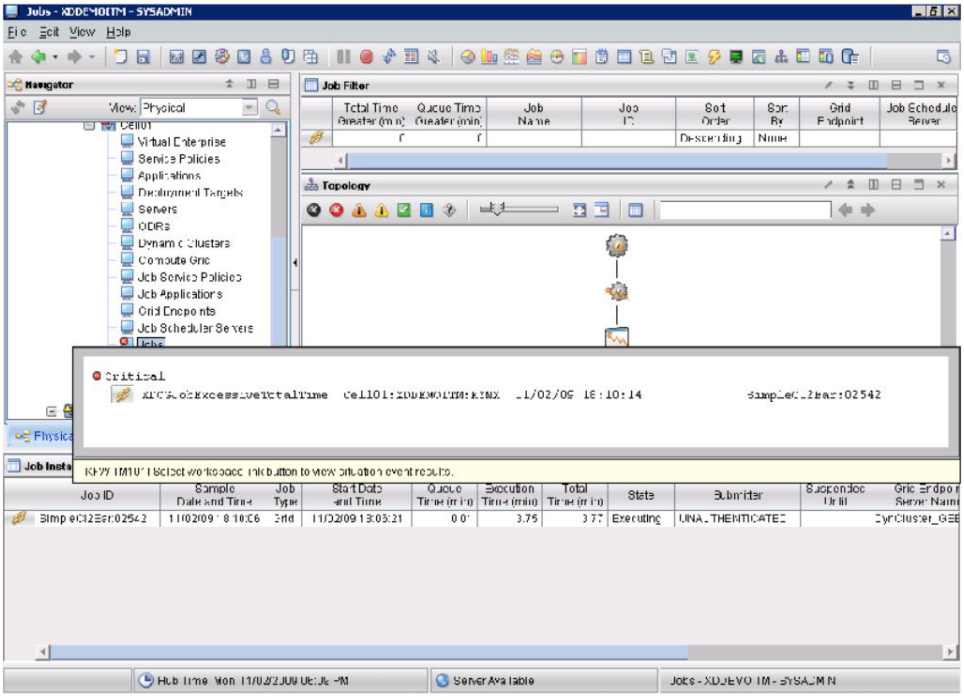
Scenario 9: Ensuring that jobs processed by Compute Grid do not execute for longer than one hour

Annette, the level 2 operator, must ensure that jobs processed by Compute Grid do not run for a longer period of time than one hour. If a job executes for longer, then Annette must capture the job information and forward it to the Middleware/Application Support SME.

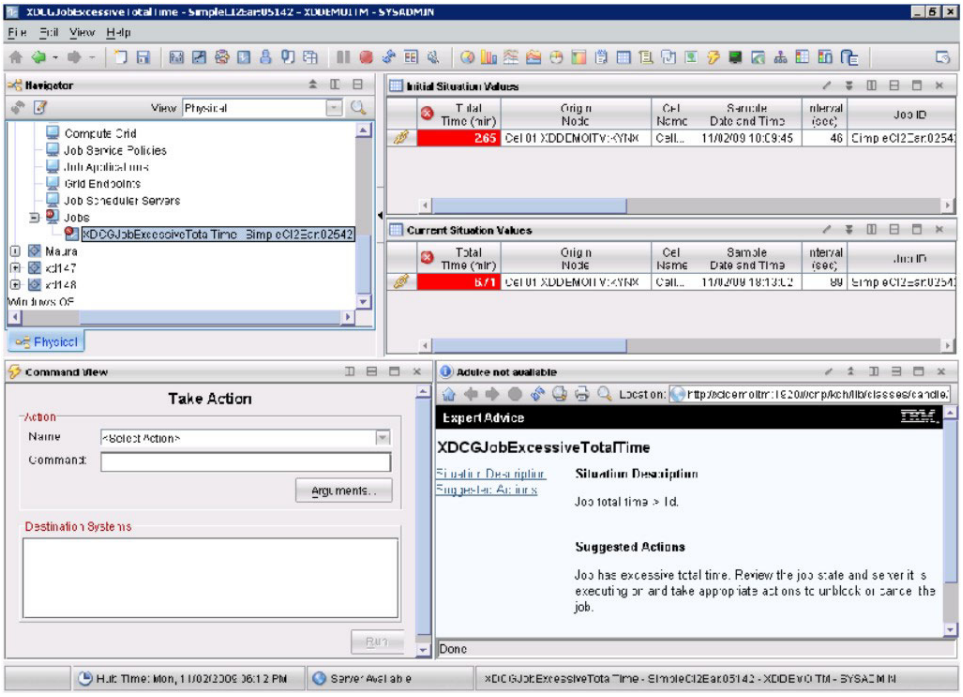
1. Middleware/Application Support SME configure jobs as custom requests via data collector configuration files.
2. Annette edits the predefined XDCGJobExcessiveTotalTime situation to trigger if the total job time is greater than 60 minutes.



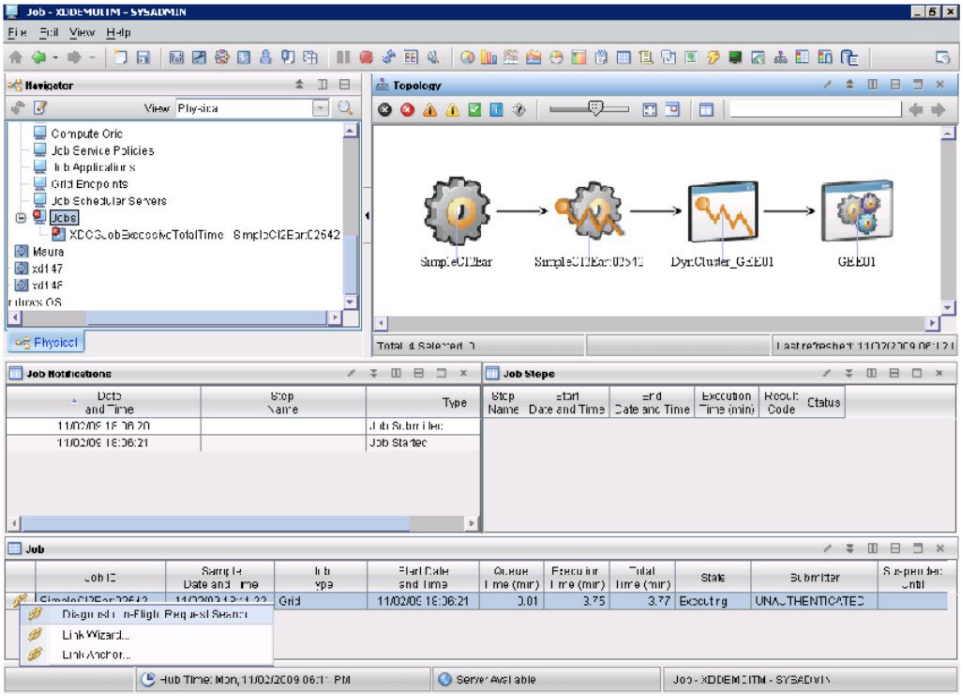
3. At some point, Annette observes that the situation triggered.



4. Annette clicks the event link and the Event workspace is displayed.



5. Annette opens the Job workspace by using the link from the Event workspace.



6. The Job workspace shows all the details about the job. From the job notifications, Annette observes that the job is running for a long time. She clicks the **Diagnostic in-Flight Request Search** link to open the MSVE in-flight workspace; from here, she clicks the link to see the request call stack.
7. Annette captures the call stack and provides to Jim, the Middleware/ Application Support SME.
8. Jim determines that job is waiting for database lock and resolves it appropriately.

Part 2. Part 2: Using the agents

© Copyright IBM Corp. 2006, 2015

Chapter 3. ITCAM for Application Diagnostics - Managing Server Visualization Engine

The Managing Server Visualization Engine (MSVE) user interface provides users with management and monitoring functions for application servers.

In addition, MSVE also provides a diagnostic function. Here are some of the diagnostic activities you can complete using the MSVE:

- Detect transactions failing
- Detect memory leaks
- Examine detailed method traces, which help to detect application code hotspots
- Generate reports to analyze historical information, such as application performance and OS performance

Using IBM Support Assistant to diagnose problems

About this task

If you experience a problem with ITCAM for Application Diagnostics and the problem was escalated to the Support team, you might be asked to use IBM Support Assistant (ISA) to create a TAR file to send to the Support team. The information in this TAR file helps the Support team to diagnose and fix your problem.

To download ISA:

Procedure

1. Go to the IBM Support Assistant web page http://www-306.ibm.com/software/support/isa/
2. If you do not have a universal IBM user ID, you must click **Register** and enter the required information. Upon completion, you can return to this page and sign in.
3. After signing in, follow the instructions to download the version of IBM Support Assistant that you want to use.

Results

To install ISA:

1. Go to the directory on your hard disk where you saved the ISA .zip file.
2. Extract the files in the ISA installation directory.
3. Double-click the `setupwin32.exe` file.
4. Follow the installation instructions to install ISA.
5. Open the ISA program.

To install the ISA plug-in:

1. Return to the web page http://www-306.ibm.com/software/support/isa/ and download the ISA plug-in to the ISA installation directory.
2. Go to the ISA installation directory where you saved the ISA plug-in .zip file.
3. Extract the files in this directory.

© Copyright IBM Corp. 2006, 2015

4. Double-click the setup file.
5. Follow the installation instructions to install the ISA plug-in.

To launch ISA:

1. From the ITCAM VE list, click **Help > Launch ISA**.
2. On the Main menu, click **Tools**.
3. On the left side under **Products**, click the link for IBM Tivoli Composite Application Manager.
4. Enter the managing server installation directory and the PMR number the Support team assigned to your problem. Click **OK**.
5. A script is called that creates a TAR file for use by the Support team. A message is displayed indicating ISA successfully created the TAR file. The TAR file is saved to the managing server installation directory with the following name: *PMR number.current timestamp*.tar. Send this file in an email to the Support team.

To collect problem determination files:

1. Open the ISA Workbench.
2. Click **Collect and Send Data**.
3. In the **Collector Selection** window, expand **IBM Tivoli Composite Application Manager for Application Diagnostics 7.1** and select **ITCAM for Application Diagnostics Managing Server Problem**. Click **Add>>** to add the collector to the **Collector Queue** list.
4. Click **Collect All** to run collecting.
5. Enter the managing server installation directory and the PMR number that the Support team assigned to your problem. Click **OK**
6. Doing this calls a script that creates a PMRnumber.current_timestamp.tar.gz file in the managing server installation directory for use by the Support team. You can send the file to this Support using ISA Workbench or you can send this file by email to the Support team.

Access the Managing Server Visualization Engine from Tivoli Enterprise Portal

You can access the MSVE from links in ITCAM Agent for WebSphere Applications workspaces.

When you access the MSVE in this way, the MSVE displays in a browser view inside a workspace. The Tivoli Enterprise Portal navigation tree is automatically hidden in the workspace. To show or hide the Tivoli Enterprise Portal navigation tree, click the small black arrow on the left side of the window.

The following table displays a list of Tivoli Enterprise Portal workspaces that have links to the MSVE.

Table 3. Tivoli Enterprise Portal workspaces that have links to MSVE

Workspace: Table View Name	Number of links to MSVE	Link Name	Link Target Pages MSVE	Pre-populated information in the link page to MSVE
WebSphere Agent Summary Status > Application Servers **WebSphere Agent Summary Status > Application Servers** **WebSphere Agent Configuration> Application Servers**	2	• Diagnostic Server Activity Display • Diagnostic In-Flight Request Search	• Server Activity Display – Active Requests • In-Flight Request Search	The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace.
OS Stack > Current® OS Stack Summary	3	• Diagnostic Server Activity Display • Diagnostic In-Flight Request Search • <platform OS> <platform> is one of the following operating systems: Linux, UNIX, Windows, or z/OS®	• Server Activity Display – Active Requests • In-Flight Request Search • Using the dynamic workspace link to the corresponding OS agent workspace. For z/OS, the link is to OMEGAMON XE for z/OS.	The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace.
JVM Stack Trend >JVM Stack Trend	1	Diagnostic Memory Leak	Memory Leak Analysis	The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace.

Table 3. Tivoli Enterprise Portal workspaces that have links to MSVE (continued)

Workspace: Table View Name	Number of links to MSVE	Link Name	Link Target Pages MSVE	Pre-populated information in the link page to MSVE
Request Analysis > Requests – Current Interval	3	• Diagnostic Recent Completed Requests • Diagnostic In-Flight Request Search • Diagnostic SMF Data (z/OS only)	• Server Activity Display – Recent Requests • In-Flight Requests • SMF Data (for z/OS data collectors only)	• The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace. • Content in Request Detail column of Requests table view in Tivoli Enterprise Portal is pre-populated in the following fields: – Recent Requests: Client Request – In-Flight Request Search: Search Request/ Transaction field
Garbage Collection Analysis >Garbage Collection Analysis	1	Diagnostic Memory Leak	Memory Leak Analysis	The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace.
Thread Pools >Thread Pools	1	Diagnostic JVM Thread Display	JVM Thread Display	The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace.

Table 3. Tivoli Enterprise Portal workspaces that have links to MSVE (continued)

Workspace: Table View Name	Number of links to MSVE	Link Name	Link Target Pages MSVE	Pre-populated information in the link page to MSVE
Datasources > Datasources – Current Interval **Web Applications >Web Applications** **EJB Containers >EJB Containers** **JMS Summary >JMS Summary – Current Interval** **DB Connection Pools > DB Connection Pools** **J2C Connection Pools > J2C Connection Pools**	1	Diagnostic Server Activity Display	Server Activity Display – Active Requests	The Group Server menu is pre-populated based on data collector information from Tivoli Enterprise workspace.

The **Server Group** feature that displays at the top of these pages applies to the MSVE. When you access any MSVE page from the Tivoli Enterprise Portal, the information in the **Groups** and **Servers** fields is automatically populated with the data collector associated with the link and workspace you selected in the Tivoli Enterprise Portal.

Groups are a set of application servers which have similar functionality. All configured data collectors are automatically assigned to the **Unassigned Servers Group**. The relationship between Server Group and data collector is many to many. A data collector can belong to one or more server groups. A server group can have one or more data collectors. You can add data collectors to groups using the **Server Management** functionality in the MSVE. For more information about adding data collectors to Server Groups, see the Composite Application Manager Help in the MSVE interface.

The **Server Activity Display** section has three tabs.

- **Active Requests**
  Provides real-time request or transaction data for an application server at the time the page displays.
- **Recent Requests**
  Displays the last 100 or less completed request or transaction data for an application server.
-

Lock Contentions
Displays requests that are hanging because they are waiting on a lock. The data shows data that is currently locked and the item that is waiting to be locked.

The **Active Requests** tab and the **Recent Requests** tab have a toolbox icon  . You can click this toolbox icon to access direct links to the following features:

- JVM Thread Display
- System Resources
- Monitoring On Demand
- Data Collector Profiles
- Trap and Alert Management

You can use this information to analyze the details and identify the areas where the issues occur. To return to the Tivoli Enterprise Portal interface at any time, click the back button on your web browser.

For more information, see:

- "Server Activity Display - recent requests" on page 119
- "Server Activity Display - active requests" on page 117
- "Server Activity Display - lock contentions" on page 121
- "JVM thread display" on page 139
- "System resources" on page 101
- "Monitoring on Demand (TM)" on page 77
- "Data Collector Profiles" on page 65
- "Trap and alert management" on page 142

Account management

Control access to features and servers.

You can use account management to control the access of users to features and servers. Use roles to restrict access to features, and use server groups to grant access to servers.

User Scenarios

Scenario 1: Granting members of Team XYZ access to ITCAM for Application Diagnostics

Team XYZ asked for access to ITCAM for Application Diagnostics, but needs access only to features that use historical data. Because the existing roles provide access to features that use both real time and historical data, create a 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 recently joined your company. John must use ITCAM for Application Diagnostics for to monitor QA systems. As the ITCAM for Application Diagnostics administrator, you create John's account with access

granted to QA server groups but not Production server groups. You also restrict Johns access to features by assigning the Operator role to his account.

User profiles

The User Profiles page shows a table of all user accounts in the visualization engine. From this page, you can create, modify, and delete user accounts.

To view the User Profiles page, from the top navigation, click **Administration > Account Management > User Profiles**.

For every account, the table shows the account name, given name, family name, and a **Delete** button.

To sort by any column, click its heading.

To modify the information for an account (including access rights), click the account name. See "Modifying a user account" on page 48.

To delete a user account, click the **Delete** button in its row. See "Deleting a user account" on page 49.

To create a user account, in the left navigation pane, click **Create User Account**. See "Creating a user account."

Creating a user account

About this task

Add new user accounts to the application monitor on the Create User Account page. Limit the rights of your user accounts to the groups of servers that you select. A valid WebSphere Global Security user name is required to create an account. The user name can be different from the operating system user ID, but it must be 3 - 255 alphabetic characters. To enable single sign-on, configure the managing server, and also configure the WebSphere application server that the MSVE runs on. Then, add every user who needs to access single sign-on. For more information about single sign-on, see information about **Setting Up Single Sign on for Tivoli Enterprise Portal Users** in the *ITCAM for Application Diagnostics Managing Server Installation and Customization Guide*.

To create a user account:

Procedure

1. From the top navigation, click **Administration > Account Management > User Profiles**. The User Profiles page opens.
2. In the left navigation pane, click **Create User Account**. The Create User Account page opens.
3. Enter the given name.
4. Enter the family name.
5. Enter the user name.
6. Enter the OS/LDAP user name (required). ITCAM uses WebSphere Global Security system for user authentication, therefore a valid WebSphere Global Security user name is required to create an account.
7. Select the role that you want to assign to the user account from the list.
8. Select **Active** or **Suspended** for the account status. Unless the user account status has been marked **active**, it is not ready for use.

9. Optional: Enter the e-mail address of the user.
10. Optional: Enter remarks in the available fields.
11. To save the user account setup, click **Save**.

Results

To grant group access rights:

1. Click to select the group name in the All Groups box.
2. Click **Add** to grant the user account rights to the selected groups. The group name is displayed in the Granted box.

   To select multiple groups, hold down the Ctrl key during your selection.

To remove group access rights:

1. Click to select the group name in the Granted box.
2. Click **Remove** to remove the user account rights from the selected groups. The group name is no longer displayed in the Granted box.
3. To select multiple groups, hold down the Ctrl key during your selection.

   Related topics

   Assigning a role

   Creating a role

Modifying a user account

About this task

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

To modify a user account:

Procedure

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 that you want to modify. The Modify User Account page opens.
3. Select the field that you want to edit, and enter the new information.
4. After entering your changes, click **Save**. You might want to suspend the user accounts when the operators are on leave. When they return, click **Active** to turn their user accounts back on.

Results

To grant group access rights:

1. Click to select the group name in the All Groups box.
2. Click **Add** to grant the user account rights to the selected groups. The group name is displayed in the Granted box.

To remove group access rights:

1. Click to select the group name in the Granted box.
2. Click **Remove** to remove the user account rights from the selected groups. The group name is no longer displayed inthe Granted box.

   Related topics

Assigning a role

Creating a group

Deleting a user account

About this task

Manage your accounts by keeping them up-to-date. Delete existing user accounts from the application monitor on the User Profiles page.

To delete a user account:

Procedure

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 the application monitor. 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. If you select **OK**, the system deletes the user account and the User Profiles page no longer displays the deleted account.
4. To sort by heading, click the heading that you want to sort. Only underlined headings can be sorted. The page displays the results sorted by the selected heading.

Related topics

Creating a role

Creating a user account

Deleting a role

Role configuration

The Role configuration page shows a table of permissions for security roles defined in the visualization engine. From this page, you can create, modify, and delete security roles.

To view the Role Configuration page, from the top navigation, click **Administration > Account Management > Role Configuration**.

To control user account access to the product functions, each user account is assigned a *security role*. The role grants access to specific product functions. A user account can have only one role; the same role can be assigned to many accounts.

The rows in the table represent product functions; columns represent security roles. Every cell of the table shows whether function (row) is allowed for the role (column).

The Administrator, Operator, and User roles are predefined and cannot be changed; you can change, add, and delete other roles.

To change permissions for a role, check or clear the boxes in its column. To save changes, click **Save**; to undo changes before they are saved, click **Reset**. (You might have to scroll the page down to reach these buttons). See “Modifying a role” on page 51.

To delete a role, click **X** next to its name. See “Deleting a role” on page 52.

To create a role with blank permissions, in the left navigation pane, click **Create Role**. See "Creating a role."

To create a role with permissions copied from another role, in the left navigation pane, click **Duplicate Role**. See "Duplicating a role" on page 51.

Creating a role

About this task

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 for your environment.

To open the Create Role page:

Procedure

1. From the top navigation, click **Administration > Account Management > Role Configuration**. The Role Configuration page opens.
2. On the left navigation pane, click **Create Role**. The Create Role page opens.
3. Type the name of the new role.
4. Click **OK**. The new role displays on the Role Configuration page.
5. Click to select the privileges user accounts can access in the application monitor.
6. Click **Save**.
7. Click **Reset** to revert to the pre-modified settings.

Related topics

Assigning a role

Creating a user account

Assigning a role

About this task

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

To assign a role:

Procedure

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 list, select the role to assign to the user account.
4. Click **Save**.

Related topics

Creating a role

Deleting a role

Modifying a role

Modifying a role

About this task

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:

Procedure

1. From the top navigation, click **Administration > Account Management > Role Configuration**. The Role Configuration page opens.
2. Click to check and clear the permissions that you want to assign this role. Changing the custom role privileges in the user accounts grants access in the application monitor.
3. Click **Save**.
4. Click **Reset** to revert to the pre-modified settings.

Related topics

Assigning a role

Creating a role

Duplicating a role

About this task

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

To duplicate a role:

Procedure

1. From the top navigation, click **Administration > Account Management > Role Configuration**. The Role Configuration page opens.
2. On the left navigation pane, click **Duplicate Role**. The Duplicate Role page opens.
3. Select a role name for the duplicated role from the Role Name list.
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 privileges user accounts can access in the Application Monitor.
7. Click **Save**.
8. Click **Reset** to revert to the pre-modified settings. The duplicated role does not have any users since its user-to-role relationship is not duplicated.

Related topics

Assigning a role

Creating a user account

Modifying a role

Deleting a role

About this task

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

To delete a role not assigned to a user account:

Procedure

1. From the top navigation, click **Administration > Account Management > Role Configuration**. The Role Configuration page opens.
2. Click the **X** next to the role that you want to delete.
3. At the confirmation box, click **OK**.

Results

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 **X** next to the role that 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 is displayed. Because the system assigned the role to a user account, you must change the role of the user account on the Update Role page.
4. Click 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 list.
6. Click **Save**. The system displays the Role Configuration page without the deleted role.

Related topics

Creating a role

Creating a user account

Deleting a user account

Server management

Manage your servers with the server group management page.

Add and delete server groups, while associating groups with individual account. You can restrict user access to data and operations on a specific group of servers.

User Scenarios

Scenario 1: Separating server groups according to applications

As the ITCAM for Application Diagnostics 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. In the Quotes server group, you include only those servers that deal with quotes. Grant users access to the appropriate server group or groups.

Scenario 2: Grouping servers by authority structure

As the ITCAM for Application Diagnostics 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.

Server groups

The Server Groups page shows a table of all server groups. From this page, you can create, modify, and delete server groups.

Groups are a set of application servers which have similar functionality. All configured data collectors are in the **Unassigned Servers Group**. The relationship between Server Group and data collector is many to many. A data collector can belong to one or more server groups. A server group can have one or more data collectors. You can add data collectors to groups using the **Server Management** functionality in the MSVE. For more information about adding data collectors to Server Groups, see the Composite Application Manager Help in the MSVE interface.

To view the Server Groups page, from the top navigation, click **Administration > Server Management > Server Groups**.

Server groups are used for convenient grouping of information in the MSVE. Every group includes one or several servers. In a number of MSVE pages, you can view information by server group. Each user can be granted access to information for some server groups but not others. A server can be a member of several groups.

When you access pages from the Tivoli Enterprise Portal, the information in the **Groups** and **Servers** fields is automatically populated with the group and data collector associated with the link and workspace you selected in the Tivoli Enterprise Portal.

For every server group, the table shows the group name, description, and a **Delete** button.

To modify a server group (including list of member servers and access rights for users), click the group name. See “Modifying a group” on page 54.

To delete a server group, click the Delete button in its row. See “Deleting a group” on page 55.

To create a server group with default settings, no member servers and no user access rights, in the left navigation pane, click **Create Group**. See “Creating a group.”

To create a server group with settings, member servers and user access rights copied from another server group, in the left navigation pane, click **Duplicate Group**. See “Duplicating a group” on page 55.

Creating a group

About this task

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:

Procedure

1. From the top navigation, click **Administration > Server Management > Server Groups**. The Server Group Management page opens.
2. On the left navigation pane, click **Create Group**. The Create Group page opens.
3. Enter a unique group name in the text box.
4. Enter a description in the text box.
5. Enter the Server Group Response Time Thresholds.
6. Enter the Portal Response Time Thresholds. (If you have a portal server, configure the thresholds for portal.) Optional.
7. Click to select a baseline definition and fill out the information. Steps 5 through 7 are all default settings based on the settings on the System Properties page under Configuring the Enterprise Overview Display section.
8. Click to select the server name in the All Servers box.
9. Click **Add** to select the server for the group. The server name is displayed in the Servers in Group box. To select multiple servers, hold down the shift key during your selection. To add multiple servers non-continuously, Ctrl + click the servers for selection.
10. In the Servers In Group box, select the server that you want to remove and click **Remove** to delete the server from the group. The server name disappears from the Servers in Group box.
11. Select the user and click **Add** to grant users access to the group. The user name is displayed in the Granted Access box.
12. Click **Remove** to remove the access of the user to the group. The user name is no longer displayed in the Granted Access box.
13. Click **Save** to save the group settings.

Related topics

Configuring a data collector

Configuring the Enterprise Overview display

Creating a configuration

Modifying a group

About this task

Maintain your groups with the most updated information. The Modify Group page provides the functionality to modify your groups and grant users access to those groups.

To modify a group:

Procedure

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 that you want to modify. The Modify Group page opens populated with the selected group information.
3. Select the field that you want to edit and enter the new information.

4. Click **Save** to save the group settings. 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 notices the change.

   Related topics

   Configuring the Enterprise Overview display

   Creating a group

Deleting a group
About this task

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

To delete a group:

Procedure

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 that you want to delete from the application monitor.
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 application monitor deletes the group and the Server Group Management page no longer displays the deleted group. Before deleting a group from the application monitor database, delete all reports attached to that group in order to maintain data integrity. To delete each report, click the report link. The group will be deleted after all reports are deleted.

   Once a group is deleted, the records in the application monitor database that belong to the group via the server relationship are no longer accessible through the group. However, the records can still be accessed either via the server name or another group which contains the servers. You can also reassign the servers to other groups on the Modify Group page.

   Related topics

   Deleting a configuration

   Modifying a group

Duplicating a group
About this task

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

To duplicate a group:

Procedure

1. From the top navigation, click **Administration > Server Management > Server Groups**. The Server Group Management page opens.
2. On the left navigation pane, click **Duplicate Group**. The Duplicate Group page opens.
3. From the Group Name list, select the group name that you want to duplicate.
4. Enter a new name for the duplicated group.

5. Click **Save** to duplicate the group. The Duplicate Group link is not displayed when no group is in the system.

Related topics

Creating a group

Deleting a group

Modifying a group

Data Collector Configuration

In the Data Collector Configuration pages, you can configure data collectors, disable, enable, and unconfigure them, and also enable/disable TTAPI support.

To view the Data Collector Configuration page, from the top navigation, click **Administration > Server Management > Data Collector Configuration**.

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. The data collector communicates this information to the managing server, or to the Tivoli Enterprise Monitoring Agent, or to both.

To work with the managing server, a data collector must first be configured using its own configuration utility. After this, you must also configure the data collector through the visualization engine. To do this, you must apply one of the defined configurations to this data collector.

You might also need to unconfigure a data collector. When you have unconfigured a data collector, you can apply a different configuration. If you want to remove the data collector, you can unconfigure it and then uninstall it.

To view the configured data collectors, enable, disable, and unconfigure them, select **Configured Data Collectors** in the left navigation pane. See "Configured data collectors."

To view the unconfigured data collectors and configure them, select **Unconfigured Data Collectors** in the left navigation pane. See "Unconfigured data collectors" on page 57.

To view the existing data collector configurations and modify, duplicate, or delete them, select **Configuration Library** in the left navigation pane. See "Configuration Library" on page 57.

To create a data collector configuration, select **Create a Configuration** in the left navigation pane. See "Creating a configuration" on page 61.

Configured data collectors

The Configured Data Collectors page shows a table of all data collectors that are configured in the visualization engine. You can view their configuration, enable, disable, and unconfigure them.

For every configured data collector, the table shows:

- Admin Server name
- Application Server name
- Cluster name
- Configuration name

- Platform that is monitored (type of application server)
- Status change button (**Disable** if the data collector is enabled, or **Enable** if it is disabled)
- **Unconfigure** check box.
- Join Time - if a data collector is not available because the data collector connection to the managing server was unconfigured, then the value for Join Time is displayed as N/A.

To disable an enabled data collector, click **Disable**. See "Disabling a data collector" on page 60.

To enable a disabled data collector, click **Enable**. See "Enabling a data collector" on page 60.

To unconfigure a data collector, check the **Unconfigure** box and click **Apply** at the bottom of the table. See "Unconfiguring a data collector" on page 59.

To view the details of a data collector configuration, click the configuration name. The "Configuration Library" page opens, showing this configuration.

You can use the left navigation pane to view unconfigured data collectors, view existing data collector configurations, create a configuration, and enable TTAPI. See "Data Collector Configuration" on page 56.

Unconfigured data collectors

The Unconfigured Data Collectors page shows a table of all data collectors that communicate with the managing server but are not configured in the visualization engine. You can configure them.

For every unconfigured data collector, the table shows:

- Admin Server name
- Application Server name
- Cluster Name
- Platform that is monitored (type of application server)
- **Apply a configuration** check box

To configure a data collector, check the **Apply a configuration** box, select the configuration name in the pull-down control at the top of the table, and click **Apply** at the bottom of the table. See "Configuring a data collector" on page 58.

You can use the left navigation pane to view configured data collectors, view existing data collector configurations, create a configuration, and enable TTAPI. See "Data Collector Configuration" on page 56.

Configuration Library

The Configuration library page shows a table of all data collector configurations. You can apply configurations to data collectors, modify them, delete them, and create new configurations as copies of existing ones.

For every data collector configuration, the table shows:

- Configuration name.
- Class names that are to be excluded from monitoring.

- Class names that are to be included in monitoring, even if they would fit the exclude list.
- Associated Server names, that is, names of servers to which the configuration is presently applied
- **Modify**,**DuplicateApply** and **Delete** buttons.

To modify a configuration, click **Modify**. See "Modifying a configuration" on page 62.

To create a configuration as a copy of an existing one, click **Duplicate**. See "Duplicating a configuration" on page 63.

To apply a configuration to data collectors, click **Apply**. Note that you can configure a previously unconfigured data collector or change the configuration of an already configured data collector in this way. See "Applying a configuration" on page 62.

To delete a configuration, click **Delete**. See "Deleting a configuration" on page 63.

You can use the left navigation pane to view configured and unconfigured data collectors, create a configuration, and enable TTAPI. See "Data Collector Configuration" on page 56.

Enable TTAPI for JDBC

The Enable TTAPI for JDBC page shows tables of data collectors that support TTAPI for JDBC and do not currently have it enabled, and those that have TTAPI for JDBC enabled. You can enable and disable TTAPI for JDBC for these data collectors.

The **TTAPI JDBC disabled Data Collectors** table shows the data collectors that support TTAPI for JDBC and do not currently have it enabled. For every data collector, the server name and the **Enable** check box are shown.

To enable TTAPI for JDBC for a data collector, check the **Enable** box and click **Apply** at the bottom of the table. See "Enabling TTAPI for JDBC for a Data Collector" on page 64.

The **TTAPI JDBC enabled Data Collectors** table shows the data collectors that support TTAPI for JDBC and do not currently have it enabled. For every data collector, the server name and the **Enable** check box are shown.

To enable TTAPI for JDBC for a data collector, check the **Enable** box and click **Apply** at the bottom of the table. See "Disabling TTAPI for JDBC for a Data Collector" on page 64.

Configuring a data collector

About this task

When a new data collector connects to a managing server for the first time, it is automatically configured.

If you want to manually configure data collectors, you must ensure that for those data collectors, the dc.autoconfigure property in the dc.properties file is set to false.

For data collectors that you want to manually configure, you can assign an initial configuration using the data collector Configuration page in MSVE. When the data

collector acknowledges the new configuration, the managing server then lists it as a configured data collector. You can also return a currently configured data collector to an unconfigured state. If this is done, all data and reports concerning the data collector are lost. After a data collector has been configured, you can enable or disable it at any time.

For WebSphere v6, the data collector name is formed from the admin server name (node name) and the application server instance name, the data collector application server name also includes the profile name.

Here is an example of a WebSphere v6 profile name: jupiterCell01.jupiterNode01.server1(default).

In many places, an additional field is appended to the end of the data collector name that indicates whether the data collector is associated with a currently running application server instance or not. When the application server instance and data collector are running, this field is the process or address space identifier: jupiterCell01.jupiterNode01.server1.12345. When the data collector is not running and there is no process or address space identifier, two dashes (--) are used, for example: jupiterNode01.server1.--

To configure a data collector:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Unconfigured Data Collectors** on the left navigation pane.
3. Select a configuration from the **Apply a Configuration** list.
4. Click **Select All** or click in the check box of the unconfigured data collector that you want to configure.
5. Click **Apply**.

   The apply procedure can take a minute or two for the data collector to receive, successfully apply, and acknowledge its success back to the managing server. You must refresh your page every 10-20 seconds until the data collector disappears from the Unconfigured Data Collector table.

   Related topics

   Applying a configuration

   Modifying a configuration

   Unconfiguring a data collector

Unconfiguring a data collector

About this task

Use the Data Collector Overview page to unconfigure the data collectors. When you unconfigure a data collector, the system removes it from the configured data collectors list and displays it with the unconfigured data collectors.

The name of the Data Collector is a combination of the admin server name and the application server name, that is, admin_server.application_server. The name cannot be changed.

To unconfigure a data collector:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Place a check in the Unconfigure check box next to the data collector that you want to unconfigure.
3. Click **Apply**. The Unconfigured Data Collector Overview page displays.

Related topics

Applying a configuration

Configuring a data collector

Enabling a data collector

Enabling a data collector

About this task

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

To enable a data collector:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Enable** next to the data collector that you want to enable.
3. The system enables the data collector and the button face changes to **Disable**. 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. Because a disabled data collector does not lose settings, you can simply turn it back on without any reconfiguration.

Related topics

Configuring a data collector

Disabling a data collector

Modifying a configuration

Disabling a data collector

About this task

Disable your data collectors on the Configured Data Collector Overview page. Manage monitoring on your system by enabling and disabling data collectors as needed.

To disable a data collector:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Disable** next to the data collector that you want to disable.
3. The system disables the data collector and the button face changes to **Enable**.

Results

Disabling a data collector in the visualization engine disables only monitoring with the managing server; IBM Tivoli Monitoring is not affected. You can also use the data collector configuration to disable communication between the data collector and the managing server. The following table provides a comparison between these two ways of disabling data collector communication to the managing server:

Table 4. Comparison of ways to disable Data Collector communication to the Managing Server.

Disable data collector communication to the managing server using data collector configuration	**Disable data collector communication to the managing server using the visualization engine**
The application server instance is not listed in the visualization engine.	The application server instance remains listed in the visualization engine.
The visualization engine shows no information about the application server instance.	The visualization engine shows whether the application server instance is up or down; monitoring information is not available.
No system or network resources are used for managing server communication.	Some system and network resources are used to maintain managing server communication.
You do not need to apply maintenance fixes for the agent that only affect managing server communication.	You must apply maintenance fixes for the agent that only affect managing server communication.
In order to re-enable communication, you must complete data collector configuration again, and restart the application server.	In order to re-enable communication using the visualization engine, you do not need to restart the application server.

Related topics

Configuring a data collector

Enabling a data collector

Modifying a configuration

Creating a configuration

About this task

Use this page to create a configuration and name it for your data collectors. Create multiple configurations that monitor different classes.

The system assigns a name to the data collector. The name of the Data Collector is a combination of the admin server name and the application server name. The name cannot be changed.

To create a configuration:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Create a Configuration** on the left navigation pane. The Create page opens.
3. Enter the names of classes that you want to ignore into the Exclude (Classname) list. You can use an asterisk (*) as a wildcard. In this case it means exclude all.

4. Enter the names of classes that you want to monitor into the Exclude Override (Classname) list.
5. Select the check box if you want to enable MQ. This provides you with an Exclude and Exclude Override box specifically for configuring MQ.
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. You can configure or change these options at any time.

Related topics

Applying a configuration

Configuring a data collector

Applying a configuration
About this task

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:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured 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 that you want to apply. The Apply page opens.
4. Click to select the data collectors name from the All Data Collectors box. To select a range of contiguous data collectors, hold the shift key down during your selection. To add multiple servers non-contiguously, Ctrl + click the servers for selection.
5. Click **Add** to apply the configuration to the data collector. The Data Collector names display in the Applied box.
6. To set the status of the configuration, select **Enable** from the Status list.
7. Click **Apply.**

Related topics

Disabling a data collector

Enabling a data collector

Unconfiguring a data collector

Modifying a configuration
About this task

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:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.

2. Click **Configuration Library** on the left navigation pane. The Data Collector Configuration List page opens.
3. Click the Modify icon next to the configuration that you want to modify. The Modify page opens.
4. Enter the names of classes that you want to ignore into the Exclude (Classname) list.
5. Enter the names of classes that you want to monitor into the Exclude Override (Classname) list.
6. Select the check box to enable MQ list.
7. Click **Save** to save your modifications to the configuration. The Configured Data Collector Configuration List displays with the updated information.

Related topics

Applying a configuration

Configuring a data collector

Enabling a data collector

Duplicating a configuration

About this task

Create a configuration using an existing configuration from your data collectors.

To duplicate a configuration:

Procedure

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

Related topics

Applying a configuration

Configuring a data collector

Enabling a data collector

Deleting a configuration

About this task

You can delete outdated configurations from the list to keep your list current.

To delete a configuration:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Configuration Library** on the left navigation pane. The Data Collector Configuration List page opens.

3. Click the **Delete** icon next to the configuration that you want to delete. A confirmation box is displayed to warn you that deleting this configuration unconfigures all the associated servers.
4. Click **OK** to delete the configuration. The Configuration List displays without the deleted configuration. Remember to apply a new configuration to the servers you unconfigured while deleting the configuration.

Related topics

Applying a configuration

Configuring a data collector

Enabling a data collector

Modifying a configuration

Enabling TTAPI for JDBC for a Data Collector

About this task

For data collectors that support TTAPI for JDBC, enable it on the Enable TTAPI for JDBC page.

To enable TTAPI for JDBC for a Data Collector:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Enable TTAPI for JDBC** on the left navigation menu.
3. Click in the **Enable** check box next to the data collector for which you want to enable TTAPI for JDBC.
4. Click **Apply** at the bottom of the table.

Related tasks:

"Disabling TTAPI for JDBC for a Data Collector"

Related topics

Disabling TTAPI for JDBC for a Data Collector

About this task

For data collectors that have TTAPI for JDBC enabled, disable it on the Enable TTAPI for JDBC page.

To disable TTAPI for JDBC for a Data Collector:

Procedure

1. From the top navigation, click **Administration > Server Management > Data Collector Configuration**. The Configured Data Collector Overview page opens.
2. Click **Enable TTAPI for JDBC** on the left navigation menu.
3. Click in the **Disable** check box next to the data collector for which you want to enable TTAPI for JDBC.
4. Click **Apply** at the bottom of the table.

Related tasks:

"Enabling TTAPI for JDBC for a Data Collector"

Related topics

Data Collector Profiles

A data collector profile is a collection of multiple configuration changes that can be stored together and applied to a data collector. You can use the **Data Collector profiles** feature to modify the data collector properties file and toolkit files on a data collector, from the MSVE. It is still possible to modify these files when configuring a data collector. Data Collector profiles can be installed on monitored application servers running ITCAM for WebSphere/J2EE Data Collector 6.1 FP4 or higher. For more information, see the *ITCAM for Application Diagnostics - Agent for WebSphere Installation and Configuration Guide*.

From the MSVE, you can check which data collector profiles are installed on which data collector. You can add and remove profiles to and from data collectors. You can also import and export data collector profiles from other managing servers. If you change a data collector profile, it must be reinstalled on the data collector and the data collector must be restarted for the changes to take effect.

You can access the **Data Collector Profiles** using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The Data Collector profiles page displays a list of existing profiles if profiles have been created already. Click a profile name to view which data collectors are using that profile. The **DC Profiles Menu** contains a list of configurable items that you can modify on each profile.

For more information, see:

- "Viewing and editing data collector profiles"
- "Exporting a data collector profile" on page 67
- "Importing a data collector profile" on page 67
- "Adding and removing data collector profiles" on page 66
- "Configuring a data collector profile" on page 68
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Viewing and editing data collector profiles

You can view and edit profiles at any stage. You can also edit data collector profiles while they are assigned to a data collector.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

1. The Data Collector profiles page displays a list of existing profiles if profiles have been created already.
2. Click a profile name to view which data collectors are using that profile.

3. In the **DC Profiles Menu**, click the individual configuration items to view the details of each item.
4. For more information about the configuration items and how to edit them see "Configuring a data collector profile" on page 68.
5. In the **DC Profiles Menu**, you can click **Data Collectors** to view a list of data collectors available.
6. When you click a data collector name you can also click **Profiles Installed** to view the profiles assigned to the data collector.
7. If you are editing a profile, select a configuration item from the **Profiles** list in **DC Profiles Menu**, click **Apply** then click **Save** to add the changes to the profile.

When you edit a profile that is assigned to a data collector, the status of the profile changes to **Outdated** in the **Profiles Installed** list. To implement the updated profile changes on the data collector, you must reinstall the profile on the data collector. For more information, see "Installing a profile on a data collector" on page 75. Then the Data Collector needs to be restarted for the changes to take effect.

See also:

- "Data Collector Profiles" on page 65
- "Exporting a data collector profile" on page 67
- "Importing a data collector profile" on page 67
- "Adding and removing data collector profiles"
- "Uninstalling a data collector profile" on page 76

Note: If you click refresh or F5 on your browser from any page in Data Collector Profiles, you return to the main **Data Collector Profiles** page.

Adding and removing data collector profiles

You can add or remove data collector profiles at any stage.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The page has two main areas: the **DC Profiles Menu** and the **Profiles** area of the page. You can use the **DC Profiles Menu** to edit, add, and remove configurable items from profiles. You can use the **Profiles** area of the page to add, edit, and remove data collector profiles.

- Click **Profiles** to view a list of existing profiles.
- Click **DC Profiles Menu** to view a list of available data collectors.

1. To add a profile, click **New**.
2. In the **Name** field, type the name that you want to give to the profile.
3. Click **Apply**.
4. In the **DC Profiles Menu**, click the items that you want to configure to add to this profile.
5. For more information, see "Configuring a data collector profile" on page 68.

6. Then click **Save** to add the profile to the list of Data collector profiles.
7. To delete a profile select the check box then click **Delete**.

When you delete a profile that is assigned to a data collector the status of the profile changes to **Outdated** in the **Profiles Installed** list. The Data Collector needs to be restarted for the changes to take effect.

See also:

- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Exporting a data collector profile"
- "Importing a data collector profile"
- "Configuring a data collector profile" on page 68
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Importing a data collector profile

You can import a data collector profile from other locations for use on other managing servers.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**. Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

1. The Data Collector profiles page displays a list of existing profiles if profiles have been created already.
2. Click **Browse** to locate the .xml file that you want to import.
3. Select the .xml file and click **Open**.
4. Select the profile that you want to overwrite by selecting the check box in the Select column.
5. Click **Import**.
6. If you are importing a file with the same profile name a message displays. Click **OK** to overwrite the existing profile.
7. If you are importing a new file, then a new profile is created.

See also:

- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Exporting a data collector profile"
- "Adding and removing data collector profiles" on page 66
- "Configuring a data collector profile" on page 68

Exporting a data collector profile

You can export data collector profiles to other locations for use on other managing servers.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

1. The Data Collector profiles page displays a list of existing profiles if profiles have been created already.
2. Select the profile by selecting the check box in the **Select** column.
3. Click **Export** to launch an .xml version of the profile in your browser.
4. From the browser menu, click **File> Save as** and select the location that you want to export the file to.
5. Then close the browser.

See also:

- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Importing a data collector profile" on page 67
- "Adding and removing data collector profiles" on page 66
- "Configuring a data collector profile"
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Configuring a data collector profile

When you create a profile, you can configure a number of items. You can select the items that you want to configure; not all items are required to be configured. You can also modify existing items. When you have configured a profile, you can install it on one or more data collectors.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The Data Collector profiles page displays a list of existing profiles if profiles have been created already. Click the Profile, a list of configurable items displays in the **DC Profiles Menu**.

Click the following links for more details on configuring each item in a data collector profile.

- "PMI" on page 69 (WebSphere products only)
- "Custom request" on page 70
- "Custom L2" on page 71
- "Method Entry and Exit" on page 72
- "Lock Analysis" on page 73
- "Memory Diagnosis" on page 73
- "Custom MBeans" on page 74

- "Tuning Parameters" on page 75

See also:
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Importing a data collector profile" on page 67
- "Exporting a data collector profile" on page 67
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

PMI:

This feature is available for WebSphere products only. Standard WebSphere Application Server PMI modules are automatically included in monitored JVM and are rendered on the MSVE. If your environment is using WebSphere Process Server or WebSphere Commerce, the PMI modules associated with these products are not rendered to the MSVE. You can use this feature to add the PMI module names for these products to data collector profiles.

For information about the PMI module names for these products, see the following documentation links:
- WebSphere Commerce http://publib.boulder.ibm.com/infocenter/wchelp/v6r0m0/index.jsp
- WebSphere Process Server http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wps.620.doc/welcome_wps.html

To add a PMI module to a data collector profile, choose one of the following options to access the PMI feature:
- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**. Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The Data Collector profiles page displays a list of existing profiles if profiles have been created already.
1. Click the profile **Name**.
2. In the **DC Profiles Menu**, click **PMI**.
3. In the **PMI Module Name** field, type the module name add a custom PMI module.

   Note:
   This information applies to WebSphere products only.
4. Click **Apply** or **Save** to implement the changes to the profile.

When the PMI module name is added to the profile, the next step is to install the profile on a data collector. When the managing server is restarted, the PMI module displays on the MSVE **System Resource** page, when the corresponding data collector is selected.

See also:
- "Custom request" on page 70

- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Custom request:

A custom request is an application class and method that you designate as an edge or nested request. This feature defines custom request methods and classes that are included in L2 Method Trace.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**. Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The Data Collector profiles page displays a list of existing profiles if profiles have been created already.

1. Click the profile **Name**.
2. In the **DC Profiles Menu**, click **Custom Request**.
3. Click **New**.
4. Complete the following fields:
   - **Request Name**: Identifies the unique name for this request. The request name is displayed in the L1 or L2 trace entry that is produced when one of the methods identified by this custom request runs.
   - **Class Name**: Identifies the name of the class.
   - **Method Name**: Identifies the names of the methods within one of the classes that are to be Byte-Code-Instrumented for custom request processing.
5. Select the **Type** by clicking one of the following options:
   a. **Class**
   b. **Superclass**
   c. **Interface**
6. Click **OK** to save the Custom Request.
7. To edit a custom request, click **Custom Request** and edit the fields on display, click **Save** to implement the changes to the profile.
8. To delete a custom request from a data collector profile, on the **Profile - Custom Requests** page select the check box in the select column, then click **Delete**.

See also:

- "Custom L2" on page 71
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66

- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Custom L2:

The data collector uses a technique called Byte Code Instrumentation (BCI) to collect data from various types of J2EE APIs that typically operate as nested requests. BCI is automatically enabled for these types of APIs. You can use Custom L2 to enable and disable L2 instrumentation components if you want to lower the monitoring workload on the data collector. You can also define new L2 events using the Custom L2 events option.

1. Click the profile **Name**.
2. In the **DC Profiles Menu**, click **Custom L2**. You can enable or disable the following events:
   - **SERVLET**
   - **JDBC** (Java Database Connectivity)
   - **JMS** (Java Message Service)
   - **HTTP Session Count**
   - **CTG** (CICS® Transaction Gateway)
   - **JDO** (Java Data Objects)
   - **MQI** (Message Queue Interface)
   - **EJB** (Enterprise Java Beans)
   - **JNDI** (Java Naming and Directory Interface)
   - **JCA** (Java Connector Architecture)
   - **Axis Web Service (JBoss and Weblogic)**
   - **IMS**
   - **RMI** (Resource Manager Interface)
3. Select the check box to enable an event.
4. Clear the check box to disable an event.
5. To add Custom L2 events In the Custom L2 event area of the page click **New**.
6. In the Event Type drop down menu, select one of the following options to enable or disable the instrumentation:
   - **SERVLET**
   - **EntityBean**
   - **SessionBean**
   - **Message Driven Bean**
7. Type the **Class Name**.
8. Select the **Enabled** check box to save the event.
9. On the main page, select the check box to assign the custom event to the profile.
10. Click **Apply**, click **Save**.

- To edit an L2 custom event, click the **Event Type**, add the changes click **Apply**, click **Save**.
- To delete an L2 custom event, select the check box for the **Event Type**, click **Delete**.

See also:

- "Method Entry and Exit" on page 72

- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Method Entry and Exit:

You can use this item to select classes to include or exclude on the monitored JVM used for classes and methods in L3 methods traces.

1. In the **DC Profiles Menu**, click **Method Entry/Exit**.
2. A list of existing classes assigned to the profile display.
3. Click **New** to add a new class to the profile.
4. From the **Lookup Sever** drop-down menu, select the server.
5. A list of available classes display.
6. Click a file to expand the view and see the classes.
7. Select the check boxes associated with the classes that you want to add and click **Add**.
8. Click **Save** to add the classes to Method Trace Entry and Exit.
9. To remove a class, select the check box for the class that you want to remove. Then, click **Apply** and **Save**.

To create a class to add to **Method Entry/Exit**.

1. Type **Class Name**.
2. Type the **New Method Name**.
3. Click **Add** to display the class.
4. Select the check box then click **Apply** to add the class to the list of existing classes for Method Exit Entry.
5. From the list, select the check boxes for the classes that you want to add.
6. Click **Save**.

- To edit a class, in the **Method Entry/Exit** page, click the **Class Name**, add the changes then click **Apply**.
- To remove a class, select the check box for the class. Then click **Delete** and then click **Save**.

See also:

- "Lock Analysis" on page 73
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Lock Analysis:

You can use this feature to probe classes for lock analysis. All application classes are included by default. You can access these classes from the **Lookup Server**. Use the add option to add new entries containing classes or methods to be included or excluded for lock analysis.

1. In the **DC Profiles Menu**, click **Lock Analysis**.
2. From the **Lookup Sever** drop-down menu, select the server.
3. A list of available classes display.
4. Click a file to expand the view and see the classes.
5. Select the check boxes associated with the classes that you want to add and click **Add**.
6. Click **Save** to add the classes to Lock Analysis.

Use the following steps to add a class:

1. To add a class, in the **DC Profiles menu**, click **Lock Analysis**.
2. Type **New Class Name**.
3. Click **Add**.
4. In the**Classes to Monitor** list, select the check box then click **Apply** and then **Save** to add the class to Lock Analysis.
5. To remove a class, select the check box for the class that you want to remove. Then click **Delete** and then click **Apply** and **Save**.

See also:

- "Memory Diagnosis"
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Memory Diagnosis:

You can use this feature to select classes to use in Memory diagnosis. All memory classes are included for memory leak diagnosis in L3 mode by default. Memory diagnosis can be restricted to a selected combination of allocating and allocated classes.

1. In the **DC Profiles Menu**, click **Memory Diagnostic**.
2. To add heap allocation classes click **New**.
3. In the **New Allocating Class Name** field, type the name of the class that allocates objects to Memory Diagnosis. **New Allocating Class Name** identifies the name of a class or classes to be modified.
4. In the **New Allocated Class Name** field, type the name of the class allocated to include in the Memory Diagnosis. The **New Allocated Class Name** identifies the specific heap allocation requests within the class or classes.
5. Click **Add** to add the class to the **Allocated Classes to monitor** list.
6. In the **Allocated classes to monitor** list, select the check box for the class name that you want to add to the Allocating classes.
7. Click **Apply** and **Save** to save the class to the Heap Allocations list.

8. To remove a class from the**Allocated Classes to monitor** list, select the check box. Then click **Delete** and then click **Apply** and **Save** to implement the changes.

- To edit a class, from the **Heap Allocations** list, click the class name, make the necessary changes then click **Apply** and **Save**.
- To delete a class from the **Heap Allocations** list, select the check box. Then click **Delete** and then click **Apply** and **Save** to implement the changes.

See also:

- "Custom MBeans"
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector" on page 75
- "Uninstalling a data collector profile" on page 76

Custom MBeans:

You can include MBeans to be rendered on the **System resource** page on the MSVE. You can add new entries or use the lookup server drop-down menu to query existing beans and select attributes.

1. In the **DC Profiles Menu**, click **Custom MBeans** to display the Profile Custom MBean page.
2. Use the following steps to manually add Custom MBeans to the managing server.
3. **Domain**: Type the Domain name.
4. **Object Name**; Type the MBean Object Name.
5. **Category**: Type the unique Category Name (used by ITCAM).
6. Select the **Retrieve all Attributes** check box if you want to add all attributes associated with the MBean.
7. **Attribute Name**: Type the name of the attribute.
8. **Mapped Key Name**: Type the unique key string to map the attribute (used by ITCAM).
9. **Object Name Pattern**: Type the MBean pattern to search in the form.
10. Click **Add** to add the MBean to the list.

To add MBeans from the **Look up Server** drop-down menu use the following steps:

1. From the **Look up Server** drop-down menu, select the application server.
2. From the **Select MBean** drop-down menu, select the MBean that you want to add.
3. From the **Select Attributes** drop-down menu, select the attribute that you want to add.
4. Click **Apply** to add the Custom MBeans to the list.
5. From the **Custom MBeans** list, select the check boxes for the Custom MBeans that you want to add to the data collector profile.
6. Click **Apply** then click **Save**.

- To edit **Custom MBeans** from the **Custom MBeans** list, click the MBean **Object name**, make the necessary changes then click **Apply** and **Save**.

- To delete **Custom MBeans** from the **Custom MBeans** list, select the check box. Then click **Delete** and then click **Apply** and **Save** to implement the changes.

See also:
- "Tuning Parameters"
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector"
- "Uninstalling a data collector profile" on page 76

Tuning Parameters:

You can configure tuning parameters and apply then to the data collector profiles on the data collectors.

Note: Data Collector Tuning parameters cannot be reversed by uninstalling the profile, However, they can be modified back to the original value.

1. In the **DC Profiles Menu**, click **Tuning Parameters**.
2. From the **Tuning Parameters** drop-down menu, select the parameter that you want to add.
3. Repeat this process for each parameter that you want to add.
4. Detailed information relating to each parameter you select displays in the **Selected Tuning Parameters** page.
5. When a selected parameter displays in the list, type a value in the **Value** column for each parameter.
6. Click **Apply** and then click **Save** to add the parameters to the data collector profile.
7. Select the check box to add a parameter to the data collector profile. Click **Apply** then **Save**.
8. To remove a parameter, select the check box then click **delete** and then **Save** to remove a parameter from the profile.

See also:
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Installing a profile on a data collector"
- "Uninstalling a data collector profile" on page 76

Installing a profile on a data collector

When you create a data collector profile, the next step is to add it to a data collector. You must then restart the managing server for the changes to take effect.

Access the data collector profiles using one of the following methods:
- From the main menu, click **Administration > Server Management > Data Collector Profiles**.

- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The data collector profiles page displays a list of existing profiles if profiles were created already.

1. To view the list of data collectors assigned to that profile, click the profile name .
2. If you want to view details of the profile, from the **DC Profiles Menu**, click the configurable items.
3. To assign a profile to a Data Collector, from the **DC Profiles Menu**, click **Data Collectors** to view all the available data collectors.
4. Select the check box in the Select column of each data collector that you want to add a new profile to.
5. To display the list of the data collector profiles that are available, click **Install**.
6. In the Select column of the profile that you want to add, select the check box.
7. To retain any manual entries made by manually editing the xml and the properties file as well as the changes made to the profile, click **Merge**.
8. To remove all manual changes and to replace them with the changes in the data collector profile, click **Overwrite**.
9. If you are reinstalling an updated version of an existing profile, click **Overwrite** to update the profile on the data collector. You must restart the managing server for all changes to take effect.

See also:

- "Uninstalling a data collector profile"
- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Importing a data collector profile" on page 67
- "Exporting a data collector profile" on page 67
- "Uninstalling a data collector profile"

Uninstalling a data collector profile

You can uninstall a data collector profile from a data collector at any stage. You must then restart the managing server for the changes to take effect.

Access the Data Collector Profiles using one of the following methods:

- From the main menu, click **Administration > Server Management > Data Collector Profiles**.
- From the main menu, click **Problem Determination > Server Activity Display**.

  Click the toolbox icon  on the **Active Requests** tab or the **Recent Requests** tab.

The Data Collector profiles page displays a list of existing profiles if profiles have been created already.

1. From the **DC Profiles Menu**, click **Data Collectors** to view all the available data collectors. Click **Refresh list** update the current list of available data collectors

2. Click a data collector name to view the list of profiles assigned to the data collector.
3. Select the check box in the Select column of the profile that you want to remove.
4. Click **Uninstall** to remove the profile from the data collector.
5. You must then restart the managing server for the changes to take effect.

See also:

- "Configuring a data collector profile" on page 68
- "Data Collector Profiles" on page 65
- "Viewing and editing data collector profiles" on page 65
- "Adding and removing data collector profiles" on page 66
- "Importing a data collector profile" on page 67
- "Exporting a data collector profile" on page 67
- "Installing a profile on a data collector" on page 75

Monitoring on Demand (TM)

In the Monitoring on Demand (MOD) pages, you can view and adjust the monitoring level for all servers, and schedule adjustment of this level at fixed times.

To view the Monitoring on Demand page, from the top navigation, click **Administration > Monitoring on Demand**. On the Monitoring Schedule (to the far right of panel), click the **Schedule Change/Override** button for the Group/Server name to display the **Modify Server Settings** page. For z/OS, click the **Schedule Change/Override** button for the WebSphere Application Server Control Region to display the **Modify Server Settings** page.

You can also access this page from the toolbox icon  on the Server Activity Display page.

You can use the **Modify Server Settings** page to change the monitoring level for the selected Data Collector, select a different schedule, or change the current sampling rate. Overriding a monitoring level lasts until the next monitoring level begins, as determined by the schedule.

Selected Group/Server displays the server the changes are applied to.

1. Select the **Schedule Selection** from the drop-down menu.
2. Select the **Override Monitoring Level** from the drop-down menu.
   - System Default
   - L1 Production Mode
   - L2 Problem Determination Mode
   - L3 Tracing Mode
3. Select the **Enable Method Profiling** check box and type the number of minutes that you want to use.
4. Select the **Sampling Rate** for the monitoring level you select.
5. **Override Transaction CPUMON Level (CICS only)** Select the level from the drop-down menu.

The *monitoring level* for a server defines the amount of data collected for it. The higher the monitoring level, the more details are collected and the larger processor usage is introduced by the Data Collector. The available Monitoring Levels are L1 (Production mode), L2 (Problem Determination mode), and L3 (Tracing mode).

You can create *schedules* that defines preset times at which the monitoring level is to be changed, then apply different schedules to different servers. You can also set (override) the monitoring level manually for any server at any time.

To view the monitoring level for all servers, adjust it manually and assign schedules for its adjustment, select **MOD Console** in the left navigation pane. See "MOD Console."

To view the schedules and modify them, select **Schedule Management** in the left navigation pane. See "Schedule Management" on page 79.

To create a new blank schedule, select **Create Schedule** in the left navigation pane. See "Creating a schedule" on page 79.

To create a schedule as a copy of an existing schedule, select **Duplicate Schedule** in the left navigation pane. See "Duplicating a schedule" on page 82.

Tip: To change the default monitoring level, used when no schedule or override is applied, see "Configuring the Data Collection settings" on page 83.

Related topics

"Configuring the Data Collection settings" on page 83

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 is when the load is the lightest. As the ITCAM for Application Diagnostics 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 ITCAM for Application Diagnostics 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.

MOD Console

The MOD Console page shows a table of all servers and the monitoring level set for them. You can manually set (override) this level and assign a schedule for automatically changing it as preset times.

To see servers only in a particular server group, use the **All Groups** pull-down control at the top of the table to select the group.

For every server, the table shows:

- Group and Server name.
- Platform (operating system) the server uses.

- The name of the Schedule currently applied to it. If a Schedule is applied, it determines automatic adjustment of server monitoring level at preset times.
- Current monitoring level.
- Current sampling rate.
- Current CPU load.
- **Schedule Change/Override** button.

To set (override) the monitoring level for a server or to assign a schedule for automatically changing it as preset times, click the **Schedule Change/Override** button. See "Overriding a monitoring level" on page 81 and "Applying a schedule" on page 80.

To view or modify a schedule, click the schedule name. See "Modifying a schedule" on page 81.

You can use the left navigation pane to view the existing schedules and create a schedule. See "Monitoring on Demand (TM)" on page 77.

Schedule Management

The Schedule Management page shows a table of all schedules. A schedule defines preset times at which the monitoring level for a server must be changed. You can modify and delete the schedules, and set the monitoring level or change the schedule for servers.

For every schedule, the table shows the schedule name, the server name, and a **Delete** button.

To modify a schedule, click the schedule name. See "Modifying a schedule" on page 81.

To set (override) the monitoring level or change the schedule for a server, click the server name. See "Overriding a monitoring level" on page 81 and "Applying a schedule" on page 80.

To delete a schedule, click the **Delete** button. See "Deleting a schedule" on page 82.

You can use the left navigation pane to view the monitoring levels for all servers and create a schedule. See "Monitoring on Demand (TM)" on page 77.

Creating a schedule

About this task

At times a server might need more detailed monitoring. 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.

To create a schedule:

Procedure

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

2. Click **Create Schedule** on the left navigation pane. The Schedule Detail page opens where you can create a 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 to take effect. For example, you might want the schedule to start on the fifth of every month or on every Monday.
5. Select the Hour and Minute when the schedule starts.
6. Select the Monitoring Level that best suits your needs: L1, L2, or L3.
7. Click **Add** to insert the settings into the schedule. Each schedule can include multiple monitoring level changes; to save each change, click **Add**.
8. To save the schedule, click **Save**. The Schedule Management page opens with the new schedule displayed. If a schedule conflict occurs, the most recently entered item takes precedence.

   Do not create an overly complicated schedule or you will never know at what level of monitoring your servers are running. Keep the rules simple. L1 has the smallest processor usage, while L3 is heavier. When L2 applies, it has optimum processor usage and allows you to switch to L3 without the need to restart. You might want at minimum one, or a maximum of 5% of your servers running at L3, either as dedicated servers, or only during non-peak hours. This arrangement gives you good quality data for workload tracing and application sizing. In an z/OS environment, you might want to create a server instance that runs at L3.

   Related topics

   Applying a schedule

   Overriding a monitoring level

   "Overriding a monitoring level" on page 81

Applying a schedule

About this task

After creating a schedule, you can apply it to a server that needs monitoring. You might also apply other existing schedules to a server.

To apply a schedule:

Procedure

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 where you want to apply a schedule. The Modify Server Settings page opens.
3. Click to select a schedule from the schedule selection list for the server.
4. Enter a percentage in the boxes for sampling setting or choose the default setting by checking the system default box.
5. Click **OK**. The MOD Console page displays the server with the schedule applied in the table.

   Note: A schedule can be applied at the group level and the all servers level as well.

   Related topics

   Creating a schedule

   Deleting a schedule

Overriding a monitoring level

About this task

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

To override a monitoring level:

Procedure

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 or group that you want to override its current monitoring level. The Modify Server settings page opens.
3. Click to select a monitoring level from the override monitoring level drop-down box. If you select L2, you can check MP for Method Profiling. The time interval setting in Enable Method Profiling is for how often the data collector is to aggregate method data and push the data to the managing server. For example, if the interval setting is 60 minutes, then every 60 minutes the Data Collector aggregates the method data and pushes the data to the managing server. To disable the Method Profiling feature, clear "Enable Method Profiling". You can view the method profile reports on the Method Profiling Management page. If you override the system default monitoring level, the request sampling setting is also changed according to the default monitoring level that the server is running on.
4. Click **OK**. Due to processor usage when collecting transaction CPU in CICS, ITCAM for Application Diagnostics offers the option to specify the level for collecting CICS CPU. Called the Transaction CPUMON Level, this option is available on the Monitoring on Demand Override page (accessed via Administration > Monitoring on Demand, click **Schedule Change/Override**) and on the System Properties page (accessed via **Administration > Managing Server > System Properties**). In both cases, the options include:
   - Do Not Collect
   - Collect at Monitoring Level L1, L2, and L3
   - Collect at Monitoring Level L2 and L3
   - Collect at Monitoring Level L3 only

   With these alternatives, you can plan to collect transaction CPU when the load on your system is not overwhelming. Collect at Monitoring Level L3 only is the default.

Modifying a schedule

About this task

If you find that an existing schedule is not providing the correct level of monitoring, modify the schedule to reflect your needs. Keep your schedules current based on the ever fluctuating needs of your data center.

To modify a schedule:

Procedure

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 that 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 setting.
5. Click **Add** to insert the settings into the schedule. Each schedule can include multiple monitoring level changes; to save each change, click **Add**.
6. To save the schedule, click **Save**.

Note: Changes take effect immediately.

Related topics

Applying a schedule

Configuring the Data Collection settings

Deleting a schedule

About this task

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

To delete a schedule:

Procedure

1. From the top navigation, click **Administration > Monitoring on Demand TM**. The Monitoring on Demand TM (MOD) Console page opens.
2. Click **Schedule Management** on the left navigation pane. The Schedule Management page opens.
3. Click **X** or **Delete** next to the schedule that you want to remove.
4. At the confirmation box, click **OK** to delete the schedule. The Schedule Management page displays without the deleted schedule.

   If a schedule is currently being used by a server, you must apply another schedule to that server or it will automatically apply the system default after you delete the schedule.

Duplicating a schedule

About this task

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

To duplicate a schedule:

Procedure

1. From the top navigation, click **Administration > Monitoring on Demand TM**. The Monitoring on Demand TM (MOD) Console page opens.
2. Click **Duplicate Schedule** on the left navigation pane. The Duplicate Schedule page opens.
3. From the Schedule list, select the schedule that 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.

Related topics

Applying a schedule

Modifying a schedule

Managing server

Tune and debug the managing server.

The managing server section is separated into two categories: system properties and self-diagnosis. System properties enable you to tune ITCAM; self-diagnosis provides you with a method for debugging the managing server when problems arise.

System properties

In the System Properties pages, you can set defaults for server data collection, baseline settings for the Enterprise Overview display in the visualization engine, and SNMP settings.

To view the System Properties pages, from the top navigation, click **Administration > Monitoring Server > System Properties**.

To set default values for server data collection, select **Data Collection Settings** in the left navigation pane. See "Configuring the Data Collection settings."

To set the values for baseline indication in the Enterprise Overview display, select **Enterprise Overview Display** in the left navigation pane. See "Configuring the Enterprise Overview display" on page 85.

To configure the SNMP network, select **SNMP** in the left navigation pane. See "Configuring the SNMP network" on page 86.

Configuring the Data Collection settings

About this task

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, and the level of monitoring. The default data collection settings can be established at the managing server level, then overridden by configuration settings specific to each configured data collector.

To configure the Data Collection settings:

Procedure

1. From the top navigation, click **Administration > Managing Server > System Properties**. The System Properties page opens.
2. Enter the appropriate value for the following properties:
   - **System Resources Polling Frequency** - Set how often the system resources requests information from your application server. 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%.
- **Default Monitoring Level** - The currently set default monitoring level for all servers connected to the application monitor. This is the case when configuring a server for the first time and bringing up the server under the management of the application monitor. The default monitoring level for the non z/OS platform is L2 (Problem Determination). As for the z/OS platform, the default monitoring level is L1 (Production Mode). The monitoring levels are as follows:
  - **L1 (Production mode)** – this monitoring level provides availability management, system resources, and basic request-level data. This monitoring level has a low affect on the CPU usage 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. Use this level when you suspect a problem or when you want to capture data about external events but do not need all the method-level data. When you select L2, you can check MP for Method Profiling. This feature allows you to determine how often the data collector aggregates method data and send the data to the managing server: 1-999 minutes. You can view the method profile reports on the Method Profiling Management page.
  - **L3 (Tracing Mode)** – this is the most powerful monitoring level, therefore only this level uses 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. Use this level for servers that have been selected for diagnostics and detailed workload characterization.
- **Transaction CPU Time (CICS only)** - This field indicates whether the CPU time for a CICS transaction is collected or not. The CPU times for CICS tasks are reported, but to avoid processor usage from the data capture, adjust these settings as necessary. Select one of the following four options for collecting transaction CPU:
  - Do Not Collect
  - Collect at Monitoring Level L1, L2, and L3
  - Collect at Monitoring Level L2 and L3
  - Collect at Monitoring Level L3 only
- **Maximum Method Records** - The maximum number of method trace records. The records are cycled through, showing the 10,000 (or Maximum Method Records value) most recent methods in the system. The default value is 10,000.
- **Maximum IMS™ Message Data Length** - The maximum length of the IMS message data. The default is 256.

3. Click **Save**.

Related topics

Configuring the Enterprise Overview display

Configuring the Enterprise Overview display

About this task

Use the Enterprise Overview Display settings to set the response time Baseline Indicators and Baseline Definitions. The Baseline Indicator is the percentage above the baseline that you determine to indicate slow or very slow response. The Baseline Definition determines how the baseline is calculated. These thresholds apply to the average of average response times for all servers in the group.

To configure the Enterprise Overview display:

Procedure

1. From the top navigation, click **Administration > Managing Server > System Properties**. The System Properties page opens.
2. On the left navigation pane, click **Enterprise Overview Display**. The Enterprise Overview Display page opens.
3. Enter the appropriate value for the following properties:
   - **Baseline Indicator Settings** - The percentage above the baseline that indicates slow or very slow response times. Slow response means that the present response time is between 26% and 50% of the baseline. A very slow response means that the present response time exceeds 50% of the baseline. All averages are over 5 minute intervals. For example, if Indicator 1 is set to 25%, and Indicator 2 is set to 50%, average response times between 125% and 150% of the baseline are considered slow responses. Average response times above 150% of the baseline are considered very slow responses.
   - **Baseline Definition Settings** - The method to be used for determining the response time baseline.
     - **Rolling Date** - Historical response time data for this number of days is averaged to determine the baseline.
     - **Fixed Date** - Baseline is the average response time over the time interval midnight on the start date to 11:59PM on the end date.
     - **Fixed Response Time** - The response time entered in this field becomes the response time against which your current response times on the enterprise overview page is compared.
4. Click **Save**. These properties are defined at the group level for the servers when they are added to a group. The group properties take precedence over the system properties. When the response time reaches Indicator 1, an orange indicator displays on the Enterprise Overview page. A red indicator means that the response time has exceeded Indicator 2.

Related topics

Configuring the Data Collection settings

Configuring the SNMP network

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

Configuring the SNMP network
About this task

Use the SNMP network settings to indicate the configuration for the SNMP server. A test message is sent to the SNMP network manager to test for an open connection.

Note: Consult the Installation and Customization Guide for further instructions on the setup required to configure the SNMP network.

To configure the SNMP network:

Procedure

1. From the top navigation, click **Administration > Managing Server > System Properties**. The System Properties page opens.
2. On the left navigation pane, click **SNMP**. The SNMP Network Configuration page opens.
3. Click the **Add SNMP Configuration** button. 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 are sent.
   - **Port Number** - The port number of your SNMP Network Managers.
   - **Community** - A string that is part of the SNMP protocol.
4. Click **Test and Save** to send a test message to the SNMP Network Manager.
5. Click **Save** to save your settings.

Related topics

Configuring the Data Collection settings

Configuring the Enterprise Overview display

Self-diagnosis

In the Self-diagnosis pages, you can view details of ITCAM operation. This information can be useful for debugging.

To view the Self-diagnosis page, from the top navigation, click **Administration > Managing Server > Self-Diagnosis**.

These pages are designed for the Support staff to service ITCAM. The self-diagnosis provides a view of all the components currently running, their states, and attributes. Since ITCAM 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.

Use the left navigation pane to select the managing server components for which debugging information is shown.

Viewing the Self-diagnosis for the kernel
About this task

The Self-diagnosis provides a view of all the components on the kernel currently running and their attributes. Because the application monitor is a dynamic environment, the components can be up or down without affecting the integrity of the whole system.

To view the Self-diagnosis for the kernel:

Procedure

1. From the top navigation, click **Administration > Managing Server > Self-diagnosis**. The Self-diagnosis page opens displaying the data for the kernel.
2. In the left navigation pane, click the **+ Kernel Instances**.
3. Click to select the kernel link that you want to view. The kernel runtime environment detail is displayed.

   Related topics

   Viewing the Self-diagnosis for the archive agent

   Viewing the Self-diagnosis for the publish server

   Viewing the Self-diagnosis for the global publish server

   Viewing the Self-diagnosis for the data collector controller

   Viewing the Self-diagnosis for the message dispatcher

Viewing the Self-diagnosis for the archive agent

About this task

The archive agent aggregates the data from the publish server and archives it into the database for reporting. The Self-diagnosis provides a view of all the components on the archive agent that are currently running and their attributes.

To view the Self-diagnosis for the archive agent:

Procedure

1. From the top navigation, click **Administration > Managing Server > Self-diagnosis**. The Self-diagnosis page opens.
2. In the left navigation pane, click **+ Archive Agents**.
3. Select the archive agent that you want to view. The data for the selected archive agent displays.

   Related topics

   Viewing the Self-diagnosis for the kernel

   Viewing the Self-diagnosis for the publish server

   Viewing the Self-diagnosis for the global publish server

   Viewing the Self-diagnosis for the data collector controller

   Viewing the Self-diagnosis for the message dispatcher

Viewing the Self-diagnosis for the publish server

About this task

The publish server retrieves data from the data collector and aggregates it based on different needs. The Self-diagnosis provides a view of all the components on the publish server currently running and their attributes.

To view the Self-diagnosis for the publish server:

Procedure

1. From the top navigation, click **Administration > Managing Server > Self-diagnosis**. The Self-diagnosis page opens.
2. In the left navigation pane, click **+ Publish Servers**.

3. Click to select the publish server that you want to view. The data for the selected publish server displays.

Related topics

Viewing the Self-diagnosis for the kernel

Viewing the Self-diagnosis for the archive agent

Viewing the Self-diagnosis for the global publish server

Viewing the Self-diagnosis for the data collector controller

Viewing the Self-diagnosis for the message dispatcher

Viewing the Self-diagnosis for the global publish server

About this task

The global publish server monitors composite requests, as they move from one server to another. The Self-diagnosis provides a view of the attributes of the global publish server, as well as all the components with which the global publish server has relationships.

To view the Self-diagnosis for the global publish server:

Procedure

1. From the top navigation, click **Administration > Managing Server > Self-diagnosis**. The Self-diagnosis page opens.
2. In the left navigation pane, click **+ Global Publish Servers**.
3. Click to select the global publish server that you want to view. The data for the selected global publish server displays.

Related topics

Viewing the Self-diagnosis for the kernel

Viewing the Self-diagnosis for the archive agent

Viewing the Self-diagnosis for the publish server

Viewing the Self-diagnosis for the data collector controller

Viewing the Self-diagnosis for the message dispatcher

Viewing the Self-diagnosis for the data collector controller

About this task

The data collector controller 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 all the components on the data collector controller currently running and their attributes.

To view the Self-diagnosis for the data collector controller:

Procedure

1. From the top navigation, click **Administration > Managing Server > Self-diagnosis**. The Self-diagnosis page opens.
2. In the left navigation pane, click the Data Collector Controller link.
3. Click to select the data collectors that you want to view. The data for the selected data collector controller displays.

Related topics

Viewing the Self-diagnosis for the kernel

Viewing the Self-diagnosis for the archive agent

Viewing the Self-diagnosis for the publish server

Viewing the Self-diagnosis for the global publish server

Viewing the Self-Diagnosis for the message dispatcher

Viewing the Self-diagnosis for the message dispatcher

About this task

The message dispatcher sends out e-mails of performance reports and trap results from the Performance Analysis and Reporting and the Trap and Alert Management applications. The Self-diagnosis shows all the attributes of the message dispatcher currently running such as total number of e-mails sent.

To view the Self-diagnosis for the message dispatcher:

Procedure

1. From the top navigation, click **Administration > Managing Server > Self-diagnosis**. The Self-diagnosis page opens.
2. In the left navigation pane, click **+ Message Dispatchers**.
3. Click to select the message dispatcher link that you want to view. The message dispatcher runtime environment detail displays.

Related topics

Viewing the Self-diagnosis for the kernel

Viewing the Self-diagnosis for the archive agent

Viewing the Self-diagnosis for the publish server

Viewing the Self-diagnosis for the global publish server

Viewing the Self-diagnosis for the data collector controller

Systems overview

Assess your entire system.

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

User Scenarios

Scenario 1: Verifying customer response time complaints

Customer service has been receiving complaints that your company 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 administrators, you go to the Enterprise Overview page and see that all the servers are up. You find the group that seems 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 page shows a table of summary availability information for each server group.

To view the Enterprise Overview page, from the top navigation, click **Availability > Systems Overview > Enterprise**.

For every server group ("ALL" is a group consisting of all servers communicating to the managing server) the table shows:

- A "traffic-light" indicator. Green means good availability, orange means problematic availability, and red means an alert. The indicator is set based on comparing the current response time with a baseline time determined from the average for a certain time.
- Number of available servers, total number of servers in the group, and the percentage of servers that are available.
- A **Maintenance Mode** column displays if a server group has one or more WebSphere Dynamic Cluster members. It shows the number of clusters in maintenance mode compared to the total number of servers in the group.
- A graph showing throughput (requests per 5 minutes) for the last hour.
- A graph showing the average response times, in milliseconds, over every 5 minutes for the last hour.
- A tool button.

To see availability information for individual servers in a group, click the name of the group. The “Group Overview” page opens.

To see the activity for a group, click the tool button and select **Server Activity Display**. See “Server Activity Display” on page 116.

To find a specific request for this group, click the tool button to select **In-flight Request Search**. The “In-flight request search” on page 114 page opens.

To set the Enterprise Overview as the default page when you open the visualization engine, click **Set as My Default page** at the upper right of the page.

To switch to the “Group Overview,””Server Overview” on page 91“Alerts and Events” on page 108“Problem Center” on page 110 and “Portal Overview” on page 92 use the tabs at the top of the page.

Group Overview

The Group Overview page shows a table of availability information for each server in a server group.

To view the Group Overview page, from the top navigation, click **Availability > Systems Overview > Group**.

To change the group, select the new group in the **Servers In** pull-down control. ("ALL" is a group consisting of all servers communicating to the managing server).

For every server in the group, the table shows:

- A "traffic-light" indicator. Green means good availability, orange means problematic availability, and red means an alert. The indicator is set based on comparing the current response time with a baseline time determined from the average for a certain time.
- Server name
- Status of the server
- Total volume of requests processed by this server in the last hour
- A graph showing throughput (requests per 5 minutes) for the last hour.
- A graph showing the average response times, in milliseconds, over every 5 minutes for the last hour.

To see detailed information for a server, click the server name. The “Server Overview” page opens.

To set the Group Overview as the default page when you open the visualization engine, click **Set as My Default page** at the upper right of the page.

To switch to the “Enterprise Overview” on page 90,“Server Overview” “Alerts and Events” on page 108,“Problem Center” on page 110 and “Portal Overview” on page 92 use the tabs at the top of the page.

Server Overview

The Server Overview page displays information and activity graphs for a server.

To view the Server Overview page, from the top navigation, click **Availability > Systems Overview > Server**.

To select a server for displaying information, select the group and server names from the pull-down controls at the top of the page.

The following information is displayed for the server under Server Information:

- Server name.
- Group name.
- Platform (operating system).
- IP address.
- Start time.
- Current monitoring level for this server.
- Number of problems detected on this server.

The following information is displayed for the server under Server Statistics:

- A graph showing JVM CPU utilization for the last hour.
- A graph showing JVM memory utilization for the last hour.
- The throughput (total volume of requests) for the last hour.
- The uptime of the server, that is, the time since it was last rebooted.
- The names of the applications currently running on the server.
- Platform (operating system).
- IP address.
- Start time.
- Current monitoring level for this server.
- Number of problems detected on this server.

The following information is displayed for the server under Activity (Last Hour):

- A graph showing average response time, in milliseconds, for every minute in the last hour.
- A graph showing the throughput, or number of requests, for every minute in the last hour.
- A graph showing the number of open sessions, or logged on users, for every minute in the last hour.

To switch to the "Runtime Environment Check" on page 159,"Runtime Environment Comparison" on page 159 and "Server statistics overview" on page 97 for this server, mouse over the tool button at Server Information and select the needed link.

To switch to following items for this server, mouse over the tool button at Activity and select the needed link

- "Server statistics overview" on page 97
- "Server Activity Display" on page 116
- "Memory Analysis" on page 134
- "Heap Analysis" on page 135
- "Memory Leak" on page 135
- "System resources" on page 101

To view WLM Associated Service Classes for a z/OS server, mouse over the tool button at Server Information and select **Workload Management**. The "WLM associated service class summary" on page 95 page opens.

To switch to the "Enterprise Overview" on page 90,"Group Overview" on page 90"Alerts and Events" on page 108"Problem Center" on page 110 and "Portal Overview" use the tabs at the top of the page.

Portal Overview

The Portal Overview page shows availability information specific for IBM WebSphere Portal Server. You can access detailed portal statistics from this page.

To view the Portal Overview page, from the top navigation, click **Availability > Systems Overview > Portal**.

To select a server for displaying information, select the group and server names from the pull-down controls at the top of the page. Only servers running IBM WebSphere Portal Server are listed.

The page shows graphs of average response time, in milliseconds, for every minute in the last hour for the following portal components:

- Portal Pages and Gateway servlets
- Portlets
- Model Building
- Page Loading
- Authentication
- Authorization

To view detailed statistics (response time and access count) for portal pages, click the **Portal Pages/Gateway Servlet** link. You see statistics for the pages that were

slowest and most popular in the last hour; to view statistics for all pages, click **View All Portal Pages** at the bottom of the table. See "Viewing the Portal Page Summary."

To view detailed statistics (response time and access count) for portlets, click the **Portlets** link. You see statistics for the portlets that were slowest and most popular in the last hour; to view statistics for all pages, click **View All Portlets** at the bottom of the table. See "Viewing the Portlet Summary" on page 94.

To switch to the "Enterprise Overview" on page 90,"Group Overview" on page 90"Server Overview" on page 91"Alerts and Events" on page 108 and "Problem Center" on page 110 use the tabs at the top of the page.

Viewing the Portal Page Summary

About this task

The Portal Page Summary offers a view of the portals in your system and how they are operating. You can monitor the status of your portals from the slowest portals to the most popular portals for the last hour. In addition, view the metrics for the portals including Average Response Time and Count for authentication and authorization, and credential and content access metrics as well.

To open the Portal Page Summary for the slowest and most popular portals:

Procedure

1. From the top navigation, click **Availability > Systems Overview > Portal**.
2. Select a group and a server from the lists. The Portal Overview page displays the portal trends for the last hour.
3. Click the Portal Pages link. The portal page summary data for the slowest and the most popular portals displays.

Results

To open the Portal Page Summary for all portals from the last hour:

1. From the top navigation, click **Availability > Systems Overview > Portal**.
2. Select a group and a server from the lists. The Portal Overview page displays the portal trends for the last hour.
3. Click the Portal Pages link. The portal page summary data for the slowest and the most popular portals displays.
4. The portal page summary data for all the portals from the last hour displays.
5. Click the View all Portal Pages link.

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

"Portal Overview" on page 92
The Portal Overview page shows availability information specific for IBM WebSphere Portal Server. You can access detailed portal statistics from this page.

Viewing the Portlet Summary

About this task

The Portlet Summary offers a view of the portlets in your system and how they are operating. You can monitor the status of your portlets from the slowest portlets to the most popular portlets for the last hour. In addition, view the metrics for the portlets including Average Response Time and Count for authentication and authorization, and credential and content access metrics as well.

To open the Portlet Summary page for the slowest and most popular portlets:

Procedure

1. From the top navigation, click **Availability > Systems Overview > Portal**.
2. Select a group and a server from the lists. The Portal Overview page displays the response times for the portals.
3. Click the Portlets link. The Portlet Summary page displays the slowest and the most popular portlets.

Results

To open the Portlet Summary page for all the portlets for the last hour:

1. From the top navigation, click **Availability > Systems Overview > Portal**.
2. Select a group and a server from the lists. The Portal Overview page displays the portal trends for the last hour.
3. Click the Portlets link. The Portlet Summary page displays the slowest and the most popular portlets.
4. Click the View All Portlets link. The Portlet Summary page displays all the portlets for the last hour.

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

"Portal Overview" on page 92
The Portal Overview page shows availability information specific for IBM WebSphere Portal Server. You can access detailed portal statistics from this page.

WLM associated service class summary

About this task

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.

This feature is only available for z/OS servers.

To open the WLM Associated Service Class Summary:

Procedure

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 lists. The Server Overview page opens displaying data for the selected server.
3. Click **Workload Management** from the tools button at Server Information. The WLM Associated Service Class Summary page opens.

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

WLM associated service class period detail

About this task

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 delay detail information about each subsystem work manager.

To open the WLM Associated Service Class Period Details page:

Procedure

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 lists. The Server Overview page opens displaying data for the selected server.
3. Click **Workload Management** 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.

Results

To view the details for one of the associated service class periods, click its name. The "WLM associated service class period detail" on page 95 page opens.

To view the WLM Enclave, click the **Enclave** tab. The WLM Enclave page opens. See "Viewing a WLM enclave."

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

Viewing a WLM enclave

About this task

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.

For z/OS 1.2, all tokens in the Enclave are shown. There is no filtering based on 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:

Procedure

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 lists. The Server Overview page opens displaying data for the selected server.
3. Click **Workload Management** from the tools button. The WLM Associated Service Class Summary page opens.
4. Click the **Enclave** tab. The WLM Enclave page opens.

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

Server statistics overview

The Server Statistics Overview page shows a configurable set of application server level statistics. You can show the statistics for several servers on one page, and access other pages to view additional details for any server.

To view the Server Statistics Overview page, from the top navigation, click **Availability > Server Statistics Overview**.

To view statistics for an application server, select the server group and server name from the **Server Selection** pull-down controls, and click the **Add Server(s)** button. You can add more than one server; every server is represented by a line in the table. You can also select All Servers in the server name control to view information for all servers in a server group.

To remove a server from the page, click the **X** icon next to the server name.

Important: For servers running the z/OS platform, when you remove a server instance from the Server Statistics Overview 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.

To remove all servers from the page, click **Clear All** at the bottom of the list.

For every server, the table shows the configured statistics.

The statistics are refreshed periodically. To pause refreshing, click the **Pause** button. To resume refreshing (if it was paused) and force an immediate refresh, click the **Refresh** button.

To configure the statistics in the table, click **Customize...**. See "Configuring the Server Statistics Overview page" on page 98.

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 critical icon representing the *unavailable* status. The support team determines the unavailable server needs to be restarted, which returns the system to full functionality.

Scenario 2: Monitoring proactively

As the administrator of production systems, you have set appropriate thresholds for the fields that are 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 processor usage. You can now increase memory, add servers, or take some similar course of action to keep production running smoothly.

Configuring the Server Statistics Overview page

About this task

Configure the Server Statistics Overview page by selecting the resources to display on the detail page. In addition, set the warning threshold for certain resources by selecting the desired function from the list.

To configure the Server Statistics Overview page:

Procedure

1. From the top navigation, click **Availability > Server Statistics Overview**. The Server Statistics Overview page opens.
2. Click **Customize**. The Server Statistics Configuration window opens.
3. Click **Select All** or select the individual check boxes to select the resource that you want to display.
4. To show a warning (color the table cell yellow) when a statistic is above or below a certain threshold, select an operator from the list, and enter the threshold limit in the field next to it.
5. Click **Save**. For each data element on the Server Statistics Configuration page, set the range from 0 - 99999.

Results

You can select the following statistics:

- Volume Delta.
- Total Volume.
- Group name: the name of the group to which the server belongs
- Uptime: the time since the server was last rebooted
- Start Time: the time when server monitoring was started; this is usually the time the server was started
- Paging Rate
- Total CPU%: the current CPU load on the server
- Platform: the operating system that the server runs
- Live Sessions: the number of sessions currently open on the server
- Platform CPU Delta: the amount of CPU time spent since the last sample
- Application Server Platform
- Volume Delta per Second (If this value is >.5, it is rounded up to 1.)
- JVM/Region CPU Delta
- JVM/Region CPU%
- IP address
- Average Response Time (1 min).
- Data Collector Uptime: the time since the Data Collector was started or restarted.
- JVM/Region (DSA, EDSA) Memory Usage.
- Delta Normal CP time: for z/OS servers, the amount of normal central processor time spent since the last sample.
- Delta zAAP time: for z/OS servers, the amount of zAAP (additional CPU used exclusively for Java applications including WebSphere; requires z/OS 1.6 or above) time spent since the last sample.

- Delta zAAP-eligible time on CP: for z/OS servers, the amount of normal central processor time since last sample that was spent on tasks eligible for a zAAP. This happens when a zAAP is not present or busy. High values of this statistic can mean that an additional zAAP would be beneficial.
- Delta zAAP-eligible time: for z/OS servers, the amount of processor time (regardless of processor type) since last sample that was spent on tasks eligible for a zAAP. This is a sum of Delta zAAP time and Delta zAAP-eligible time on CP.

Related topics

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

Viewing the Server Statistics Overview

About this task

The Server Statistics Overview page provides application server-level statistics for quick assessment of server activity and related platform data. For thread data or system data on a server, or system data across a server group, click the tools button to view the information about the Server Activity Display page, the System Resources page, or the System Resources Comparison page respectively.

To open the Server Statistics Overview page:

Procedure

1. From the top navigation, click **Availability > Server Statistics Overview** or by selecting from the Tools button on the Server Overview page.
2. The Server Statistics Overview page opens. The zAAP (zSeries Application Assist processor) is a new hardware feature for zSeries. It is an additional CPU used exclusively for Java applications like WebSphere. zAAP requires z/OS 1.6 or above. ITCAM for WebSphere Application Server provides the following utilization statistics for the zAAP processors:
   - **Delta zAAP time** is the amount of CPU time spent on zAAP since the last sample.
   - **Delta Normal CP time** is the regular CP time for the address space.
   - **Delta zAAP-eligible time on CP** is the amount of time that could have been executed on zAAP but was not because it was already busy. High eligible time could mean that an additional zAAP processor could be needed.
   - **Delta zAAP-eligible time** is the total amount of time that could have been executed on zAAP, for example, delta zAAP + delta zAAP on CP.

Results

To remove a server from the Server Statistics Overview display:

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.

   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.

   Related topics

   Configuring the Server Statistics Overview

   Enterprise Overview
   The Enterprise Overview page shows a table of summary availability information for each server group.

Recent Activity Display

In the Recent Activity Display page, you can create a report that can help you to discover problems related to memory or other resources.

To create the report, see "Creating a Recent Activity report."

User Scenarios

Scenario 1: Evaluating the affect 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.

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

Creating a Recent Activity report

About this task

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

To create a Recent Activity report:

Procedure

1. From the top navigation, click **Availability > Recent Activity Display**. The Recent Activity Display page opens.
2. Select a group and a server from the lists.
3. For Metric 1 and Metric 2, select the two metrics that you want to compare from the lists. The available metrics are as follows:
   - # of Requests
   - Avg. Response Time
   - Live Sessions
   - System Paging Rate
   - JVM CPU
   - JVM Heap Size
   - # of GCs

- Total GC Time
- Avg. Heap Size after GC

Note: The Garbage Collection options are not supported for either CICS or IMS.
When the date range is Last 48 Hours, the Avg. Response Time value equals the result of the summary of AVG_RES_TIME in this hour divided by the summary counts of AVG_RES_TIME in this hour. If no AVG_RES_TIME occurs within the hour then the result is 0. When the date range is Last 60 Minutes, the Avg. Response Time value equals the result of the summary of AVG_RES_TIME in this minutes divided by the summary count of AVG_RES_TIME in this minute. If no AVG_RES_TIME occurs within the minute then the result is 0.

4. For Time, select the time when you want the system to extract the data.
5. Click the **OK** button. A new report displays based on your selection. You can click the bars in the graph for further details.
6. Using the Recent Activity Options, you can select a different group or server, compare two different metrics, or view a different time increment.

Results

Either heap size or garbage collection can cause a slow down in the performance of your server. Check if the 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 on the throughput. Analyze the memory in your system using the Recent Activity Display and then make the necessary adjustments.

Related topics

Creating a Memory Analysis report

Creating a Memory Leak Candidate Finder report

Setting up a Heap Analysis

System resources

The System Resources pages provide views of system, resource metrics available for an application server. You can use this information to tune the application server.

Select the group and server names to view the System Resources Browser. See "Viewing the System Resources Browser" on page 102.

This feature is not available for CICS or IMS.

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 JCA Connection Pools. If any pool is at or near 100%, it is likely that demand for application A is saturating those resources. You might be able to fix the problem by creating more or larger pools.

Viewing the System Resources Browser

About this task

The System Resources Browser displays summary information for all the resources on the selected application server. You can view data, such as EJBs, Database/JCA Connection Pools, Servlet/Session Manager, Thread Pools, JTA Transactions, Web Applications, SQL Data, JCA-CICS, ORB, and JVM/System.

To open the System Resources Browser:

Procedure

1. From the top navigation, click **Availability > System Resources**. The System Resources Overview page opens. The right side of the page is the System Resources Overview page while the left side is the navigation for the System Resources Browser. You can also access this page from the toolbox icon  on the Server Activity Display page.
2. Select a group and a server from the list on the left navigation panel. The System Resources Browser opens displaying the information for the selected group and server. If you access this page from the Tivoli Enterprise Portal and from the Server Activity Display page, the server group name and server name display automatically.
3. Point to the question mark icon next to each of the following items: EJB Activity, Servlet JSP Activity EJB Coverage, and Servlet JSP Coverage for more information.

Results

Use the Resource Categories left navigation pane to select the category of resources that you want to view. For the available resource categories, see "Resources Performance Metrics" on page 103.

To change the application server:

1. On the left navigation pane, select a group from the list.
2. On the left navigation pane, select a server from the list.

Note: In the Current Snapshot view, if more that 50 objects are retrieved, then sorting is disabled. You can view 5, 10 or 15 rows per page.

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.

"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

Resources Performance Metrics

About this task

You can drill down into different pages of the resources and click on subcategory links (for example, individual rows of a table) to view detailed information. The following information provides the metrics for various metric categories and the application servers that support them.

WebSphere Application Server 6.x and 7, WebSphere Portal Server 6.x and 7

- Servlet
- JVM
- J2EE Domain
- J2EE Server
- J2EE Application
- EJB Module
- Web Module
- RAR Module
- Entity Bean
- Stateful Session Bean
- Stateless Session Bean
- Message Driven Bean
- Resource Adapter
- JDBC Resource
- JDBC Data Source
- JDBC Driver
- JCA Resource
- JCA Connection Factory
- JCA Managed Connection Factory
- JNDI Resource
- JMS Resource
- JTA Resource
- RMI IIOP Resource
- Thread Pool
- ORB
- Dynamic Cache
- Servlet Session Manager
- Transaction Service
- Security Authentication (WebSphere v7 only)
- Security Authorization (WebSphere v7 only)
- SIP Container Module
- SCA Model Object Tree (WebSphere v7 only)
- High Availability Manager Module
- System Module
- J2C Module
- Web Services Module
- Workload Manager Module

- Web Service Gateway Module
- Object Pool Module
- Alarm Manager Module
- Scheduler Module
- DCS Statistics Module
- Connection Pool Module
- SQL

WebSphere ESB (Enterprise Service Bus) 6.x and 7

All metrics for WebSphere Application Server, and also:
- **SIB (System Integration Bus)**
  - MessageStore
  - Mediation Framework
  - Message Processor
  - Communications

WebSphere Process Server 6.x and 7

All metrics for WebSphere Application Server, and also:
- WebSphere Business Integration (WBI) (version 6.x only)
- **SIB (System Integration Bus)**
  - MessageStore
  - Mediation Framework
  - Message Processor
  - Communications

J2EE

WebLogic 8
- Web module
- Servlet
- Entity EJB
- Stateless EJB
- Stateful EJB
- Message Driven EJB
- Message Driven EJB Destination
- JDBC Connection Pool Resource
- JCA Connection Pool
- JMS Session
- JTA Resources
- JVM System
- JVM System Server
- Execute Queue
- Execute Queue Runtime

WebLogic 9.1
- Web module

- Servlet
- Entity EJB
- Stateless EJB
- Stateful EJB
- Message Driven EJB
- Message Driven EJB Destination
- JDBC Connection Pool Resource
- JCA Connection Pool
- JMS Session
- JTA Resources
- JVM System Server
- Execute Queue
- Thread Pool

JBOSS 3 and 4
- Web module
- Servlet
- EJB Module
- Entity EJB
- Stateless EJB
- Stateful EJB
- Message Driven EJB
- JDBC Connection Pool
- JCA Connection Pool
- JMS Session
- JTA Resources
- JVM/System

Tomcat 5.0 and 5.5
- Web module
- Servlet
- Cache
- Thread Pool
- Session Manager

Oracle Application Server 9.x
- Operating System

Oracle Application Server 10.x
- Web module
- Servlet
- EJB Module
- Entity EJB
- Stateful Session Bean
- Stateless Session Bean
- Message Driven Bean

- JDBC Data Source
- JVM

Netweaver 6
- Component Performance
- Request Performance
- Performance Summary
- Thread Pool
- Web Container
- Entity EJB
- Stateless EJB
- Stateful EJB
- Message EJB
- Transaction
- Memory
- JVM
- System
- Web Service Performance
- Web Service Request
- HTTP

WebSphere Community Edition
- Web Module
- Servlet
- EJB Module
- Entity Bean
- JCA Connection
- JVM

J2SE - IBM and Sun
- OS
- Memory
- JVM
- Threading

General

The following performance metrics are supported on any application server which supports the underlying call interface. The following categories are available:
- CICS transactions
- Queue manager
- SQL

Related topics

Configuring the Enterprise Overview

"Enterprise Overview" on page 90
The Enterprise Overview page shows a table of summary availability information for each server group.

"Group Overview" on page 90
The Group Overview page shows a table of availability information for each server in a server group.
"Server Overview" on page 91
The Server Overview page displays information and activity graphs for a server.

SMF data

The SMF data pages show the information collected by the System Management Facilities (SMF) on z/OS servers.

In these pages, you can view detailed information about Server, EJBs, Servlet Session Manager, Web Applications, and Server Regions. The source of the data comes primarily from the SMF records published periodically by WebSphere. As these records are published, the Application Monitor intercepts the transfer of the records and makes a copy in real time before writing it to the SMF dataset.

To view SMF data, select the group name and server name. (SMF data is only provided for z/OS servers). See "Viewing SMF data."

User Scenarios

Scenario 1: Pinpointing problems

The Application A response time slows to unusable when the server experiences even moderate throughput, while resident time is idle. Using SMF data, view the percentage of threads used in the Database Connection Pools, Thread Pools, and JCA Connection Pools. This tells you if there is a bottleneck in the application server pools. Any pool that is at or near 100% is probably being overwhelmed by the requests of application A for those resources. To fix the problem, you can create more or larger pools.

Viewing SMF data

About this task

The following information provides the metrics for the SMF data type and the application servers it supports.

WebSphere 5 and 6

- Overview
- Server
- EJBs
- Servlet Session Manager
- Web Applications
- SQL
- JCA-CICS

z/WebSphere only

The following information provides the metrics for the z/WAS only data type and the application servers it supports.

WebSphere 5

Server Regions

To view the SMF data page:

Procedure

1. From the top navigation, click **Availability > SMF Data**. The SMF Data page opens.
2. Select a group and a server from the list on the left navigation panel. The SMF Data page opens displaying the information for the selected group and server. If you have come to this page directly from the Tivoli Enterprise Portal, the group name and server name display automatically.

   Related topics

   Viewing the Server Statistics Overview

Alerts and Events

In the Alerts and Events page, you can view the alerts and events generated by managing server traps and by the Tivoli Enterprise Portal, and escalate these events into the Problem Center for diagnosis and tracking.

To display ITCAM situations in Alerts and Events, complete the following setup:

1. Go to the MSVE host.
2. Open the file `$MS_HOME/etc/dal/dal.properties` set the correct values for the following properties:

   ```
   dal.itmsoap.hostName=
   dal.itmsoap.port=1920
   dal.itmsoap.useHttps=false
   ```

   For UNIX and Linux platforms, `$MS_HOME` is by default `/opt/IBM/itcam/WebSphere/MS`.

   For Windows platforms, `$MS_HOME` is by default `C:\Program Files\IBM\itcam\WebSphere\MS`.
3. Restart the MSVE server instance.

To view the Alerts and Events page, from the top navigation, click **Problem Determination > Alerts and Events**.

The page shows a table of all high-priority trap alerts and Tivoli Enterprise Portal events for the last 24 hours for a selected server.

To change the server, select the group and server names from the pull-down controls, and click the **OK** button.

To show only alerts or only events, select Alerts or Events from the **Status** control, and click the **OK** button.

To filter the displayed alerts and events by date and time, click **Show Advanced Filters**, check the **Date Range** box, then select the date and time range in the **From** and **To** controls and click **OK**.

For every alert or event, the table shows:

- Date and Time of the alert or event.
- Server group name.

- Server name.
- Name of alert or event.
- Origin of the event (trap or Tivoli Enterprise Portal).
- **Escalate** button.

To escalate an alert or event, adding it to the Problem Center, click the **Escalate** button. See "Escalating alerts and events to the Problem Center."

Escalating alerts and events to the Problem Center

About this task

When you need to diagnose an alert in order to resolve it, you can escalate it as a problem in the Problem Center. This allows you to track the data related to the alert and provides further information that can help you resolve the issue. If you escalate an alert to a problem, your actions are recorded in the audit trail.

To open the Alerts and Events page:

Procedure

1. From the top navigation, click **Problem Determination > Alerts and Events**. The Alerts and Events page opens.
2. Select a group and a server from the list. This limits the alerts to the server and group you specify.
3. To escalate an alert to a problem for tracking, click **Escalate**.
4. Type a description of the problem in the text box. (A description is not required.)
5. Select the Category from the list. The available categories are as follows:
   - **Unknown**–The category could not be determined by the system.
   - **Application Performance**–Under capacity, hanging, or waiting incomplete transaction, CPU hogging, poor load balancing, slow completed transaction, slow JDBC call, slow, LDAP look-up, slow JMS call, and slow method call.
   - **Application Outage**–exception, incorrect output, intermittent outage, binaries discrepancies within a group/cluster, and properties discrepancies across a group/cluster.
   - **Resource Consumption**–JDBC connection leak, JMS connection leak, heap usage, heap/object lead, excessive JVM CPU%, high garbage collection frequency, high garbage collection time, and fragmentation.
6. Click **OK** to save as a problem. The problem displays in the Problem Center.
7. Click the Problem Center tab to view your problem.

Results

Using the advanced filters:

1. Click the **Show Advanced Filters** link.
2. Click the check box to set the advanced filters.
3. Select the dates and times that you want to use to limit your list of problems.
4. Click **OK**. The list displays based on the filtering you selected.

Related topics

"Problem Center"
The Problem Center page shows problems that were previously escalated from actions or events or added manually. You can investigate problems, close them, and add new problems.

Problem Center

The Problem Center page shows problems that were previously escalated from actions or events or added manually. You can investigate problems, close them, and add new problems.

To view the Problem Center, from the top navigation, click **Problem Determination > Problem Center**.

Escalated problems are normally high-priority trap alerts and Tivoli Enterprise Portal events, escalated from the "Alerts and Events" on page 108 page. The details of each problem, including a snapshot of the problem details and the state of the application server at the time the problem occurred, are available for review.

A problem has a *status* value:

- New: the status assigned by the system to newly created problems.
- Open: the status assigned by the system after a user opens a problem.
- Closed: the status assigned by the user when the user closes a problem. Closed problems are deleted from the system after 30 days.

Initially, all problems are shown in the Problem Center table.

To show only the problems for a particular server, select the group and server names from the **Filter** controls, and click the **OK** button.

To show only the problems with a particular status, select the status from the **status** control, and click the **OK** button.

To show only problems that occurred in a limited date/time range, click **Show Advanced Filters**, check the **Date Range** box. Then, select the date and time range in the **From** and **To** controls and click **OK**.

For every problem, the table shows:

- Date and Time the problem occurred.
- Server group name.
- Server name.
- Problem category, set when the problem was escalated. See "Escalating alerts and events to the Problem Center" on page 109.
- Origin of the problem (trap, Tivoli Enterprise Portal, or manual creation).
- Name of the problem.
- Description of the problem, entered when the problem was escalated. See "Escalating alerts and events to the Problem Center" on page 109.
- Closing comment, entered when the problem is closed. The field is empty for new or open problems.
- Problem status (New, Open, or Closed).
- **Delete** button.

To view the details of a problem, click its date/time. See "Viewing the details of a problem."

To close a problem, select Closed in the **Problem status** control in the table row. See "Closing a problem" on page 114.

To delete a problem, click the **Delete** button.

To add a problem manually, click the **Add Problem Manually** button at the bottom of the table. See "Adding a problem manually" on page 113.

Viewing the details of a problem

About this task

The Problem Center displays a list of all the high-priority trap alerts and Tivoli Enterprise Portal events that were escalated to problems as well as problems entered manually. To view additional details about each problem, select the Date/Time link for that problem. Doing this allows you to access the tabs that furnish further diagnostic information.

To view the details of a problem:

Procedure

1. From the top navigation, click **Problem Determination > Problem Center**. The Problem Center opens.
2. Select the Date/Time link for the problem. The details for the problem are displayed on the Problem tab. Several tabs provide further problem details including memory, transactions, resources, traps, logs, and configuration. The information provided by each tab is useful in diagnosing the problem. The function of each tab is described in the following section:
   - **Problem**: Displays an analysis based on the user-defined category of the problem that was assigned when the problem was escalated from Alerts and Events. Each category contains a set of contributing factors. The system analyzes whether there was a positive result or not and suggest a solution based on the data. You can change the status of a problem, edit the problem description, and change the problem category.
   - **Memory**: Displays information related to the following things:
     - The event or alert
     - The identification for the server
     - The status of the server
     - The filter to display previous hours of data (1, 2, 6, 12, 24, or 48 hours)
     - The actions to perform (Heap dump and Thread dump)
     - A snapshot of the current memory usage information
     - A graph for the Average Heap Usage after Garbage Collection (GC
     - Trends for the Heap Size, Number of Requests, Number of Sessions, Response time, Number of GCs, CPU usage, GC time, and Paging rate
   - **Transactions**: Displays information related to the following things:
     - The event or alert
     - The identification for the server
     - The status of the server
     - The filter to display previous hours of data (1, 2, 6, 12, 24, or 48 hours)

- The action to perform (Thread dump)
- A snapshot of the current transaction usage information
- Trends for the Response time, JVM CPU Usage, Number of Sessions, Number of Requests, and Heap Usage
- Rankings of the slowest transactions in a list
- Transaction snapshots of all the transactions

- **Resources** - Displays information related to the event or alert, identification for the server, and opens the JMX browser page. In the Current Snapshot table, the list displays based on the filtering you select. If more that 50 objects are retrieved, then sorting is disabled. You can view 5, 10 or 15 rows per page.

  Complete the following setup steps to see data in the Tivoli Enterprise Monitoring Agent data snapshot section.

  a. Go to the MSVE host.
  b. Open the f`$MS_HOME/etc/dal/dal.properties` file and set the correct values for the following properties:

     ```
     dal.itmsoap.hostName=
     dal.itmsoap.port=1920
     dal.itmsoap.useHttps=false
     ```

     For UNIX and Linux systems, `$MS_HOME` is by default `/opt/IBM/itcam/WebSphere/MS`.

     For Windows systems, `$MS_HOME` is by default `C:\Program Files\IBM\itcam\WebSphere\MS`.
  c. Restart the MSVE server instance.

  Monitoring Console - the Monitoring Console button launches the Tivoli Enterprise Portal browser client from the MSVE. Complete the following steps to enable this feature:

  a. Go to the MSVE host.
  b. Open the `$MS_HOME/etc/ve.properties` file and set the correct values for the following properties:

     ```
     tep.hostname=
     tep.port=1920
     tep.baseurl=///cnp/kdh/lib/cnp.html
     tep.userid=
     ```

     For UNIX and Linux platforms, `$MS_HOME` is by default `/opt/IBM/itcam/WebSphere/MS`.

     For Windows platforms, `$MS_HOME` is by default `C:\Program Files\IBM\itcam\WebSphere\MS`.
  c. Restart the MSVE server instance.

- **Traps**: Displays information related to the event or alert, identification for the server, action to perform (New Trap), and Trap Action History, which provides the date the trap occurred, name of the trap, server name, severity, and the action taken. From the **Action Taken** button, you can access the results of the action. For example, if a method trace was taken, you can click this link to go to the properties page for this method trace. In addition, you can delete the history of a trap you no longer need.
- **Logs**: Displays information related to the event or alert, identification for the server, filter to display by entry type (All, Warning, or Error), and a scrape of the log files.

Viewing log data. Data under the Logs tab comes from the ITCAM Tivoli Enterprise Monitoring Agent agents. Complete the following setup steps to view this data:

a. Go to the MSVE host.

b. Open the `$MS_HOME/etc/dal/dal.properties` file and set the correct values for the following properties:

```
dal.itmsoap.hostName=

dal.itmsoap.port=1920

dal.itmsoap.useHttps=false

tep.userid=
```

For UNIX and Linux systems, `$MS_HOME` is by default `/opt/IBM/itcam/WebSphere/MS`.

For Windows , `$MS_HOME` is by default `C:\Program Files\IBM\itcam\WebSphere\MS`.

c. Restart the MSVE server instance.

- **Configuration** - displays information related to the event or alert, identification for the server, and a filter to display previous hours of data (1, 2, 6, 12, 24, or 48 hours).

3. You can add a description, change the problem status, or change the category for a problem by clicking the **Edit** button in the Event box. See the Contributing Factors section for more information about the problem.

Related topics

Adding a problem manually

Closing a problem

Adding a problem manually

About this task

You might have to add a problem to the Problem Center manually, for example, if your monitoring software does not interface with ITCAM or if you discover a problem that does not have a trap associated with it. You can do this in the Problem Center using the **Add Problem Manually** button. As a result, you can access the problem details after creating the problem.

To add a problem manually:

Procedure

1. From the top navigation, click **Problem Determination > Problem Center**. The Problem Center opens.
2. At the bottom of the page, click the **Add Problem Manually** button.
3. Select the group and server where the problem exists from the lists.
4. Select the date from the list and enter a time in the text box.
5. Enter a name and description for the problem in the text box.
6. Select the Category from the list. The available categories are as follows:
   - **Unknown**–The category could not be determined by the system.
   - **Application Performance**–Under capacity, hanging, or waiting incomplete transaction, CPU hogging, poor load balancing, slow completed transaction, slow JDBC call, slow, LDAP look-up, slow JMS call, and slow method call.

- **Application Outage**–Exception, incorrect output, intermittent outage, binaries discrepancies within a group/cluster, and properties discrepancies across a group/cluster.
- **Resource Consumption**–JDBC connection leak, JMS connection leak, heap usage, heap/object leak, excessive JVM CPU%, high garbage collection frequency, high garbage collection time, and fragmentation.

7. Click **OK** to save as a problem. The problem displays in the list in the Problem Center with the status as new.
8. Additional details on the problem are available when you select the Date/Time link for the problem. See "Viewing the details of a problem" on page 111 for more information.

Related topics

"Problem Center" on page 110
The Problem Center page shows problems that were previously escalated from actions or events or added manually. You can investigate problems, close them, and add new problems.

Viewing the details of a problem

Closing a problem

Closing a problem

About this task

Close a problem in the Problem Center when the issue is resolved. After you close a problem, you are given the option to delete it from the Problem Center. If you elect not to delete a problem, your closed problem will be deleted from the system after 30 days.

To close a problem:

Procedure

1. From the top navigation, click **Problem Determination > Problem Center**. The Problem Center opens.
2. View all Open problems by selecting Open from the Status list and click **OK**. Find the problem that you want to close.
3. To change the status of an open problem to closed, select Closed from the box in the Status column. A Closing Comment text box opens.
4. Add your final notes to the problem in the Enter Closing Comment text box and click **OK**. Your notes display in the Closing Comment column and the problem status is changed to Closed.

Related topics

Adding a problem manually

Viewing the details of a problem

In-flight request search

In the In-flight request search page, you can search for requests that are currently open on an application server. As a request normally closes relatively quickly, this search is useful for locating hanging transactions, which result from an application malfunction.

To search for a request in a server group, select the group name in the **Group** control.

To search for a request in an individual server, select the group name in the **Group** control and the server name in the **Server** control.

If you have come to this page directly from the Tivoli Enterprise Portal, the group name and server name display automatically.

To search for a request that has a substring in the URL string (for Web requests) or class name (for EJB requests), enter the substring in the **Search Request/Transaction** control.

Click the **OK** button to perform the search and view the results. See "Searching for an Application Request."

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 high. You go to the In-flight Request Search to see if a transaction is present. It appears that 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

About this task

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 Thread/Task ID link of a request to view the Request Detail for that request. Click 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.

The In-flight Request Search is not case-sensitive.

To search for a request:

Procedure

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 box.

   Note: If you do not select a group or server, requests from all servers display.
3. Enter a string in the Search Request box.

Note: The system searches 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 displays those requests. In addition, if you leave the search request box empty, all active requests display.

4. Click **OK**. All the active requests associated with your search display in the order of descending Total Resident Time. To change the order, see "Sorting search results."

Related topics

Activating a thread

Canceling a request

Changing a thread's priority

E-mailing a PDF file - SAD

Exporting to a file - SAD

Searching a Method Trace

Suspending a thread

Viewing a Method/Component Trace - Flow View

Viewing a PDF file - SAD

Viewing a Stack Trace

Viewing Request detail

Viewing the request object and session object

Sorting search results

About this task

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

To sort the search results:

Procedure

1. Click a column heading to sort the results. You can sort only 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.

Related topics

Searching for an Application Request

Viewing a Composite Method Trace - In-flight Request Search

Viewing a Composite Request Detail - In-flight Request Search

Viewing a Composite Stack Trace - In-flight Request Search

Server Activity Display

The Server Activity Display page shows a table of request activity on an application server. This page can help you troubleshoot and fix hanging requests and evaluate the current performance of your applications.

You can access the Server Display using one of the following options:

- From the top navigation, click **Problem Determination > Server Activity Display** to display the Server Activity Display page.
- If you have logged in from the Tivoli Enterprise Portal the information about the sever group name and server name files is already complete.

For more information, see "Access the Managing Server Visualization Engine from Tivoli Enterprise Portal" on page 42.

To set the server for which the activity is displayed, select the server group name and server name.

To show requests currently being processed by the server, click the **Active Requests** tab. See "Server Activity Display - active requests."

To show requests recently processed by the server, click the **Recent Requests** tab. See "Server Activity Display - recent requests" on page 119.

To show requests that are hanging because they are waiting on a lock, click the **Lock Contentions** tab. See "Server Activity Display - lock contentions" on page 121.

User Scenarios

Scenario 1: Troubleshooting an application that hangs.

Several users of application Z have reported that they cannot 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 the average response time of a server 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 can 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 - active requests

About this task

The Active Requests tab displays thread data for an application server at a specific point in time.

Data in this section is constantly fluctuating. Active requests display a snapshot of the data at a specific point in time. As a result, requests can be completed and disappear from the display on refresh, or by the time you drill down.

To open the Server Activity Display (active requests) page:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the box. If you have come to this page directly from the Tivoli Enterprise Portal, the group name and server name display automatically.

Results

The Active Requests tab has three main sections

- Server Information
- Recent Activity (Last Minute)
- Active Requests (In JVM Memory Now)

Server Information provides details relating to the server. It also has a toolbox icon  . You can click this toolbox icon to access direct links to the following features. The feature you select displays information specific to the group sever and server name you select.

- JVM Display
- System Resources
- Monitoring On Demand
- Data Collector Profiles
- Trap and Alert Management

Recent Activity (Last Minute) displays the following features. You can click a link to create a Recent Activity Report for the item that you want to select.

- JVM CPU
- JVM Heap Size (MB)
- # of Requests
- # of live Sessions
- Average Response Time (MS)

Active Requests (In JVM Memory Now) You can use this section of the page to view information about thread types and to view request details.

To filter the active requests data:

1. You can filter active request by selecting the options available on the **Thread Type** and **Thread Status** drop-down menus.
2. Click **Thread Type**, select one of the following options from the drop-down menu:
   - Any
   - EJB
   - Servlet
   - JSP
   - SCA
   - CICS
3. Click **Thread Status**, select one of the following options from the drop-down menu:
   - Any

- Active
- Suspended
- Waiting

4. Click **Refresh**. The active requests data displays based on the selected filter.

To sort the active requests data:

1. Click a heading link:
   - Client Requests
   - Client Requests 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

   The data refreshes sorted by the selected heading.
2. Click the heading link a second time to invert the sorting.

To view request detail, click the link in the **Client Requests** column. From the request detail view, you can investigate further details, suspend and reactivate the request thread, and change its priority.

Related topics

Activating a thread

Canceling a request

Server Activity Display - recent requests

Viewing the request detail

Server Activity Display - recent requests

About this task

The Recent Requests page displays list of the recently completed requests for the user to review the recent activity data on a per-server basis. The default maximum number of recent activity data is 100. The maximum number of recent activity data applies to each server. When the queue is full, the newest request data replaces the oldest data.

To open the Server Activity Display (recent requests) page:

If you have logged in from the Tivoli Enterprise Portal page, all the relevant information including the sever group name and server name display automatically.

Procedure

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

2. Select a group and a server from the lists. If you have come to this page directly from the Tivoli Enterprise Portal, the group name and server name display automatically. 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.

Results

The Recent Requests tab has three main sections

- Server Information
- Recent Activity (Last Minute)
- Recent Requests (Last 100 Completed Requests)

Server Information provides details relating to the server. It also has a toolbox icon  . You can click this toolbox icon to access direct links to the following features. The feature you select displays information specific to the group sever and server name you select.

- JVM Display
- System Resources
- Monitoring On Demand
- Data Collector Profiles
- Trap and Alert Management

Recent Activity (Last Minute) displays the following features. You can click a link to create a Recent Activity Report for the selected item.

- JVM CPU
- JVM Heap Size (MB)
- # of Requests
- # of live Sessions
- Average Response Time (MS)

Recent Requests (Last 100 Completed Requests) displays completed requests you can filter this information by Thread Type. **To filter the recent request data:**

1. You can filter recent requests by selecting the options available on the **Thread Type** drop-down menu and then clicking Refresh to display the results.
2. Click **Thread Type**, select one of the following options from the drop-down menu:
   - Any
   - EJB
   - Servlet
   - JSP
   - SCA
   - CICS
3. Click **Refresh**. The active requests data displays based on the selected filter.

To sort the recent request data:

1. Click a heading link:
   - Client Requests

- Client Requests Start
- Response Time (ms)
- Accumulated CPU (ms)
- Idle Time (ms)
- Thread Type
- User ID

2. The data refreshes sorted by the selected heading.

To view request detail, click the link in the **Client Requests** column.

Related topics

Activating a thread

Server Activity Display - active requests

Viewing the request detail

Server Activity Display - lock contentions

About this task

Use the Lock Contentions tab when a request is taking too long to process and you want to know why. The Lock Contentions tab displays any requests that are hanging because they are waiting on a lock. The data shows both the object that has the lock and the one that is waiting for a lock.

To open the Server Activity Display (lock contentions) page:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the box. The Server Activity Display (active requests) page opens. The information for the selected server group displays. If you have come to this page directly from the Tivoli Enterprise Portal, the group name and server name display automatically.
3. Click **Lock Contentions**. The Lock Contentions tab opens displaying the active locks. If no locks have occurred, the system message says, "There are no classes instrumented for lock analysis."

Results

If there are any **Active Locks (For active requests)**, the information displays under the following column headings:

- Locked Object Class
- Owners Request/Transaction Name
- Owners Request/Transaction Type
- Owner Class
- Owner Method
- Waiting Time (ms)
- Waiting Class
- Waiting Method
- Waiting Request/Transaction Name
- Waiting Workload Type

Related topics

Activating a thread

Canceling a request

Server Activity Display - active requests

Server Activity Display - recent requests

Viewing the request detail

Viewing request detail

About this task

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

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

To open the Request Detail page:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (active requests) page opens. The information for the selected server group displays unfiltered.
3. Click the link in the Client Requests column.
4. The Request Detail page for that request opens. This page displays data for that request only.

Results

If the request is not yet completed, you can perform the following additional actions on it:

- To suspend the request thread, select Suspend from the **Change Thread Status** control. See "Suspending a thread" on page 123.
- To reactivate a suspended thread, select Active from the **Change Thread Status** control. See "Activating a thread" on page 123.
- To cancel the request, click the **Cancel Request** button. See "Canceling a request" on page 124.
- To change the priority of the request thread, select a priority from the **Change Priority** control. See "Changing the priority of a thread" on page 124.
- To view a Stack Trace, showing which methods were called to reach the current state of the request thread, click **Stack Trace** on the left navigation pane. See "Viewing a Stack Trace" on page 125.
- To view a request object and session object, click **Request/Session Object** on the left navigation pane. See "Viewing a request object and session object" on page 126.
- To view a Method/Component Trace, click **Method/Component Trace** on the left navigation pane. From this view, you can search the trace and export it to a file. See "Viewing a Method/Component Trace - flow view" on page 125.

Related topics

Canceling a request

Changing a thread's priority

Suspending a thread

About this task

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

To suspend a thread:

Procedure

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

Results

When suspending a thread, there is a danger that the request might hold database locks or system resources. After you suspend the request, any other requests that require the removal of those locks or monitors are also suspended. Any locks in the application server and database server are not released after the system suspends a thread. This can cause other applications to fail or hang.

Related topics

Activating a thread

Canceling a request

Changing a thread's priority

Activating a thread

About this task

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

To activate a thread:

Procedure

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

Related topics

Canceling a request

Changing a thread priority

Suspending a thread

Canceling a request

About this task

If an application request from the system is looping or abusing resources, it might be necessary to cancel the request. This terminates the request by throwing a runtime exception.

To cancel a request:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Requests column of the request that is hanging. A Request Detail page opens, where you can click the Cancel Request button.
4. Click **Cancel Request**. A confirmation box displays.
5. At the confirmation box, click **OK**. Canceling a thread can cause JVM and application server instability. Use the Cancel Thread function only when necessary, and with careful consideration of the consequences.

Related topics

Activating a thread

Changing a thread's priority

Suspending a thread

Changing the priority of a thread

About this task

If a thread is executing too slowly, you can increase the priority of a thread. This moves the thread up in the stack so it executes more quickly. (Alternatively, you can decrease the priority of a thread to allow other threads to execute more quickly.)

To change the priority of a thread:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Request column. The Request Detail page for that request opens.
4. From the Change Priority box, select a priority. Priority 1 is the lowest and priority 10 is the highest.
5. Click **Save**.

Related topics

Activating a thread

Canceling a request

Suspending a thread

Viewing a request detail

Viewing a Stack Trace

Viewing a Stack Trace

About this task

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:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Request column. The Request Detail page for that request opens.
4. Click **Stack Trace** from the left navigation pane. The Stack Trace page opens. The most recently executed method displays first in the Stack Trace.

Related topics

Canceling a request

Changing a thread's priority

Viewing a Method/Component Trace - flow view

About this task

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

To view the flow view of a Method/Component Trace:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Request column. The Request Detail page for that request opens.
4. Click **Method/Component Trace** on the left navigation pane. The Method/Component Trace (flow view) page opens. The last executed method displays first in the Method/Component Trace.
5. Enter the Delta Elapsed Time and the Delta CPU Time value under the Threshold Highlighter table to highlight the data with the features that you want to view throughout the whole trace.

6. Click **Apply**. The Complete Flow View table displays the method flow list with the highlighted data that you selected to view.
7. Click **Reset to Default** for using the default threshold highlighter value, if necessary.
8. Click to select the number of rows of data that you want to view per page from the Pagination box. The Flow View tab refreshes displaying the number of rows of data you selected to view on each page.

Results

To export the method/component trace to a comma-delimited file, click the **Export to File** button. See "Exporting to a file: Server Activity Display (SAD)" on page 128.

To export the method/component trace to a PDF file and view or save this file, click the **View PDF** button. See "Viewing a PDF file: Server Activity Display (SAD)" on page 128.

To export the method/component trace to a PDF file and e-mail this file, click the **E-Mail** button. See "Sending a PDF file by email: Server Activity Display (SAD)" on page 127.

Related topics

Canceling a request

Changing a thread's priority

Viewing a request object and session object

About this task

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

To view the request object and session object trace:

Procedure

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 lists. The Server Activity Display page opens.
3. Click the link in the Client Request column. The Request Detail page for that thread opens.
4. Click **Request/Session Object** from the left navigation pane. The Request Object and Session Object page opens.

Related topics

Canceling a request

Changing a thread's priority

Searching a Method/Component Trace

About this task

Use the search to specify any of the columns that are available in the Flow View (Elapsed Time, CPU Time, Delta Elapsed Time, Delta CPU Time, Event Type, or Event Data), together with a numeric threshold (or a string,) and presents a list of

events from the method trace whose metrics cross the threshold (or match the string). The Event Type and Event Data searches are case-sensitive.

To search a Method/Component Trace:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display page opens.
2. Select a server from the Server box. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Request column. The Request Detail page for that request opens.
4. Click **Method/Component Trace** from the left navigation pane. The Component Trace/Method (Flow View) page opens. The method that was started last is displayed first in the Method Trace.
5. Click the **Search** tab. The Search page opens.
6. Enter the search criteria and the search value.
7. Click **Search**. The Method Trace page refreshes and displays the method trace that matches your search criteria and value. 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 20th method on the Flow View page, then clicking the Event Data link of the first result brings up the Flow View tab starting with the page that includes that 20th record.

   Related topics

   Viewing a Composite Method Trace - SAD

   Viewing a Method/Component Trace - Flow View

Sending a PDF file by email: Server Activity Display (SAD)

About this task

You can send a PDF file of the Method Trace/Component Trace by email to one or a group of the application monitor users. Recipients must have valid user accounts and proper permissions in order to view the report.

To send a PDF file by email:

Procedure

1. From the top navigation, click **Problem Determination >Server Activity Display**. The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (Active Requests) page opens.
3. To view the detail, click the link in the Client Requests column. The Request Detail page for that thread opens.
4. Click **Method/Component Trace**. The Method/Component Trace page opens.
5. Click **E-mail**. The email page opens.
6. Enter the email address of the recipient. Separate multiple addresses with a comma.
7. Click **OK**.

   Related topics

   Exporting to a file - SAD

Viewing a PDF file: Server Activity Display (SAD)

About this task

Before sending a PDF file, download the file to view it.

To view a PDF file:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display** . The Server Activity Display page opens.
2. Select a group and a server from the lists. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Requests column. The Request Detail page for that thread opens.
4. Click **Method/Component Trace**. The Method/Component Trace page opens.
5. Click the **View PDF** button.
6. From the File Download window, click either **Open** to view the file immediately or click **Save** to download the file.

Related topics

Exporting to a file - SAD

E-mailing a PDF file - SAD

Exporting to a file: Server Activity Display (SAD)

About this task

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

To export to a file:

Procedure

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 lists. The Server Activity Display (Active Requests) page opens.
3. Click the link in the Client Requests column. The Request Detail page for that thread opens.
4. Click **Method/Component Trace**. The Method/Component Trace page opens.
5. Click **Export to File**.
6. Click either **Open** to view the file immediately or click **Save** to download the file.

Related topics

E-mailing a PDF file - SAD

Viewing a PDF file - SAD

Web Session Browser

The Web Session Browser page provides information about open HTTP sessions. You can search a server, a group, or all servers and groups for all sessions or sessions with a specific user name.

Visualization Engine displays only sessions on servers that have the monitoring level set to L2 or L3.

To search for web sessions on all servers, select **All Groups** in the **Group** control and **All Servers** in the **Server** control.

To search for web sessions on all servers in a group, select the group name in the **Group** control and **All Servers** in the **Server** control.

To search for web sessions on specific servers, select the group name in the **Group** control and the server name in the **Server** control.

If you want to search for sessions with a specific user name, enter it in the **Username** control.

To launch the search, click the **OK** button to view the search results. See "Viewing the Web Session Browser."

Viewing the Web Session Browser

About this task

Use the Web Session Browser to find information about HTTP sessions. Search a server, a group, or all servers and groups for a specific session. After activating the search, the system takes a snapshot of the servers and returns a list of sessions. Using a wildcard (*) indicates that all data is searched. Blank fields are disallowed in the search. The data collector must be at an L2 or higher monitoring level to provide the data for this display page.

To view the Web Session Browser:

Procedure

1. From the top navigation, click **Problem Determination > Web Session Browser**. The Web Session Browser page opens.
2. Select a group and server and enter the user name that you want to search for. The Web Session Browser page opens showing the sessions that match your search criteria. After the system returns your search results, you can review the attributes of a session by clicking the **View** link in the Attributes field.

   Many different kinds of Java objects, including customized objects, can be in a session; sometimes in the form of binary data. ITCAM cannot display this data on the front end in a useful way. In order to display the data correctly, define the toString() method. ITCAM calls this method and displays the data returned in the Web Session Browser. If the toString() method is not defined, ITCAM publishes the content of the Java objects as-is, which might not contain useful data.

   Related topics

   Memory Diagnosis
   View heap and memory information.

Server activity
The Server Activity Display page shows a table of request activity on an application server. This page can help you troubleshoot and fix hanging requests and evaluate the current performance of your applications.

Memory diagnosis

View heap and memory information.

Memory Diagnosis includes the following features: Memory Analysis, Heap Analysis, Memory Leak, and Heap Dump Management. Gain insight into the JVM heap and memory information through memory diagnosis. Use this information to tune the JVM parameters, assess your resources, and find evidence of memory leaks.

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. Heap Dump files can be viewed using Memory Dump Diagnostic for Java (MDD4J).

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 must 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.

Heap Dump Management

Use the Heap Dump Management pages to view and schedule heap dumps for the monitored servers.

To access Heap Dump Management, from the top navigation click **Problem Determination > Memory Diagnosis > Heap Dump Management**.

To create heap dumps, you need to install IBM Support Assistant and Memory Dump Diagnostic for Java on the monitored servers. See “Downloading Memory Dump Diagnostic for Java from IBM Support Assistant” on page 131.

To view information about existing heap dumps and delete the heap dumps you no longer want to store, select **Heap Dumps** in the left navigation pane. See “Heap Dumps” on page 132.

To view all scheduled heap dumps, modify and delete the schedules, select **Heap Dump Schedule** in the left navigation pane. See "Heap Dump Schedule" on page 133.

To schedule a new heap dump, select **Schedule a Heap Dump** in the left navigation pane. See "Scheduling a heap dump" on page 133.

Downloading Memory Dump Diagnostic for Java from IBM Support Assistant

About this task

In order to download Memory Dump Diagnostic for Java (MDD for Java), you must first download IBM Support Assistant (ISA). ISA provides extra help with diagnosing problems and provides extra tools and components for troubleshooting as well as providing a place to write problems (PMR). MDD for Java analyzes either a single heap dump or analyzes and compares two heap dumps and searches for evidence of a memory leak. You can either manually take a heap dump or schedule a heap dump using the Heap Dump Management tool and then download the heap dump to your PC and analyze it using MDD for Java.

MDD for Java analyzes only heap dumps from IBM JDKs. For non-IBM JDKs use ITCAM Heap Analysis features.

Searching capabilities are not supported for ITCAM for WebSphere in ISA.

To download ISA:

Procedure

1. Go to the URL https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=isa
2. If you do not have a universal IBM user ID, you must click **register now** and enter the required information. Upon completion, you can return to this page and sign in to download ISA.
3. After signing in, select the radio button to download IBM Support Assistant Version 3.0.0.1.
4. Click **Continue**.
5. Select **View license**. After reading the license, click the **I agree** check box and then click **I confirm** to continue with the download.
6. Click **Download now** next to the correct platform. We suggest you download to the server on which the data collector is installed. Do not download to the server that has the managing server.
7. Click **Save** to download ISA to your hard disk.

Results

To install ISA:

1. Go to the directory on your hard disk where you saved the ISA compressed file.
2. Extract the files.
3. Double-click the setupwin32.exe file.
4. Follow the installation instructions to install ISA.
5. Open the ISA program.

To install MDD for Java:

1. Double-click Updater to open.
2. Select the **New Products and Tools** tab.
3. Open the WebSphere directory by clicking the + sign.
4. Click the check box to select WebSphere Application Server. (The version you select does not matter.)
5. Click **Install**.
6. Select the feature under Features to Install and review the license.
7. To accept the license, click **Yes**.
8. Click **OK** to accept the message that displays, "New product plug-ins or tool plug-ins were installed successfully. Make sure to restart IBM Support Assistant for these changes to take effect."
9. Click the + sign next to the Common Component Tools directory.
10. Click the check box to select Memory Dump Diagnostic for Java (MDD4J) version 3 or above.
11. Click **Install**.
12. Click **OK** to accept the message that displays, "Reminder–You are installing a common component tool. After installation, you might not see the common component tool in the Tools component. Common component tools display only in the Tools component if a product is added that uses them." This is why you must install a version of WebSphere.
13. Select Memory Dump Diagnostic for Java under the Features to Install and review the license.
14. Click **Yes** to accept the license.
15. Click **OK** at the restart IBM Support Assistant message.
16. Close ISA and restart.

Related topics

Scheduling a heap dump

"Heap Dump Schedule" on page 133
The Heap Dump Schedule page shows a table of scheduled heap dumps on the monitored servers. From this page, you can modify and delete the schedules.

Heap Dumps

The Heap Dumps page shows a table of existing heap dumps on the monitored servers. From this page, you can delete the heap dumps you no longer want to store.

To view the Heap Dumps page, from the top navigation click **Problem Determination > Memory Diagnosis > Heap Dump Management**.

In the table, heap dumps are grouped by server. You can hide the heap dumps for a server by clicking **-** next to the server name, and show them again by clicking **+**.

To view heap dumps for servers in a server group, select the group name in the **Group** box. To view heap dumps for a single server, select the server name in the **Server** box.

For every heap dump, the table shows:

- The fully qualified file name. (The file is located on the monitored server).
- The date and time when the heap dump was created.
- The origin of its creation ("Schedule" for normal scheduled heap dumps).

- Whether garbage collection was forced before creation of the heap dump. This option can be set in the heap dump schedule.
- A **Delete** button.

To delete a heap dump, click the **Delete** button.

You can use the left navigation pane to view scheduled heap dumps and schedule a new heap dump. See "Heap Dump Management" on page 130.

Heap Dump Schedule

The Heap Dump Schedule page shows a table of scheduled heap dumps on the monitored servers. From this page, you can modify and delete the schedules.

To view the Heap Dump Schedule page, from the top navigation click **Problem Determination > Memory Diagnosis > Heap Dump Management**. Then from the left navigation pane, Select **Heap Dump Schedule**

In the table, scheduled heap dumps are grouped by server. You can hide the heap dumps for a server by clicking **-** next to the server name, and show them again by clicking **+**.

To view heap dump schedules for servers in a server group, select the group name in the **Group** box. To view heap dumps for a single server, select the server name in the **Server** box.

For every heap dump schedule, the table shows:

- The date and time when the heap dump is to be created.
- Whether garbage collection is to be forced before creation of the heap dump.
- A **Modify** button.
- A **Delete** button.

To modify a heap dump schedule, click the **Modify** button. This opens the Schedule a Heap Dump page (see "Scheduling a heap dump"). You can modify the settings. To save the changes, click **OK**

To delete a heap dump schedule, click the **Delete** button.

You can use the left navigation pane to view existing heap dumps and schedule a new heap dump. See "Heap Dump Management" on page 130.

Scheduling a heap dump

About this task

You have the flexibility to schedule a heap dump to take place now or in the future.

To schedule a heap dump:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Heap Dump Management**. The Heap Dump Management page opens.
2. Click the **Schedule a Heap Dump** link.
3. Select a group and a server from the lists.

4. Under Heap Dump Schedule, select Now to schedule a heap dump to take place immediately, or select your own date and time in the future.
5. Under Garbage Collector Before Heap Dump, select yes or no based on whether you want a garbage collection performed before the heap dump.
6. Click **OK** to save the schedule for the heap dump.

Related topics

Downloading Memory Dump Diagnostic for Java from IBM Support Assistant

"Heap Dump Schedule" on page 133
The Heap Dump Schedule page shows a table of scheduled heap dumps on the monitored servers. From this page, you can modify and delete the schedules.

Memory Analysis

Use the Memory Analysis page to create a memory analysis report, to help investigate potential memory problems related to garbage collection and the JVM heap size.

To access the Memory Analysis page, from the top navigation click **Problem Determination > Memory Diagnosis > Memory Analysis**.

To create a Memory Analysis report, select the server group name in the **Group** box, select the server name in the **Server** box, select the analysis type (Garbage Collection or Java Heap Size), and click **Next**. In the Metric Selection, select the option that contains the two metrics that you want to compare, and click the **View Results** button. See "Creating a Memory Analysis report."

Creating a Memory Analysis report

About this task

Investigate potential memory problems related to garbage collection and the JVM heap size by using the Memory Analysis. When there is over 24 hours of data, your reports show the last 48 hours; in all other cases, the last 60 minutes of data displays.

To create a Memory Analysis report:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Analysis**. The Memory Analysis page opens.
2. Select a group and a server or all servers from the lists.
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 that you want to compare.
6. Click **View Results**. The Memory Analysis report displays.
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. If there is over 24 hours of data available, your report shows the last 48 hours. Otherwise your report displays the last 60 minutes.

Related topics

Creating a Memory Leak Candidate Finder report

Creating a Memory Leak Confirmation report

Heap Analysis

Use the Heap Analysis page to set up a heap analysis, which captures the runtime heap of an application server, breaks it down by class names of objects in the heap, and provides the number of instances and the size they occupy.

To access the Heap Analysis page, from the top navigation click **Problem Determination > Memory Diagnosis > Heap Analysis**.

To set up a heap analysis, select the server group name in the **Group** box, select the server name in the **Server** box, select whether a Garbage Collection must be forced before taking the heap snapshot, and click **OK**. See "Setting up a Heap Analysis."

Setting up a Heap Analysis

About this task

The 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 they occupy.

To set up a Heap Analysis:

Procedure

1. From the top navigation, 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 before the Heap Analysis snapshot.
4. Click **OK**. The Heap Analysis results display in the same window.
5. To narrow the results, enter the names of the classes that you want to ignore into the Exclude (Class name) list. If you specify regular expressions in the Exclude list, but want to monitor a subset of these, enter the names of classes that you want to monitor into the Exclude Override (Class name) list.
6. Click **Apply**. The new Heap Analysis displays.
7. Click **Reset** to return the class name filters to their original settings.

Related topics

Creating a Memory Analysis report

Creating a Memory Leak Candidate Finder report

Creating a Memory Leak Confirmation report

Viewing a Memory Leak Candidate Finder report

Memory Leak

Use the Memory Leak page to create a Memory Leak Confirmation report, which might help determine whether a memory leak is occurring in one of the applications on a server. You can also use this report to create Memory Leak Diagnosis and Memory Leak Candidate Finder report, which might help diagnose the source of a memory leak.

To access the Memory Leak page, from the top navigation click **Problem Determination > Memory Diagnosis > Memory Leak**.

To create a Memory Leak Confirmation report, select the server group name in the **Group** box, select the server name in the **Server** box, and click **OK**. See "Creating a Memory Leak Confirmation report."

Creating a Memory Leak Confirmation report

About this task

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:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**. The Select a server page opens.
2. Select a group and a server from the lists and click **OK**. The Memory Leak Confirmation report displays.
3. Use the lists in the Report Control box to select a new group, server, date range, metric 1, 2, or 3 and click **OK** to rerun the report. If there is over 24 hours of data available, your report shows the last 48 hours. Otherwise your report displays the last 60 minutes.

Results

To get more information, you can view Memory Leak Diagnosis and Memory Leak Candidate Finder reports.

A Memory Leak Diagnosis report provides application diagnostic details. To create and view a Memory Leak Diagnosis report, click **Memory Leak Diagnosis** in the left navigation pane. See "Creating a Memory Leak Diagnosis report."

A Memory Leak Candidate Finder report creates a heap snapshot, waits a specified time, and creates another snapshot. Comparing these snapshots can help reveal memory leaks in applications. To start the creation of a Memory Leak Candidate Finder report, click **Create New Candidate** in the left navigation pane, see "Creating a Memory Leak Candidate Finder report" on page 137. After the specified time has elapsed, view the report by clicking **View Existing Candidates** in the left navigation pane, see "Viewing a Memory Leak Candidate Finder report" on page 138.

Related topics

Creating a Memory Analysis report

Creating a Memory Leak Candidate Finder report

Setting up a Heap Analysis

Creating a Memory Leak Diagnosis report

About this task

In the Memory Leak Diagnosis report each row in the Suspected Memory Leaks table represents an allocation pattern for which memory allocation data has been collected. When a memory leak occurs objects accumulate on the heap and increase over time. The growth data informs you as to what is growing on the heap.

To create a Memory Leak Diagnosis report:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**. The Select a server page opens.
2. Select a group and a server from the lists and click **OK**. The Memory Leak Confirmation report displays.
3. From the Additional Tools box, click the **Memory Leak Diagnosis** link. The Memory Leak Diagnosis report page now displays. It shows that the server and group you selected, as well as essential server information and the date and time that this report was generated. The list of suspected memory leaks follows.

   If you receive the error CYNVE0746E: *This server not instrumented for detail heap data collection.* This might indicate that the application server has not received any requests yet. Run Memory Leak Diagnosis again when you know that requests are in the system. Check that the Level 3 monitoring mode and the heap analysis flag on the data collector have been enabled. If you still receive this error, contact the system administrator.
4. Click the **Class Name** link to view references to live objects on the heap. See "Viewing References to Live Objects on the Heap" on page 138.

   Related topics

   Creating a Memory Analysis report

   Creating a Memory Leak Confirmation report

   Viewing a Memory Leak Candidate Finder report

   Viewing References to Live Objects on the Heap

Creating a Memory Leak Candidate Finder report

About this task

The Memory Leak Candidate Finder report lets you create a comparison report of two heap snapshots. Taking two heap snapshots will show if, over time, the number of instances of a specific class is increasing. In cases, where the number of instances of a class continues to rise over a time, the report demonstrates this leak candidate.

To create a Memory Leak Candidate Finder Report:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**. The Select a server page opens.
2. Select a group and a server from the lists and click **OK**. The Memory Leak Confirmation report displays.
3. From the Additional Tools box, click **Create New Candidate** link. The Create New Candidate page opens.
4. Select a group and a server.
5. Enter the Wait Time and select hours or minutes from the box. (There is a 48 hour maximum.) The Wait Time is the amount of time the system must wait before taking the second heap snapshot.
6. 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. 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.

Related topics

Creating a Memory Analysis report

Creating a Memory Leak Confirmation report

Viewing a Memory Leak Candidate Finder report

Viewing a Memory Leak Candidate Finder report

About this task

The Memory Leak Candidate Finder report displays the data comparison between two heap snapshots. The heap results display unfiltered. You can filter the heap results by class name using the Exclude (Class name) and Exclude Override (Class name) lists. Use this data to evaluate whether a memory leak is in progress on your system.

To view a Memory Leak Candidate Finder report:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**. The Select a server page opens.
2. Select a group and a server from the lists and click **OK**. The Memory Leak Confirmation report displays.
3. From the Additional Tools box, click **View Existing Candidates** link. The Memory Leak Candidate Finder Management page opens.
4. The status for your report displays. Click the **Server Names** link to open your previously created report. The Memory Leak Candidate Finder report opens.
5. Click **Comparison Data** on the left navigation pane. The comparison data displays with the data for each heap snapshot.
6. To view each heap individually, click either Heap 1 or Heap 2 on the left navigation pane. Doing this gives you another view of the heap analysis for your current data.
7. To filter your data more precisely, enter the classes you do not want to analyze into the Exclude (Class name) list. If you specify regular expressions in the Exclude list, but want to monitor a subset of these classes, enter the classes that you want to monitor into the Exclude Override (Class name) list. The report refreshes and displays with the current data. When the comparison for the Memory Leak Candidate Finder report displays the heap snapshot data, the data includes the class name, 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.

   Related topics

   Creating a Memory Leak Candidate Finder report

   Creating a Memory Analysis report

Viewing References to Live Objects on the Heap

About this task

Occasionally objects on the heap do not get garbage collected because another object has a reference to the original object. After clicking the Class Name link, the data displays which object is linked to the object that is growing.

To view the References to Live Objects on the Heap:

Procedure

1. From the top navigation, click **Problem Determination > Memory Diagnosis > Memory Leak**. The Memory Leak Overview page opens.
2. From step 3, select a group and a server and click **View Diagnosis**. The Memory Leak Diagnosis page opens.
3. Select the Class Name link. The References to Live Objects on the Heap page displays.

Related topics

Creating a Memory Analysis report

Creating a Memory Leak Confirmation report

Creating a Memory Leak Diagnosis report

Viewing a Memory Leak Candidate Finder report

JVM thread display

Use the JVM thread display page to view all threads running within the JVM of an application server.

To view the JVM thread display page, complete the following steps:

1. From the top navigation click **Problem Determination > JVM Thread Display**.

   You can also access this page from the toolbox icon  on the Server Activity Display page.
2. Select the server group name in the **Group** box, and the server name in the **Server** box. If you access this page from the Tivoli Enterprise Portal and from the Server Activity Display page, the server group name and server name is displayed automatically.
3. At the top of the page, the snapshot date and time and the server name and IP address are displayed. The timestamp for Snapshot Date and Snapshot Time is collected from the data collector and converted to the time zone of the managing server.
4. The page initially displays a list of top-level thread groups and active threads.
5. To view thread groups and active threads within a thread group, click the name of the thread group. A **Thread Group Properties** pane is displayed on the right of the page, showing information for the current thread group. To return to an upper level thread group, click the button that has two dots (**..**).
6. To view information for a thread, click its name. A **Thread properties** pane is displayed on the right of the page, showing information for the thread.
7. To change priority for the thread, in the **Thread properties** pane select the new priority in the **Priority** box and click **Change Priority**. See "Changing the priority of a JVM thread" on page 140.

   Important: The new priority remains for the life of the thread. As a result, any requests assigned to that thread after the change hold that priority.

To view a stack trace for the thread, in the **Thread properties** pane click **View Stack Trace**. See "Viewing a stack trace" on page 140.

Attention: Canceling a thread might cause JVM and application server instability. Use the cancel thread function only when necessary, and with careful consideration of the consequences.

To cancel the thread, in the **Thread properties** pane click the **Cancel Thread** button. See “Canceling a thread” on page 141.

To troubleshoot a multithreaded application, you might need to view the thread dump, which includes detailed information about memory allocation of all threads in the JVM. To view the thread dump, click the **Thread Dump** button, located in Server Properties at the top of the page. See “Viewing a thread dump” on page 142.

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 do not see any active requests in the In-flight Request Search, so you suspect there might be threads running outside the application server. You access the JVM Thread Display and notice a couple of suspect threads. After a thread dump for the JVM, find the details of the current thread that is malfunctioning and either reprioritize or cancel the thread.

Changing the priority of a JVM thread

About this task

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

To change the priority of a thread:

Procedure

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 group running in the selected server.
3. Select and click to the right of the thread for detailed information.
4. From the priority box 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 box displays the priority you selected for the thread to execute request. When changing the priority of a thread, note that the new priority remains for the life of the thread. As a result, any requests assigned to that thread after the change hold that priority during the lifetime of that request.

   Related topics

   Canceling a thread

   Viewing a stack trace

Viewing a stack trace

About this task

The Stack Trace page displays the sequence of method execution and in last-in first-out order. The last executed method displays first in the stack trace.

To view a stack trace:

Procedure

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 all the top-level thread groups that are 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 Stack Trace** The Stack Trace page opens.

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

   Related topics

   Changing a JVM thread's priority

   Canceling a thread

Canceling a thread

About this task

If a thread is malfunctioning, for example looping, sleeping, or abusing resources, it might be necessary to cancel the thread and stop the Java thread that is running to let other threads proceed. By default, only the Administrator role has access to the canceling a thread function.

To cancel a thread:

Procedure

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, displaying all the top-level thread group that running in the selected server.
3. Select and click a thread that you want to view its 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 refreshed displaying without the canceled thread.

   Canceling a thread might cause JVM and application server instability. Use the cancel thread function only when necessary, and with careful consideration of the consequences.

   Related topics

   Changing a JVM thread priority

   Viewing a stack trace

Viewing a thread dump

About this task

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

When a user clicks thread dump on the JVM Thread Display page, a snapshot is taken showing the data about all threads.You can view the Thread Dump page for detailed information about memory allocation of threads in a JVM.

To view the Thread Dump page:

Procedure

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 all of the top-level thread groups that are running in the selected server.
3. Click **Thread Dump**. The Thread Dump page opens.

Related topics

Changing a JVM thread's priority

Canceling a thread

Trap and alert management

Use the Trap and alert management page to define and modify traps and alerts that monitor server health and determine problems with applications, and to view the history of triggered traps and alerts.

To access trap and alert management, from the top navigation, click **Problem Determination > Trap & Alert Management**.

You can also access this page from the toolbox icon  on the Server Activity Display page.

Use trap and 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.

The page shows a table of active traps and another table of defined trap profiles. (You can activate a trap based on a trap profile).

For each active trap the table shows:

- The name of the trap.
- The name of the server that the trap applies to. If the trap applies to multiple servers, you can click + to view the list.
- The number of minutes for which the trap is suppressed before triggering, if applicable. This avoids triggering a trap on a short load spike.
- The duration of the trap; the trap is deactivated when this duration expires. Traps can also have an infinite duration.

- Time left until the duration expires (N/A if the duration is infinite).
- The time when this trap was activated.
- The user who has activated the trap.
- A **Deactivate** button.

To deactivate a trap, click the **Deactivate** button. See “Deactivating a trap” on page 151.

For each trap profile the table shows:

- The name of the trap profile.
- The description.
- The user who has created the profile.
- **Activate**,**ModifyModify** and **Delete** buttons.

To activate a trap based on the profile, click the **Activate** button. See “Activating a trap” on page 150.

To modify a trap profile, click the **Modify** button. See “Modifying a trap” on page 152.

To create a trap profile as a copy of an existing profile, click the **Duplicate** button. See “Duplicating a trap” on page 153.

To delete a trap profile, click the **Delete** button. See “Deleting a trap” on page 153.

To create a trap profile, click **Create trap** in the left navigation pane, and select the trap type and target type. See:

- “Setting an Application trap” on page 144
- “Setting an Application trap using the Resident Time - Misbehaving Transaction target type” on page 146
- “Setting a Server Resource trap” on page 148

To view the history of triggered traps, click **Trap Action History** in the left navigation pane. See “Viewing the trap action history” on page 154.

User Scenario

Scenario 1: Debugging complex applications

You are monitoring application A, which has a J2EE component on server S and a preexisting 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 want 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 2 seconds, the system takes a stack trace of server S. Look in the Trap Action History to obtain the stack trace, to determine where application A is waiting.

Setting an Application trap

About this task

An Application trap detects metrics in a request, method, or 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, any alerts you have activated (whose conditions have been met) are sent, and any actions you have specified (whose conditions have been met) are performed.

To set an Application trap:

Procedure

1. From the top navigation, click **Problem Determination > Trap & Alert Management**. The Trap and Alert Management page opens.
2. On the left navigation pane, click **Create Trap**. The Trap Type selection page opens.
3. Select **Application Trap** as the trap type.
4. Select one of the Target Types from the box. Based on the target type you select, the system dynamically generates the trap definition options in the next step. The following list shows the Target Types:
   - **Occurrence** - The number of times the specified unit occurred. The Occurrence trap has three available filters: By Request, By Method, and By SQL.
   - **CPU Time** - The amount of time the CPU is executing instructions. The CPU Time trap has two available filters: By Request and By Method.
   - **Wait Time** - The amount of time the CPU is idle. The Wait Time trap has two available filters: By Request and By Method.
   - **Resident Time - In-flight** - Based on the resident in-flight time of a transaction, if the resident time of the request exceeds the time configured in the trap condition, the Publish Server tracks all the active (in-flight) requests and their resident times and triggers the trap. The Resident Time - In-flight trap has one available filter, the By Request filter.
   - **Resident Time - Completed** - The wall clock time for when the unit of a transaction, method, and so on ends, minus the wall clock time when it started. The Resident Time - Completed trap has three available filters; By Request; By Method, and By SQL.
   - **Resident Time - Misbehaving Transaction** - This trap has one available filter, the By Request filter. With this target type, when the complete response request time violates the threshold in the trap definition, the monitoring level for the request switches from L1/L2 to L3 and component/method trace detail is captured. Because switching from L1/L2 to L3 has a performance impact on the data collector, you can use two fields to deactivate the trap and return to the original L1/L2 monitoring level once the required detail has been captured:
     – **Number of occurrences of every request after which the trap will be deactivated** - This field prevents the data collector from running at L3 indefinitely. The value in this field determines the number of times you want every request to reach the threshold before the trap is deactivated. Using this field enables you to capture component/method trace detail at L3 when the threshold is exceeded, and to then automatically revert to the original monitoring level, reducing the performance cost to the server.

– **Number of occurrences of every request that doesn't violate this trap after which mod level is reverted back and trap is deactivated** - Once L3 is enabled, after the trap condition is violated the first time, it remains at L3 until the request violates for the predetermined number of times as set in the **Number of occurrences of every request after which the trap will be deactivated** field. As a result, the request in the data collector remains at L3 if the request does not violate for the predetermined number of times, resulting in performance cost to the data collector. To prevent this, use this field: **Number of occurrences of every request that doesn't violate this trap after which mod level is reverted back and trap is deactivated**.

  Once the trap triggers and the monitoring level switches to L3, if the number of requests that does not reach the threshold is equal to the value in this field, the trap is deactivated. For more details about this target type, see "Setting an Application trap using the Resident Time - Misbehaving Transaction target type" on page 146

- **Uncaught Exception** - Capture exceptions that occur in applications and data about the failure. The Uncaught Exception trap has three available filters; By Request, Exception or Error Class Name.
- **Lock Acquisition Time - In-flight** - The in-flight transaction has not completed and might be hanging, using this type provides data on how to fix the problem. What is being measured is the amount of time taken to obtain an acquisition. The acquisition clock starts when the class/method begins trying to acquire the lock, and ends when the lock is acquired. The Lock Acquisition Time - In-flight trap has two available filters, By Request and By Method.
- **Lock Acquisition Time - Completed** - The amount of time taken to obtain an acquisition. The acquisition clock starts when the class/method begins trying to acquire the lock, and ends when the lock is acquired. The Lock Acquisition Time - Complete trap has two available filters, By Request and By Method.

Traps are now supported in CICS. CICS has the following trap types available: Occurrence, CPU time, Resident Time - Completed, Resident Time - In-flight, and Wait Time.

Important: By default, the Publish Server does not process requests which run longer than 5 minutes. Therefore, if an application trap threshold is greater than or equal to 300 seconds, the trap is not triggered. To change this default setting in the Publish Server, in the `psX.properties` file, change the TIMEOUT_LIMIT property to greater than 5 minutes as required. The properties files for the Publish Servers are located here: `MS_HOME/etc/`. They are named with the convention `psX.properties`, where X is an integer. By default, there are two files: `ps1.properties` and `ps2.properties`. If you add another Publish Server instance, the properties file is called `ps3.properties` and for all additional instances of the Publish Server, the integer value in the properties file name increment by 1.

5. Click **Next**. The Step 2 - Define Trap page opens.
6. Complete the rest of the fields in the Trap Definition section, which restrict which events trigger the trap to fire.
7. Click **Next**. The Set Trap Alerts page opens.
8. For the Trap Alert settings, under Condition enter the number of times the trap occurs before the system takes an action. Specify the amount of time under Time Interval to monitor how many times the trap met its conditions.

9. Click to select the severity level from the box. The application monitor has three severity levels. Because the application monitor provides SNMP integration with Tivoli, map the three severity levels of the application monitor to the warning levels of Tivoli listed in the following table:

ITCAM severity level	Tivoli warning level
Low	Harmless
Medium	Minor
High	Critical

10. Select an action or multiple actions, such as sending an email or SNMP message, for the system to take when the condition is met.
11. Select one or all Data Actions, such as Component/Method Trace, Stack Trace, or Thread Dump (not applicable to the Windows platform), to get detailed information. We recommend that you select Component/Method Trace as the data action, since a request executes quickly and it is difficult to catch before completion. Make sure that you have selected L3 monitoring level if you choose to collect Component/Method Trace as the Data Action. When setting a trap, you can select multiple trigger conditions and alerts for each action set. Each trap is required to have at least one action but might have multiple actions set. Thread Dump is not available for CICS.
12. Click **Add** to add the alert to your trap. If you select Component/Method Trace as the Data Action for an In-flight-based trap, the method trace might contain a -1 for Depth on some events in the method trace. By definition, In-flight transactions, are incomplete transactions, so the request stacks of those transactions are incomplete.
13. Set the Default Suppression settings by entering the amount of time that you want to delay alerts after the first alert is sent.
14. Click **Next** to proceed. The Name Trap page opens.
15. Enter a name and descriptive text for your trap.
16. Click either **Save** or **Save & Activate**.
17. If you click **Save**, the Trap and Alert Management page opens displaying your new trap.
18. If you click **Save & Activate**, the Activate page opens. To activate a trap, see **Activating a Trap**.

Related topics

Activating a trap

Setting a Server Resource trap

Setting an Application trap using the Resident Time - Misbehaving Transaction target type

About this task

A Resident Time - Misbehaving Transaction is a target type for an Application Trap. With this target type, when the resident time of a request violates the specified level in the trap definition, the monitoring level for that specified request switches from L1/L2 to L3. For all the subsequent resident time violations for that request, method trace detail is captured.

This target type provides an efficient means of collecting method trace detail at L3 because you can configure the data collector to return to L1/L2 after the threshold

is reached a certain number of times within a given time period, reducing the performance cost to the data collector. This target type also provides a dynamic means of collecting method trace detail as detail is collected at the time the problem is occurring.

To set an Application trap with a Resident Time - Misbehaving Transaction target type:

Procedure

1. From the top navigation, click **Problem Determination > Trap & Alert Management**. The Trap and Alert Management page opens.
2. On the left navigation pane, click **Create Trap**. The Trap Type selection page opens.
3. Select **Application Trap** as the trap type.
4. Select **Resident Time - Misbehaving Transaction** as the target type.
5. Click **Next**. The Step 2 - Define Trap page is displayed. In the **Threshold** field, enter a value. This value is the complete request response time expressed in milliseconds. When a transaction reaches this value, the trap is triggered.
6. In the **By Request** field, choose **Request Contains** and then enter the value *. When the **By Request** field contains the value *, the Resident Time - Misbehaving Transaction trap will deactivate after every request has reached the specified number of occurrences as specified in the **Number of occurrences of every request after which the trap will be deactivated** field.
7. Click **Next**. The Step 3 - Set Trap Alerts page is displayed. In the **Severity** field, select a severity level. The application monitor has three severity levels. Because the application monitor provides SNMP integration with Tivoli, map the three severity levels of the application monitor to the warning levels of Tivoli listed in the following table:

ITCAM severity level	Tivoli warning level
Low	Harmless
Medium	Minor
High	Critical

8. In the **Alert Action(s)** section, choose how and with whom you want to communicate details of the trap threshold being reached. You can choose email, SNMP or both.
9. Select **Collect Component/Method Trace**.
10. Click **Add** to add the alert to your trap. In the **Name** section, enter a name and a description for the trap. Click **Save & Activate**. The Activate page is displayed
11. In the **Server Selection** section, select the servers that you want to monitor for this trap.
12. In the **Alert Suppression Settings** section, enter the amount of time that you want to delay 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.
13. In the **Deactivation Settings** section, there are two fields. Here is a description of each field:

- **Number of occurrences of every request after which the trap will be deactivated**:This field prevents the data collector from running at L3 indefinitely. The value in this field determines the number of times you want every request to reach the threshold before the trap is deactivated. Using this field enables you to capture component/method trace detail at L3 when the threshold is exceeded, and to then automatically revert to the original monitoring level, reducing the performance cost to the server.

  A problem can occur if this value is not reached. If the **Number of occurrences of every request after which the trap will be deactivated** value is not reached, the server will remain at L3 resulting in a continuous performance cost. To prevent this from happening, use the field - **Number of consecutive non-violating requests after which mod level is reverted back and trap deactivated**.
- **Number of consecutive non-violating requests after which mod level is reverted back and trap deactivated**: Once L3 is enabled, after the trap condition is violated the first time, it remains at L3 until the request violates for the predetermined number of times as set in the **Number of occurrences of every request after which the trap will be deactivated** field. As a result, the request in the data collector remains at L3 if the request does not violate for the predetermined number of times, resulting in performance cost to the data collector. To prevent this, use this field - **Number of consecutive non-violating requests after which mod level is reverted back and trap deactivated**. If the number of requests that do not reach the threshold is equal to the value in this field when the trap triggers and the monitoring level switches to L3, the trap is deactivated. A field for this value is also displayed in the Active Traps table in the Trap and Alert Management page. The field is **Non Violating Requests Left**, it indicates the number of occurrences of non-violating requests before the trap is deactivated.

14. Click **Activate** to activate the trap.

Results

Related topics

Activating a trap

Setting a Server Resource trap

Setting an Application Trap

Setting a Server Resource trap

About this task

A Server Resource trap measures various target types. The system will trigger a trap after exceeding the threshold for the metric you set. When the system meets the definition of the trap an alert occurs. For example, set a trap to alert you when a server is unavailable two times, and after a server is unavailable you can select to receive an email.

To set a Server Resource trap:

Procedure

1. From the top navigation, click **Problem Determination > Trap & Alert Management**. The Trap and Alert Management page opens.
2. On the left navigation pane, click **Create Trap**. The Trap Type selection page opens.
3. Select Server Resource trap as the Trap Type.

4. Select one of the following target types from the box.
   - ***CPU***: **Average Platform CPU % Usage** - Based on the average platform CPU usage over 5 minutes, the Publish Server retrieves CPU usage at regular intervals (60 seconds by default) and calculates the average platform CPU over 5 minutes.
   - **Average JVM CPU % Usage** - Based on the average JVM CPU usage over 5 minutes, the Publish Server retrieves CPU usage at regular intervals (60 seconds by default) and calculates the average JVM CPU over 5 minutes.
   - *Memory*:
   - **JVM Heap Size** - Based on the JVM Heap Size of the data collector, the Publish Server retrieves JVM Heap Size from the data collector at regular intervals (60 seconds by default) and checks the heap size from that measure.
   - **Garbage Collection Frequency** - Garbage Collection is calculated over 1 minute (supported in ITCAM J2EE WebLogic).
   - **Average JVM Heap Size after Garbage Collection** - The trap triggers when the average JVM Heap size exceeds the size configured in the trap (supported in ITCAM J2EE WebLogic)
   - *Application Capacity*:
   - **Number of Sessions** - Based on the number of user sessions that are currently in use by the application server.
   - **Average Response Time** - If the average response time exceeds the time configured in the trap condition (supported in ITCAM J2EE WebLogic), the Publish Server triggers the trap .
   - **Server available** - Publish server triggers the trap when the server (data collector) becomes available (supported in ITCAM J2EE WebLogic).
   - **Server unavailable** - If the server (data collector) goes down or becomes unavailable (supported in ITCAM J2EE WebLogic), the Publish server triggers the trap .
   - **Uncaught Java Exceptions** - Based on the rate of the Java exceptions that occur in applications and includes data about the failure. It is calculated over 60 seconds. Publish server triggers the trap if the Servlet error rates exceed the number configured in the trap condition.
   - **Request Frequency** - Number of requests per minute.
   - *Resource Pool* :
   - **Thread Pool % Usage** - Publish Server triggers the trap if the Thread Pool % Usage of a particular server exceeds the threshold that is specified in the trap condition (supported in ITCAM J2EE WebLogic).
   - **JCA Pool % Usage** - Publish Server triggers the trap if the JCA Pool % Usage of a particular server exceeds the threshold that is specified in the trap condition (supported in ITCAM J2EE WebLogic).
   - **JDBC Pool % Usage** - Publish Server triggers the trap if the JDBC Pool % Usage of a particular server exceeds the threshold that is specified in the trap condition (supported in ITCAM J2EE WebLogic).
5. Click **Next**. The Define Trap page opens.
6. Enter a threshold that will send out an alert when it triggers the trap after meeting the condition.
7. Click **Next**. The Set Trap Alerts page opens.
8. For the Trap Alert settings, under Condition enter the number of times the trap occurs before the system takes an action. Specify the amount of time under Time Interval to monitor how many times the trap met its conditions.

9. Click to select the severity level from the box.

   Note: The application monitor has three severity levels. Because the application monitor provides SNMP integration with Tivoli, map the three severity levels of the application monitor to the warning levels of Tivoli listed in the following table:

ITCAM severity level	Tivoli warning level
Low	Harmless
Medium	Minor
High	Critical

10. Select an action or multiple actions, such as sending an email or SNMP message, for the system to take when the condition is met.
11. The Data Action–Heap Dump is only available if you select JVM Heap Size or Average JVM Heap Size after Garbage Collection as your Target type.
12. Click **Add** to add the alert to your trap.
13. Set the Default Suppression settings by entering the amount of time that you want to delay alerts after the first alert is sent.
14. Click **Next** to proceed. The Name Trap page opens.
15. Enter a name and descriptive text for your trap.
16. Click either **Save** or **Save & Activate**.
17. If you click **Save**, the Trap and Alert Management page opens displaying your new trap.
18. If you click **Save & Activate**, the Activate page opens. To activate a trap, see *Activating a Trap* .

Results

If you select the <= operator while creating or modifying the Request Frequency target type, then the following sequence of events occurs:

1. If no transactions are received by the Publish Server after activating the trap, the trap action does not trigger .
2. Trap threshold checking begins after the first transaction received by the Publish Server activates the trap.

Tip: When setting a trap, you can select multiple trigger conditions and alerts for each action set. Each trap is required to have at least one action but can have multiple actions set.

Related topics

Activating a trap

Activating a trap

About this task

You can turn traps on and off by activating and deactivating them. Traps use resources on the managing server, including database storage, and generate network traffic, so make sure the thresholds you set to trigger your traps are realistic.

To activate a trap:

Procedure

1. A trap is activated when you click **Save & Activate** at the end of the trap creation process; see “Setting an Application trap” on page 144 or “Setting a Server Resource trap” on page 148 to start from the beginning of the trap creation process. To activate an existing trap, 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 that you want to activate. The Activate page opens.
3. Select a group and a server.

   Tip: If you select all servers, the trap is applied to all servers that are assigned to the group (whether currently available or unavailable) at the time the trap is activated. Any new servers that are assigned to the group do not use the trap.
4. To set the Alert Suppression settings, enter the amount of time that you want to delay alerts after the first alert is sent. Click **Trap Default** to use the default suppression for the trap. Alternatively, click **Override Default** 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 select either of the check boxes in the deactivation settings section. If you want the trap to deactivate, select one or both of the check boxes for deactivation, and enter the values for minutes or occurrences. If both deactivation settings are selected, the trap deactivates 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.

Related topics

Deactivating a trap

Duplicating a trap

Modifying a trap

Setting a Server Resource trap

Setting an Application trap

Deactivating a trap

About this task

Deactivate your traps when they are no longer required because they can add processor usage to the system. The traps in the Trap Profiles table are not active.

To deactivate a trap:

Procedure

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 that you want to deactivate.
3. Click **OK**. The trap is displayed in the Trap Profiles table as deactivated.

Results

Tip: A trap must be deactivated before modification.

Related topics

Activating a trap

Modifying a trap

Setting a Server Resource trap

Setting an Application trap

Modifying a trap

About this task

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

Remember: A trap must be deactivated prior to modification. (See "Deactivating a trap" on page 151.)

To modify a trap:

Procedure

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 **Modify** next to the trap that you want to modify. The Modify page opens.
3. To change the Trap Definition, enter a new threshold that will trigger the trap after meeting the condition.
4. To change the Trap Alert settings, under **Condition**, enter the number of times the trap occurs before the system takes any action, including sending an email or sending an SNMP message. Specify the amount of time under Time Interval to monitor how many times the trap met its conditions. In addition, set the Alert Suppression settings by entering the amount of time that you want to delay alerts after the first alert is sent.
5. Select one or all Data Actions to get detailed information (not applicable a Windows system), such as Component/Method Trace, Stack Trace, and Thread Dump. Data Action is not applicable to Server Resource trap.
6. Click **Add** to add a new alert to your trap. If you select Component/Method Trace as the Data Action for an In-flight-based trap, the method trace might contain a `-1` for Depth on some events in the method trace. In-flight transactions, by definition, are incomplete transactions, so the request stacks of those transactions are incomplete.
7. To change the name, enter a new name and descriptive text for your trap. When saved, the new name replaces the old name.
8. Click **Save**. The Trap and Alert Management page opens displaying your modified trap.

Related topics

Activating a trap

Setting a Server Resource trap

Setting an Application trap

Duplicating a trap

About this task

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

To duplicate a trap:

Procedure

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 that you want to duplicate. The Duplicate page opens.
3. Select the trap that you want to duplicate from the box.
4. Enter a name for the new trap.
5. Click **Save**. The new trap is displayed in the Trap and Alert Management page.

Related topics

Activating a trap

Deactivating a trap

Modifying a trap

Deleting a trap

About this task

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

To delete a trap:

Procedure

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

   Remember: A trap must be deactivated before deletion. (See "Deactivating a trap" on page 151)
2. In the Trap Profiles table list, click **Delete** next to the trap that you want to delete.
3. Click **OK** at the confirmation box. The Trap and Alert Management page opens displaying without the deleted trap.

Related topics

Activating a trap

Modifying a trap

Setting a Server Resource trap

Setting an Application trap

Viewing the trap action history

About this task

The Trap Action History page provides a record of traps that met the set conditions. You can 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.

To view a fired trap:

Procedure

1. From the top navigation, click **Problem Determination > Trap & Alert Management**. The Trap and Alert Management page opens.
2. On the left navigation pane, click **Trap Action History**. The Trap Action History opens displaying the information for the fired traps.
3. Click **Show Filters**. You can either filter the information by server or by server and trap name but not by the trap name only.
4. 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.

   The history of a method trace for an in-flight transaction might show `-1` for the Depth. In-flight transactions, by definition, are incomplete transactions, so the request stacks of those transactions would be incomplete which makes calculating depth for partial events (a start without an end event) impossible.

   The Trap Action History page now displays the data collector process ID information.
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.

   Related topics

   Activating a trap

   Setting alert actions and data actions

Setting alert actions and data actions

About this task

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 can configure alert actions to be suppressed, to avoid getting spammed by alerts.

To set alert actions and data actions:

Procedure

1. You can set trap alerts on the Step 3 - Set Trap Alerts page, which is part of the trap creation process. To arrive at this page, see “Setting an Application trap” on page 144 or “Setting a Server Resource trap” on page 148.

2. For the Trap Alert settings, in the condition field, enter the number of times the trap triggers before the action is taken.

   Important: This value is applied to all the trap actions defined in the next two steps. To define multiple actions, each with a different condition, repeat steps 2-4 once for each distinct condition.
3. Click to select the severity level from the box.

   Important: The application monitor has three severity levels. Because the application monitor provides SNMP integration with Tivoli, the three severity levels of the application monitor are mapped to the warning levels of Tivoli listed in the following table:

ITCAM severity level	Tivoli warning level
Low	Harmless
Medium	Minor
High	Critical

4. 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.)
5. To select an action, click its check box. For the email action, also enter the list of email addresses to which the message is sent.
6. 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. If you select Component/Method Trace as the Data Action for an In-flight-based trap, the method trace can contain a -1 for Depth on some events in the method trace. In-flight transactions, by definition, are incomplete transactions, so the request stacks of those transactions are incomplete.
7. To avoid being spammed by Alert Actions that might occur close together in time, set the Default Suppression setting
8. Click **Next** to proceed. The Name Trap page opens.
9. Enter a name and descriptive text for your trap.
10. 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".

Results

You can configure the Time Interval in the Trap Alert Settings page to milliseconds, seconds, minutes, hours, days, or weeks. The smallest available time interval unit is milliseconds. Certain trap types might not be able to use milliseconds as the time unit. Use the following table as a guideline for setting the Time Interval value with default managing server and data collector settings.

Trap Type	Target Type	Minimum Time Interval

Server Resource Trap	Average Platform CPU % Usage	5 (min)
	Average JVM CPU % Usage	5 (min)
	JVM Heap Size	1 (min)
	Average JVM Heap Size after GC	1 (min)
	Request Frequency	1 (min)
	Number of Sessions	1 (min)
Application Trap	Occurrence	1 (ms)
	CPU Time	1 (ms)
	Wait Time	1 (ms)
	Resident Time – In-Flight	1 (ms)
	Resident Time – Complete	1 (ms)
	Resident Time – Misbehaving Transaction	1 (ms)
	Uncaught Exceptions	1 (ms)
	Lock Acquisition Time – In-Flight	1 (ms)
	Lock Acquisition Time – Completed	1 (ms)

Related topics

Activating a trap

Software consistency check

Identify irregular servers.

Use the software consistency check to troubleshoot aberrant servers in an otherwise homogenous server group.

This feature is not available for CICS or IMS.

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 occasionally hang while all other servers work normally. 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.

Installed Binary Comparison

Use the Installed Binary Comparison page to compare the installed binaries on a chosen server (the Authoritative Server) with up to 10 additional servers (the Comparison Servers).

The following binaries are compared:

- Application binaries installed in WebSphere Application Server
- Asset binaries installed in WebSphere Application Server (SCA Feature Pack must be installed on the server)
- Directories and `jar` files enumerated in the WebSphere classpath (`java.class.path` environment variable).

To set up the installed binary comparison, select the authoritative servers and the comparison servers. See "Setting up an Installed Binary Comparison."

Setting up an Installed Binary Comparison

About this task

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 your servers contain the same installed binaries.

To set up an Installed Binary Comparison:

Procedure

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.
5. Click to select the File Source (EAR file or class path) and the File Types (JAR, Web, class, or image files).
6. Click **OK**. The Installed Binary Comparison results page displays the overview data first with the results of the comparison.

Results

See "Viewing the results of the Installed Binary Comparison."

Related topics

Viewing the results of the Installed Binary Comparison

Viewing the results of the Installed Binary Comparison

About this task

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.

To view the results of the Installed Binary Comparison:

Procedure

1. Navigate the results of the binary comparison by clicking the expansion icon next to the server name on the left navigation pane.
2. To view further details, click the server name and select either the Matched or Unmatched folders.

3. 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.
4. To perform an MD5 on a file, click **Perform MD5**. You can perform an MD5 only on files that are a Full Match or a File Name/Path/Size Match. The results display whether the files matched or not at the MD5 level.

Results

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, the user can opt to further perform an MD5 operation on the files. An MD5 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. A user can opt to perform an MD5 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 exists only on the Authoritative Server.
- A **Comparison Only** indicates that the file exists only on the Comparison Server.

Related topics

Setting up an Installed Binary Comparison

Installed Binary Check

The Installed Binary Check page provides a list of the installed binaries deployed to the selected server.

To access the Installed Binary Check page, from the top navigation, click **Problem Determination > Software Consistency Check > Installed Binary Check**.

Select the server group name in the **Group** box, and the server name in the **Server** box, and click the **Refresh** button to view the binary information for the server.

The page shows a table of all application binaries installed on the server; the type, fully qualified name, modification time, and size are displayed.

Information about the following binaries is displayed:

- Application binaries installed in WebSphere Application Server
- Asset binaries installed in WebSphere Application Server (SCA Feature Pack must be installed on the server)
- Directories and `jar` files enumerated in the WebSphere classpath (`java.class.path` environment variable).

In the **Name** column, click directory names (these names display **DIR** in the **Type** column) to view details on binaries within the directories.

Click on files of type JAR and EAR to view a list of their contents.

Runtime Environment Comparison

Use the Runtime Environment Comparison page to compare the runtime environments on a chosen server (the Authoritative Server) with up to 10 additional servers (the Comparison Servers).

To set up the runtime environment comparison, select the authoritative servers and the comparison servers. See "Setting up a Runtime Environment Comparison."

Setting up a Runtime Environment Comparison

About this task

Analyze the data in the Runtime Environment Comparison and find out if the runtime environments on all your clone servers are set up the same.

To set up a Runtime Environment Comparison:

Procedure

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 + the server name**.
4. Click **Next** to continue. The data is displayed on the Runtime Environment Comparison results page.
5. For specific data on the server, click any of the options in the left navigation under System Runtime Environment, Java Runtime Environment, and the App Server Runtime Environment.
6. For a complete detail report on a particular server, click the server 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.

Related topics

Viewing the results of the Installed Binary Comparison

Runtime Environment Check

The Runtime Environment Check page provides runtime environment details for the selected server, including host computer, JVM and application server information.

To access the Installed Binary Check page, from the top navigation, click **Problem Determination > Software Consistency Check > Runtime Environment Check**.

Select the server group name in the **Group** box, and the server name in the **Server** box in the left pane.

Important: If you have a WebSphere Stack Product, such as WebSphere Process Server, WebSphere Portal Server, WebSphere Enterprise Service Bus, and so on,

which runs on an IBM WebSphere Application Server, the name of the application server is displayed when you Run the Runtime Environment Check.

Important: After configuring data source with the administration web application provided by Tomcat, the MBean from MBeanServer cannot be retrieved. As the result, the JDBC Connection Pools column is always 0 on the page Runtime Environment Check. This is a Tomcat 5.0 bug (latest version Tomcat 5.0.28) and occurs only in the Tomcat 5.0 series. Tomcat 5.5 series does not have the problem.

Performance analysis and reporting

Analyze applications and servers.

Use performance analysis and reporting to analyze historical data. Performance analysis and reporting helps you understand the performance of your applications and the utilization of your servers.

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 ITCAM for Application Diagnostics and draw up a response Trend Report for Application A for the last week to verify the claims of the customer. When you are able to see that there indeed are instances of poor response time, you decompose the problematic period to see how different requests affect 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 must 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.

Defining reports

Set report requirements.

Set different requirements for generating reports to analyze the performance of application servers.

Defining a Request/Transaction Analysis report
About this task

The Request/Transaction Analysis report provides a whole picture about the behavior on the application server. After defining the request/transaction analysis, several reports become available: Trend report, Decomposition report, Request report detail, and Trace report. Each of these reports provides more specific data for understanding the application performance at every level.

To define a Request/Transaction Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports > 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 these instructions, we are selecting no. If you want further instructions on scheduling reports, see Defining a Schedule report.
3. Select the group and the server from the box.
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** - Throughput per Hour, Throughput per Second, Throughput per Minute, Response Time, and CPU Time.
   - **Request Type** - All, EJB, JSP, Servlet, SCA, CICS, Web Service, OTMA, VTAM®, BTAM, APPC, Portal, and RMI-IIOP.
   - **Request Name** - Unless you know the exact request string, always leave the field blank to return all requests or type the specific request 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.
8. Click **View Report** to view the report. 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**.
10. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a Method/Program Analysis report

About this task

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/Program report are available.

To define a Method/Program Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports > Method/Program**. The Create reports page opens.

2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting no. If you want further instructions on scheduling reports, refer to Defining a Schedule report.
3. Select the group and the server from the box.
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** - Throughput per Second, Throughput per Minute, Throughput per Hour, Response Time (ms), and CPU Time (ms).
   - **Method/Component Trace** - Unless you know the exact string, always leave the field blank to return all traces or type the specific method/program/component name.

     Important: The name of this field represents the names used in J2/WAS and z/OS. Method name is used in the J2/WAS environments, and Program and component name are used in the z/OS environments.
   - **Request Type** - ALL, EJB, JSP, Servlet, CICS, Web Service, and Portal.
   - **Request Name** - Unless you know the exact request string, always leave the field blank to return all requests or type the specific request 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.
8. Click **View Report** to view the report. 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**.
10. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a SQL Analysis report

About this task

The SQL Analysis report provides the information for the SQL call performance in the requests that have been processed by the application server. You can also view the Trend report, Decomposition report, and detailed SQL report after defining the SQL Analysis report.

To define a SQL Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports > SQL**. The Create report page opens.
2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting No. If you want further instructions on scheduling reports, refer to Defining a Schedule report.
3. Select the group and the server on which you want to report from the box.
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** - Throughput per Second, Throughput per Minute, Throughput per Hour and Response Time.
   - **SQL Call** - All, Insert, Delete, Update, Execute, Select, Lock, Unlock, Open, Close, and Fetch.
   - **Table Name** - Leave blank for all table names or type the specific table name.
   - **Request Type** - All, EJB, JSP, Servlet, CICS, Web Service, and Portal.
   - **Request Name** - Unless you know exactly what the request string is, otherwise always leave the field blank to return all requests or type the specific request name.
   - **Method/Component Trace** - Leave blank for all methods or type the specific 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.
8. Click **View Report** to view the report. 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**.
10. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining an MQI Analysis report

About this task

The MQI Analysis report provides the information for the MQI calls performance in the requests that were processed by the application server. You can also view the Trend report, Decomposition report, and detailed MQI report after you define the MQI Analysis report.

To define an MQI Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports > MQI.** The Create report page opens.
2. Select **Yes** or **No** to decide if you want the report to recur and click **Next**. For these instructions, we are selecting **No**. If you want further instructions on scheduling reports, refer to Defining a Schedule report.
3. From the list, select the group and the server on which you want to report.
4. To continue creating the report, click **Next**. 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**: Throughput per Second, Throughput per Minute, Throughput per Hour, and Response Time (ms).
   - **MQI Call**: BACK, BEGIN, CLOSE, CMIT CONN, CONNX, DISC, GET, INQ, OPEN, PUT, PUT1, and SET.
   - **Queue Manager**: Manager of the queue.
   - **Queue Name**: Name of the queue.
   - **Request Type/Transaction Type**: All, EJB, JSP, Servlet, CICS, Web Service, OTMA, VTAM, BTAM, APPC, Portal, and RMI-IIOP.
   - **Request/Transaction Name**: Unless you know the exact request string, always leave the field blank to return all requests or type the specific request name.
   - **Method/Program/Component Trace**: Leave the field blank to return all methods. Type the specific method name if you know that the method name that you are looking for.
6. To continue creating the report, click **Next**. 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.
8. Click **View Report** to view the report. 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**.
10. To save the report, click **Save**. For more information about how to view reports, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Lock Analysis report

About this task

If you suspect that you have lock contention issues with a specific application, you can run a lock analysis report. This report gives you the history of locking in the application and shows whether there is a trend. The lock reports show you the difference between the total locks and the ones that are contentious.

To define a Lock Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports > Lock Analysis**. The Create report page opens.
2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting no. If you want further instructions on scheduling reports, refer to Defining a Schedule report.
3. Select the group and the server from the box.
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 that you want to measure: Number of Lock Acquisitions, Number of Lock Contentions and Total Acquisition Time.
   - **Request Type** - All, EJB, JSP, Servlet, Web Service, and Portal.
   - **Request Name** - Unless you know the exact request string, always leave the field blank to return all requests or typethe specific request name.
   - **Method/Component Trace** - Leave the field blank to return all methods or typethe specific method 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.
8. Click **View Report** to view the report.
9. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a Portal Page report

About this task

The Portal Page report provides a picture of the behavior within the Portal Page requests. Once you isolate the problem requests, the portal-specific nested requests allow you to quickly get a high-level picture of where your problem is taking place. After defining the portal page report, several reports become available: Trend report, Decomposition report, and Portal Page Detail report.

To define a Portal Page report:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports > Portal**. The Create report page opens.
2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting no. If you want further instructions on scheduling reports, refer to Defining a Schedule report.
3. Select the group and the server from the box.

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 that you want to measure: Throughput per hour, Throughput per minute, Throughput per second, Response Time and CPU Time.
   - **Nested Request Type** - Portal page and Portlet. You must set this to Portal Page for a Portal Page report.
   - **Portal Page Name** - Limits the requests that are being reported on to include only requests whose name matches the string you enter.
   - **Portlet Name** - Type a specific portlet name to show only those portal pages that contain the portlet name specified.
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.
8. Click **View Report** to view the report.
9. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a Portlet report

About this task

To compare the response time of a portlet across multiple pages, or debug an application rather than the entire portal server, you can run a portlet report. This shows the performance of a portlet and not the whole request. Then you can see if the same portlet has different response times on different pages. Drilling down allows you to get detailed performance information about a per-portlet basis. You can see how a portlet performs across the different pages that it is used on. Doing this can help to determine whether the problem you are encountering is caused by the portlet itself or by the things surrounding it on the page.

To define a Portlet report:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports > Portal**. The Create report page opens.
2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting no. If you want further instructions on scheduling reports, refer to Defining a Schedule report.
3. Select the group and the server from the box.
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 that you want to measure: Throughput per hour, Throughput per minute, Throughput per second, Response Time and CPU Time.
- **Nested Request Type** - Portal page and Portlet. You must set this to Portlet for a Portlet report.
- **Portal Page Name** - type a specific portal page name to show only those portlets which are contained in the portal pages whose name matches the one you specified.
- **Portlet Name** - Type a specific portlet name to show only those portlets that contain the portlet name specified.

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.
8. Click **View Report** to view the report.
9. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining Top reports

About this task

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:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports > Top Reports**. The Recurrence report page opens.
2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting no. If you want further instructions on scheduling reports, see Defining a Scheduled report.
3. Select the group and the server from the lists. The Report and Data Range selection page opens.
4. Select the type of Top report that you want to run and set the date range using the lists. If applicable, set the Advanced Filtering to extract the data of a specific time period. For detailed instructions, see step 2 of Understanding the Date Range settings.
5. Click **View Report** to open the report. The Top report opens.
6. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Results

Important: The Top Slowest Request Report calculates the average response time by (sum of the response time)/(total # of requests) given the selected group/server and time period. This might cause the data in the Slowest Request Report to vary for the group report and server report for the same time span.

Related topics

Defining a Scheduled report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a System Resource Analysis report

About this task

The System Resource Analysis report gives you the information of the utilization of the memory, and the connection pool for the application servers. You can also view a Trend report and Decomposition report after defining the System Resource Analysis report.

Restriction: This feature does not apply to the z/OS platform.

To define a System Resource Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis > Create Server Reports > System Resource**. The Create report page opens.
2. Select yes or no to decide if you want the report to recur and click **Next**. For these instructions, we are selecting no. If you want further instructions on scheduling reports, see Defining a Schedule report. The Server and Report Type selection page opens.
3. Select the group and the server on which you want to report from the box.
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** - Amount of memory used, JVM/Region CPU %, System CPU %, Average % of Pool in Use (supported in ITCAM J2EE WebLogic), JDBC Connection Pool Size (supported in ITCAM J2EE WebLogic), and Live Sessions.
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.
8. Click **View Report** to view the report. 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**.
10. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a Server Availability Analysis report

About this task

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 in the group are up, divided by the total elapsed time.

To define a Server Availability Analysis report:

Procedure

1. From the top navigation, click **Performance Analysis > Create 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 these instructions, we are selecting No. If you want further instructions on scheduling reports, see Defining a Schedule report.
3. Select the group and the server on which you want to report from the box.
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.
6. Click **View Report** to view the report. 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** .
8. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a Capacity Analysis report

About this task

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:

Procedure

1. From the top navigation, click **Performance Analysis > Create Server Reports > Capacity Analysis**. The Server Selection page opens.
2. Select the group and the server on which you want to report from the box.
3. Click **Next** to continue creating the report.
4. Set the following options to filter the records returned in the report:
   - **X - Axis** - Throughput per Minute and Users.
   - **Y - Axis** - System CPU (%), JVM/Process CPU (%), JVM/Process Memory (MB), and Response Time (ms).
5. Click the check box to select Set Y-axis Max and enter the value that you want to be set to the maximum.
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.
8. Click **View Report**.
9. Click **Save** to save the report.
10. To save the report, click **Save**. For more information, see "Viewing saved reports" on page 177.

Related topics

Understanding the Date Range settings

Defining a Schedule report

E-mailing a report

Exporting a file

Modifying a report

Viewing a PDF file

Defining a Schedule report

About this task

Scheduling reports allows you to create a time for your reports to automatically activate at a time you preselect.

To define a Scheduled report:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports > Method/Program**.

   Important: We are going to use the Method/Program report as an example. You can create a Scheduled report from a selection of the available reports.
2. Select **Yes** to have the report recur and click **Next**. The Server Selection page opens.
3. Select the group and the server on which you want to report from the lists. Select the **Create a separate report for each server** check box to create one report for each server without repeating the steps for each one. If you do not select the check box, the report provides all the data aggregated across all servers.

4. Click **Next** to continue creating the report.
5. Set the following options to filter the records returned in the report:
   - **Metric** - Throughput per Second, Throughput per Minute, Throughput per Hour, Response Time (ms), and CPU Time (ms).
   - **Method/Component Trace** - Unless you know the exact string, always leave the field blank to return all traces or type the specific method/program/component name.

     Note: The name of this field represents the names used in J2/WAS and z/OS. Method name is used in the J2/WAS environments, and Program and component name are used in the z/OS environments.
   - **Request Type** - ALL, EJB, JSP, Servlet, CICS, Portal, and Web Service.
   - **Request Name** - Unless you know the exact request string, always leave the field blank to return all requests or type the specific request 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 on setting the parameters, see Understanding the Date Range settings.
8. To view the report, click **Preview**. To set a schedule for the report, click **Schedule**.
9. Set the schedule for the report and setup an email distribution list of the people you want the report sent to when it is completed. Unlike a saved report, you cannot click Run Report to view the report at anytime. For a scheduled report, you can only view the PDF of the report that was sent by email. Click **Next**.
10. Enter the name of the report. Click **Save** or **Save & Activate**. For more information, see "Viewing saved reports" on page 177.

    Tip: You can either save the report now and activate it later or you can save and activate the report at the same time.
11. The Scheduled reports page opens displaying your report in the list.

Related topics

E-mailing a report

Exporting a file

Modifying a report

Understanding the Date Range settings

Viewing a PDF file

Understanding date range settings

About this task

Date Range settings allow you to set date and time parameters to restrict the data that you want to capture for a report. You can modify these settings while creating a report. The Date Range settings contain three main sections: **Date Range**, **Advanced Filtering (optional)** and **Graphing Option**.

This task is not applicable to defining a Top report.

Procedure

To set the Date Range settings:

1. In the Date Range section, either click the **Preset** list to select a preset date range, or enter a custom start date and end date for the period from which you want to extract data.
2. To extract data for a specific time period, define the custom data set in the Advanced Filtering section by taking the following steps:
   a. Uncheck the hours of the day for which you do not want the data to be extracted. For example, to select only data for transactions occurring between 9:00am and 5:00pm, uncheck 00:00-08:00 hours and 18:00-23:00 hours.
   b. Uncheck the days of the week for which you do not want the data to be extracted. For example, to select only data occurring Monday through Friday, uncheck Sunday and Saturday.
   c. Uncheck the days of the month for which you do not want the data to be extracted.
   d. Uncheck the months of the year for which you do not want the data to be extracted.

   By default, the Advanced Filtering section selects all the options.
3. Select any of the following options in the Graphing section to analyze certain patterns in the data based on time characteristics, or if you want to compile large amounts of data over a long period and plot all the points:
   - Time series in minutes
   - 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
4. Optional: when viewing the trend report, save the date range. To be able to save the date range, you must select one of the following preset date ranges in step 1:
   - Last 60 Minutes
   - Last Complete Clock Hour
   - Previous 24 Hour
   - Last Day
   - Previous 7 Days
   - Last Week
   - Last Month
   - Month to Date

   When viewing the report, select the **Save** option from the menu. In the **Save Report** window, select the **Save preset date range** checkbox.

Results

On the Trend report, if the date range selected is 60 minutes or lower, both the graph and data table displays. If the date range selected is higher than 60 minutes, only the data table dispalys.

If you have opted to save the date range (see step 4 on page 172), when your saved report is re-executed, ITCAM reevaluates the report results based on a date range formula. For example, the report displays the last 60 minutes of data based on the current time and the most recent data in the database. If you have saved a report without selecting the **Save preset date range** box, ITCAM evaluates the report using the original date and time, and the report presents the same historical data.

Related topics

Defining a Schedule report

E-mailing a report

Exporting a file

Viewing a PDF file

Viewing the Detail report

About this task

The Detail report allows you to drill down into the data for more information about the requests. Each detail report provides detail, summary, worst performers, and locks information for your review.

To view the detail report:

Procedure

1. Create a report.
2. View the report.
3. Access the Trend report and click any part of the graph/chart to go to the Decomposition report.
4. From the Decomposition report, drill down into the detail report.
5. The Detail report has four tabs: detail, summary, worst performers, and locks.
   - The **Detail** tab provides information about each request that made up the data point selected from the decomposition report.
   - The **Summary** tab provides information about requests across all requests breaking it down by nested requests so you can view the data component by component.
   - The **Worst Performers** tab provides information about the requests containing the worst-performing nested requests based on the selected metrics.
   - The **Lock** tab provides detailed lock information with the ability to toggle between the lock acquisition and contention information. Lock Tab is only available for the Lock Analysis report. This data is the average for all the requests on the Detail tab of the Detail report.

     Tip: The following reports can be viewed in PDF format: Trend report, Decomposition report, Detail report, and Flow View in Trace report. All these reports can be mailed to users in PDF format and also these reports can be exported to a file in CSV format.

     Important: The detail report is available for all application reports except Top reports.

Related topics

Defining a Schedule report

Understanding the Date Range settings

Defining a Schedule report
E-mailing a report
Exporting a file
Modifying a report
Viewing a PDF file
Defining a Lock Analysis report
Defining a Method/Program Analysis report
Defining a Portal report
Defining a Request/Transaction Analysis report
Defining a SQL Analysis report
Defining an MQI Analysis report

Viewing the Trace Report
About this task

The Trace report allows you to drill down into the method flow of a selected request from the Detail report, in terms of the method/component entry and exit events. Each trace report provides the following options: Nesting Summary, Drilldown View, Flow View, and Search tabs to view method trace data in different formats.

To view the trace report:

Procedure

1. Create a report.
2. View the report.
3. Access the Trend report and click any part of the graph/chart to go to the Decomposition report.
4. From the Decomposition report, drill down into the detail report.
5. The Detail report, Detail tab, click Request/Transaction Name, drill down into the trace report.
6. Inside the Trace report, there are four tabs: Nesting Summary, Drilldown View, Flow View, and Search.
   - The **Nesting Summary** tab provides information about the top 10 slowest components. It also provides the total number of calls, the average response time, and the average CPU time for each component in the selected request.
   - The **Drilldown** tab provides information about method trace at each level.
   - The **Flow View** tab provides the complete method flow of the selected request.
   - The **Search** tab allows you to specify any of the following types, together with a numeric threshold (or a string) and presents a list of events from the method trace whose metrics cross the threshold (or match the string). The Event Type and Event Data searches are case-sensitive.
     - Elapsed Time
     - CPU Time
     - Delta Elapsed Time
     - Delta CPU Time
     - Event Type
     - Event Data

– Total Acquisition Time

Important: The trace report is available for Request/Transaction report, Lock Analysis report, and Portal report.

Related topics

Defining a Lock Analysis report

Defining a Portal report

Defining a Request/Transaction Analysis report

Report management

Manage saved reports.

Manage the reports you save on your system: view the reports (click the Report Name) or run the reports (click Run Report).

Modifying a report

About this task

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:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **Modify** next to the report that you want to change. The Recurrence page opens. Select Yes to have the report recur.
3. Change the group or server, and click **Next**. The Report Filtering options page displays different options based on the report type you select.

   While you are choosing a server by navigating through the groups, note 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 compiles all records that are generated by the chosen server regardless 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.
7. Click **View Report** to view the report. 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.
9. To save the report, click **Save**. For more information, see “Viewing saved reports” on page 177

   Related topics

   Understanding the Date Range settings

Defining a Schedule report

Running a report

Modifying a Top report
About this task

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:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports** . The Reports page opens.
2. Click **Modify** next to the top report that you want to change. The Recurrence page opens. Select Yes to have the report recur.
3. Change the group or server, and click **Next**. While you are choosing a server by navigating through the groups, note 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 compiles 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 box.
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 step 2 of Understanding the Date Range Settings.
7. Click **Finish** to create the report. The Top report opens.

Related topics

Understanding the Date Range settings

E-mailing a report

Exporting a file

Running a report

Viewing a PDF file

Running a report
About this task

Return to the Performance Analysis and Reporting Management page to run a saved report and retrieve the current data. Additionally, you can save a report, send a link or PDF of a report by email, view PDF report, or export a PDF report to a comma delimited file format. If you send a link by email, remember that the recipient must be a user of the application monitor with the appropriate rights to view the servers to where the report runs.

To run a report:

Procedure

1. From the top navigation, click **Performance Analysis > 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 that you want to run.

   Important: 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.

   Related topics

   E-mailing a report

   Exporting a file

   Viewing a PDF file

   Viewing the reports

Duplicating a report
About this task

Save time creating new reports by duplicating an existing report.

To duplicate a report:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **Duplicate** next to the report that you want to duplicate.
3. Enter a name for the duplicated report, and click **Save**. The Reports page displays with the new duplicated report.

   Related topics

   E-mailing a report

   Exporting a file

   Viewing a PDF file

Viewing saved reports
About this task

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 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** - displays Trend, Decomposition, Request Detail, and Trace reports.
- **Method/Program** - displays Trend, Decomposition, and Method Detail reports.
- **SQL** - displays Trend, Decomposition, and SQL Detail reports.
- **MQI** - displays Trend, Decomposition, and MQI Detail reports.
- **Lock Analysis** - displays Trend, Decomposition, and Lock Analysis Detail reports.

- **Portal** - displays Trend, Decomposition, and Portal Page and Portlet Detail reports.
- **Server Availability** - displays the Trend report.
- **System Resources** - displays Trend, and Decomposition reports.
- **Capacity Analysis** - displays Capacity Analysis Detail report.

To view the reports:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **Run Report** next to the report that you want to run. The Trend report opens first.

   Tip: Use the left navigation to return to the Saved Reports page, Modify reports, or Save a report.
3. Select an option from the Additional Details box to decompose the Trend report.
4. Click the bar displayed in the graph or a data point in the table to view more details. The Decomposition report opens.
5. Click a section of the chart or a data point in the table to view more details. The Request/Transaction Report Detail page opens displaying the Detail data, the Summary, and the Worst Performers.

Results

If you selected Request/Transaction Analysis as the Report Type, to access the Trace report:

1. Click the Request Name to view the Trace report.
2. The Trace report page - Nesting Summary page opens. For more information, see “Viewing the Nesting Summary” on page 198.

   Note: These instructions apply to all the report types available. However, remember that the reports available depend on the Report Type selected. Top reports have no additional detail.

   Related topics

   Defining a Capacity Analysis report

   Defining a Request/Transaction Analysis report

   Defining Top reports

Deleting a report
About this task

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:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **X** next to the report that you want to remove.

3. At the confirmation box, click **OK** to delete the report. The Reports page displays without the deleted report.

Related topics

E-mailing a report

Exporting a file

Viewing a PDF file

E-mailing a report

About this task

You can send a PDF file of a report by email to either the application monitor users or non-application monitor users. You can also send a link of a report by email to a group of application monitor users. The recipient will be brought to a particular page by the link after logging in. You can send a PDF of the following reports by email: Trend report, Decomposition report, Detail report, and Flow View in Trace report

To e-mail a report/PDF:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **Run Report** next to the report that you want to run. The selected report opens.
3. Click the **E-mail 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.

Results

To e-mail a link:

1. From the top navigation, click **Performance Analysis > View Saved Reports.**

   The Reports page opens.
2. Click **Run Report** next to the report that you want to run.

   The selected report opens.
3. Click the **E-mail Link** to email a link of a report.

   The E-mail page opens.
4. On the E-mail page, enter the email address of the recipient. Separate multiple addresses with a comma.
5. Click **OK** to email the link of the report. When you email a link, the recipient must be a user of the application monitor with the appropriate rights to view the servers in the report.

   Related topics

   Exporting a file

   Viewing a PDF file

Viewing a PDF file
About this task

You can view a PDF file of a report before you send out the file to another recipients.

You can view a PDF of the following reports: Trend report, Decomposition report, Detail report, and Flow View in Trace report

To view a PDF file:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **Run Report** next to the report that you want to run. The selected report opens.
3. Click **View PDF** 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.

Related topics

E-mailing a report

Exporting a file

Exporting a file
About this task

You can export a report to a comma delimited file format, if necessary. You can export a PDF of the following reports: Trend report, Decomposition report, Detail report, and Flow View in Trace report

To export to a file:

Procedure

1. From the top navigation, click **Performance Analysis > View Saved Reports**. The Reports page opens.
2. Click **Run Report** next to the report that you want to run. The selected report opens.
3. Click **Export to File**.
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.

Related topics

E-mailing a report

Viewing a PDF file

Method Profiling

Use the Method profiling pages to create, view, and delete method profile reports.

View the method profile reports that have been run and view their details. Delete reports you no longer want to save.

See “Viewing Method Profiling Management” on page 181.

Viewing Method Profiling Management
About this task

The Method Profiling Management page provides a list of all the method profiles collected and stored in the method profile reports generated as a result. From this page, you can delete method profile reports and view data about the reports.

To collect method profiles you are required to use L2 and select method profiling, see "Overriding a monitoring level" on page 81.

To open Method Profiling Management:

Procedure

1. From the top navigation, click **Performance Analysis > Method Profiling**. The Method Profiling Management page opens.
2. Select the group and the server whose method profile that you want to view. If there are no method profiles on that server "Data not available." displays.
3. Click the Date/Time link to view the data for the report. The report opens and displays the method names contained in the method profile as well as the Total CPU Time, Total Elapsed Time, Total Hits, Avg. CPU Time, and Avg. Elapsed Time.
4. To delete the report, click **Delete**
5. Click **OK** to finish deleting the report.

Related topics

Activating Method Profiling

"Monitoring on Demand (TM)" on page 77
In the Monitoring on Demand (MOD) pages, you can view and adjust the monitoring level for all servers, and schedule adjustment of this level at fixed times.

Activating method profiling
About this task

Method profiling is a new feature that is part of L2 monitoring. You can activate L2 with method profiling by selecting the monitoring level, overriding the monitoring level, or creating a schedule that selects the monitoring level. Links to the instructions to perform these actions are available in the following related topics.

Procedure

1. To activate method profiling by creating a schedule, see "Creating a schedule" on page 79 for more information.
2. To activate method profiling by overriding the monitoring level, see "Overriding a monitoring level" on page 81 for more information.
3. To activate method profiling by configuring the data collection settings, see "Configuring the Data Collection settings" on page 83 for more information.

Related topics

"Monitoring on Demand (TM)" on page 77
In the Monitoring on Demand (MOD) pages, you can view and adjust the monitoring level for all servers, and schedule adjustment of this level at fixed times.

Viewing Method Profiling Management

Daily Statistics

The Daily Statistics pages provide daily SMF information snapshots for z/OS WebSphere servers.

The Daily Statistics page provides a list of daily SMF statistics reports available for a date. Initially, it shows the server reports for the previous day.

To view a report, click the server name. See "Viewing the Daily Statistics Overview."

To view reports for a different date, change the date in the **Enter Report Date** controls and click the **Go** button.

To delete all daily statistics report created before a certain date, set this date in the **Delete Reports Older Than** controls and click the **Delete** button. See "Deleting Daily Statistics" on page 183.

Viewing the Daily Statistics Overview

About this task

The Daily Statistics Overview page provides daily SMF information snapshots for z/OS WebSphere servers only. This data is shown for the server selected on the Daily Statistics page.

To open the Daily Statistics Overview page:

Procedure

1. From the top navigation, click **Performance Analysis > Daily Statistics**. The Daily Statistics selection page opens displaying data from the previous day.
2. To view snapshots from a different date, from the left navigation pane, select a month, day, and year under the **Enter Report Date** heading and click **Go**.
3. Click a server name to view the Daily Statistics Overview, where the snapshot data displays.

Results

Tip: The side navigation can be used to find out more information regarding: Server, EJBs, Servlet Session Manager, Server Regions, SQL, JCA-CICS, and Web Applications.

To change the date of the report:

1. On the left navigation pane, select a month from the box.
2. On the left navigation pane, select a date from the box.
3. On the left navigation pane, select a year from the box.
4. Click **Go**.

Related topics

System Resources Metrics

Viewing the System Resources Browser

Deleting Daily Statistics

About this task

You can permanently delete daily statistics by purging the system.

To open the Daily Statistics page:

Procedure

1. From the top navigation, click **Performance Analysis > 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.
3. Click **Delete**.
4. Click **Yes** in the confirmation box. The system deletes all reports created earlier than the date you select.

Related topics

System Resources Metrics

Viewing the Daily Statistics Overview

Custom requests

Custom requests are defined for application-specific operations that do not fall under the normal pre-defined J2EE operations. Use custom requests to track specific application operations as separate requests. For example, if the application is completing some well-defined processing like parsing documents, or CPU-intensive numeric calculation, it can be useful to track this operation as a separate request.

When you use custom requests, you can set the monitoring level to L2 rather than L3. L3 generally applies to a large set of methods, typically every application method, as a result L3 monitoring processor usage is high. Custom requests are defined for specific application methods, and the processor usage is much lower. Also, custom request definitions are not limited to application methods, you can define specific system methods as custom requests.

Types of requests

Request

A request is a call made to a component that performs a service. For example, a call to execute a SQL statement through a JDBC driver or a data source. The following list provides some of the typical J2EE requests that are interesting to monitor, they are all standard J2EE requests that are monitored by the data collector, with no extra custom definitions required:

- Invoking a Servlet through the doGet() or doPost() methods
- EJB create() and other business methods
- JDBC API invocations like getConnection() or executeQuery()
- JMS operations to send and receive messages
- JNDI operations like lookup() used to find objects stored in the registry
- JCA resource adapter operations like getManagedConnection()

Edge request

The outermost request that needs to be monitored is called an *edge request*. This involves a request that enters the application server from an outside client, for example invoking a Servlet/JSP or a call to an EJB business method through the Object Request Brokers (ORB). However, sometimes this request might not be of interest as it might be a generic call like when Apache Struts applications are involved. In such cases, a custom request might be defined to act as an edge request. In ITCAM for Application Diagnostics, edge requests are tracked starting at L1 monitoring level.

Nested request

A request that is invoked within another request is called a nested request. For example, a JDBC call like getConnection() invoked from within an EJB business method is a nested request. In ITCAM for Application Diagnostics, nested requests are tracked starting at L2 monitoring level.

Custom request

Custom requests are defined for application-specific operations that do not fall under the normal pre-defined J2EE operations. A custom request defines a user-specified class and method as the start and end point of a request. If it is enclosed in another request like a Servlet or EJB request, then it becomes a custom nested request.

Creating custom requests

To create a custom request, complete the following steps:

1. Edit the *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/toolkit_custom.properties file and uncomment the following line:

   ```
   am.camtoolkit.gpe.customxml.custom=/opt/IBM/itcam/WebSphere/DC/itcamdc
        /etc/custom_requests.xml
   ```

   You can specify a different location and file name.

   Note: If the custom_request.xml file resides in the *DC_home*/runtime/*app_server_version.node_name.server_home*/custom/ directory, then you can specify the simple file name, without having to specify the fully qualified path.

2. Create or edit the custom_requests.xml - it is typically easier to make a copy of the file and edit it to add the custom request definitions. The following example shows an XML specification for a custom request:

   ```
   <gpe>
     <bci>
       <customEdgeRequests>

         <edgeRequest>
            <requestName>MonteCarlo</requestName>
                  <Matches>com.myco.investment.modeler.Simulator</Matches>
                  <type>application</type>
            <methodName>executeMonteCarlo</methodName>
          </edgeRequest>

        </customEdgeRequests>
     </bci>
   </gpe>
   ```

   The following table explains the XML elements in the custom_requests.xml file.

Table 5. XML elements in custom_requests.xml

Tag Name	Description
edgeRequest	Identifies one or more application methods that are to be monitored for custom request processing. By modifying the requestName, Matches, type, and methodName tags within the edgeRequest tag, you can customize the selection. Each edgeRequest tag must contain exactly one methodName tag, and one or more Matches tags. Multiple edgeRequest tags can be specified.
requestName	Defines a unique name for this request. The request name is displayed in the L1 or L2 trace entry that is produced when one of the methods identified by this custom request runs.
Matches	Identifies a class or classes that contain the methods that are to be defined as custom requests. Multiple Matches tags can be present within a single edgeRequest tag.
type	Indicates whether the classes specified are loaded by system or application class loader. If the classes are present within an application EAR file, then the type is "application". However, in rare cases, the classes might be present in JAR files specified in ws.ext.dir, system class path or even in bootstrap class path. In such cases the type is "system".
methodName	Identifies the names of the methods within one of the classes identified by the Matches tag that are to be monitored for custom request processing. One methodName tag can be specified in each edgeRequest tag.

The Matches and the methodName tags can include wildcard characters. The following list is a summary of the wildcard functionality:

- Asterisk (*) stands for zero or more occurrences of any character when used by itself. When embedded within a sequence of characters, for example, java.*.String, it matches zero or more occurrences of any character except the package separator (.).
- Two periods (..) can be used to specify all subpackages, for example, java..String matches java.lang.String. It matches any sequence of characters that starts and ends with the package separator (.).

3. After defining the custom requests, the application server JVM needs to be recycled for them to take effect.

Making a custom request invoke an edge request

If the class and method defined in a custom request definition are invoked from within another request, then the custom request becomes a nested request. It is necessary to turn on the L2 monitoring to view these requests in the ITCAM MSVE console.

For example, in Model-View-Controller(MVC) architectures, the controller receives all the requests from the clients. Based on the content of the request, it then redirects to the appropriate model. A well-known implementation of the MVC architecture is the Apache Struts framework that is widely used in the J2EE applications.

In Apache Struts, the controller is the ActionServlet and it receives all the requests. The ActionServlet interprets the URL and, based on the Apache Struts configuration files, it gives the handling to one of the Action classes written by the user. When monitoring Apache Struts applications, the user is interested in making these Action classes the edge request instead of the ActionServlet. These Action classes receive all the requests so that different types of URLs used by the application can be tracked.

This can be done by defining each of the Action classes as a custom request. However, since the ActionServlet is invoked first, these action classes are inside an already created edge request and hence can only be tracked as nested requests.

To make the Action classes an edge request, it in necessary to prevent the Apache Struts ActionServlet from creating the edge request. This is done by preventing the normal Struts action servlet from being considered an edge request by the data collector, by excluding the action servlet class from instrumentation.

Excluding classes from instrumentation

Complete the following steps:

1. Edit the configuration file *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/toolkit_custom.properties to add the following new property:

   ```
   am.camtoolkit.gpe.customxml.exclude=excludes.xml
   ```

2. Create the file excludes.xml in the same custom directory with the following content:

   ```
   <gpe>
     <bci>
       <classExcludes>
         <exclude>org.apache.struts.action.ActionServlet</exclude>
         <exclude>com.company.package.*</exclude>
       </classExcludes>
     </bci>
   </gpe>
   ```

3. Add as many classes as needed.
4. Restart the application server.
5. To verify that the class has been excluded, look in toolkit.xml. It is displayed as follows:

   ```
   <classExcludes>
         <include>*</include>
         <include>org.eclipse.osgi.framework.adaptor.core.*</include>
         <exclude>com.company.class</exclude>
         <exclude>com.company.package.*</exclude>
         <exclude>com.sun.net.ssl.internal.ssl.JSA_RSAKeyFactory</exclude>
         <exclude>COM.rsa.jsafe*</exclude>
         <exclude>org.eclipse.osgi.*</exclude>
       </classExcludes>
   ```

   Important: The toolkit.xml file contains runtime settings and it is refreshed every time the application server is restarted

Viewing custom requests

Viewing recent custom edge request

1. From the top navigation, select **Problem Determination** - > **Server Activity Display**.
2. Select the **Recent Requests** tab. Custom edge requests are identified in the **ThreadType** field.

Viewing Active Custom Requests and Nested Custom Requests

1. From the top navigation, select **Problem Determination** - > **Server Activity Display**.
2. Select the **Active Requests** tab. Custom edge requests are identified in the **Last Know Action** field.

3. To view nested custom requests, click the edge request link name and select **Method/Component Trace**.
4. In the **Complete Flow View**, details of the nested custom request are displayed.

Viewing custom requests in reports

Custom requests can be seen in the following reports:

- Custom edge requests are displayed in the Decomposition report, in the Decomposition Data Table section.
- Nested custom requests are displayed in the Trace report in the Nesting Summary tab, see "Viewing the Nesting Summary" on page 198

Composite requests

Use the Composite Request features in the MSVE to monitor transactions that use resources on more than one server.

The Composite Request features help you to:

- Determine if the reason a top-level request is hanging 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.

There are a number of areas of MSVE that help you to locate, view, and analyze composite requests.

Table 6. Composite Request functionality

Area of functionality	Description of functionality
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 and Reporting	Locate completed requests/transactions
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

Each of these features produces a list of requests/transactions which might participate in a composite request. The presence of the composite request icon indicates that a request/transaction participates in a composite request: 

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 and Reporting for examples of the airline reservation application, some of which have the composite request indicator. Clicking the indicator brings you to the composite request view of the Method Trace. This enables you to navigate among these requests, so that you can see which application calls which one, and by what mechanism (MQ, CTG, or DPL). You can send a PDF of each request involved in the composite transaction to your manager by email.

The scope of composite requests

To understand the scope of what ITCAM for Application Diagnostics can monitor, it is important to understand two terms: managed space and composite request space.

- Managed space is the entire scope of what ITCAM for Application Diagnostics can monitor. Since ITCAM for Application Diagnostics can monitor servers and application servers, along with applications and J2EE components like EJB, the managed space has many dimensions.
- Composite request space is a subset of the managed space. Composite requests are requests that conform to an Enterprise Application Integration (EAI) architecture.

Managed space

The basic model of ITCAM for Application Diagnostics is to have a single managing server and many data collectors. The data collectors are dynamically controlled through the Managing Server. The data collectors deliver their collected data to the Managing Server.

The Managing Server is the heart and brain of ITCAM for Application Diagnostics. It is the entity to which each of the many data collectors communicate, and provides the ITCAM for Application Diagnostics User Interface.

The data collectors are the eyes and ears of ITCAM for Application Diagnostics. For each application server that is being monitored, a data collector is deployed on the computer hosting the application server. (If a server has two application servers, then you must configure two data collectors on the server in order to monitor both application servers.)

The following table describes what is in the managed space:

Table 7. Components of the managed space

Component	Description
Servers	Any server on which a data collector is installed is in the managed space. For z/OS systems, a server is considered to be equivalent to an LPAR.

Table 7. Components of the managed space (continued)

Component	Description
Application Servers	Any application server running in a JVM in which a data collector is configured is 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 with a control region and one or more servant regions. The definition and the regions are called an application server instance. What ITCAM for Application Diagnostics 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). In most cases, the application server is an individual application servant region.
Resources	ITCAM for Application Diagnostics 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.
Application	ITCAM for Application Diagnostics 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 is in the managed space. As a corollary, stand-alone applications, which are not served through an application server, are not in the managed space.

Composite request space

Although the managed space includes servers, application servers, requests, and resources, the composite request space includes only a subset of the requests in the managed space.

In order to define the composite request space and understand how requests interact, it is important to understand EAI architecture.

EAI is the term used to describe the integration of the computer applications in an enterprise to maximize their utility throughout the enterprise. Typically, an enterprise has earlier single purpose applications and databases, and wants to continue to use them while adding or migrating to a new set of applications. The new applications use the Internet, e-commerce, extranet, and other new technologies. EAI might involve developing a new total view of the applications in an enterprise. This can involve seeing how existing application fits into the new view, and then devising ways to efficiently reuse what exists while adding new applications and data. From the J2EE perspective, this means that an initial request, served by a J2EE application server, might start a resource on an earlier single purpose system through the JCA API.

When describing EAI transactions, the name used for the initial J2EE request is the *home request*. The server on which the transaction occurs is called the *home server*. The transaction on the earlier single purpose system is called a *participating request*, and the server is called a *participating server*. There can be more than one participating request if the earlier single purpose application invokes resources on other single purpose applications.

In ITCAM for Application Diagnostics operations, both the home request and the participating requests are displayed. However, without the composite request enhancement, these requests are displayed independently, and no explicit indication that they are part of the same transaction is given. The composite request enhancement makes this relationship explicit and also provides diagnostic tools, like Method Trace and Stack Trace, that you can apply across all requests in the composite request.

Composite requests involving CICS and IMS systems

If a participating server is a CICS server, and a CICS data collector has been installed, then this system is in the managed space. Similarly, if a participating server system is IMS, and an IMS data collector has been installed, then the system is in the managed space. The J2EE application server is in the managed space if a data collector is installed.

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 various interfaces. However, ITCAM for Application Diagnostics does 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 are displayed in ITCAM for Application Diagnostics, they are not necessarily treated as part of a composite request, even if they start programs on other regions.

A transaction on a CICS or IMS region is 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 11 bytes of available space.
- For IMS: The application on the Home Server uses IMS Connect for Java (IC4J) to access IMS Connect.

If any of these criteria are not met for an EAI request, ITCAM for Application Diagnostics does not identify the request as being part of a composite request.

However, the core ITCAM for Application Diagnostics features are still 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 is displayed as a request inITCAM for Application Diagnostics. However, the C++ application request is not displayed in ITCAM for Application Diagnostics. The reason for this is that ITCAM for Application Diagnostics does not monitor C++ applications. In this case, ITCAM for Application Diagnostics does not identify the CICS transaction as part of a composite request.

Likewise, if a Java application uses EPI to access CTG, ITCAM for Application Diagnostics does 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 are displayed in ITCAM for Application Diagnostics, but are displayed independently, and are not identified as a composite request.

The final condition, based on the use by the application 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 is not room for this correlation information. In these exceptional cases, ITCAM for Application Diagnostics does not attempt to identify the EAI as a composite request, and the individual requests are displayed in ITCAM for Application Diagnostics as independent requests.

Multiple hops

Composite requests 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. ITCAM for Application Diagnostics 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 in an 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. Composite requests can include an unlimited depth of "hops". However, composite requests place a limit on the number of calls that can be tacked by any single request.

Configuring data collectors that use WebSphere MQ

If you are monitoring composite requests for applications that use WebSphere MQ as a mechanism to bridge J2EE and CICS or IMS, then you must configure each participating data collector to monitor WebSphere MQ.

Note: These instructions assume that your data collectors have already been configured.

To enable WebSphere 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. Click the **Configuration Library** link in the left navigation.

5. Locate the application server in the Associated Server column of the Configuration Library table and click the Modify icon for that row.

   Restriction: You cannot modify ITCAM for Application Diagnostics supplied default configurations. You can modify configurations that only you have created.
6. Select **Enable MQ**.
7. Enter the queues that you want to monitor in the **Exclude (Classname)** and **Exclude Override (Classname)**.
8. Click the **Save** button.

Locate, view, and analyze composite requests

There are a number of areas of ITCAM for Application Diagnostics that help you to locate, view, and analyze composite requests.

Table 8. Composite Request functionality

Area of functionality	Description of functionality
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 and Reporting	Locate completed requests/transactions
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

Each of these features produces a list of requests/transactions which might participate in a composite request. The presence of the composite request icon indicates that a request/transaction participates in a composite request: 

In-Flight Request Search

To search for in-flight requests/transactions that participate in composite requests, use the In-Flight Request Search. See “In-flight request search” on page 114

The results display: Server Name, Client Request/Transaction, Start Date Time, Thread ID, and Total Resident Time. In addition, ITCAM for Application Diagnostics identifies requests/transactions that are part of a composite request by displaying the composite request icon.

Server Activity Display

To search a server for resident requests/transactions that participate in composite requests, use the Server Activity Display. See “Server Activity Display” on page 116

The results display: Client Requests, Client Requests Start, Thread ID, Resident Time, Accumulated CPU, Idle Time, Thread Status, Last Known Class, Last Known Method, Last Known Action, and User ID. In addition, ITCAM for Application

Diagnostics identifies those requests/transactions that are part of a composite request by displaying the composite request icon.

Performance Analysis and Reporting

To search for completed requests/transactions that participated in composite requests, use Performance Analysis and Reporting. Create a Trend Report, then drill down to a Decomposition Report, and then to a Detail report to find individual requests/transactions that are part of composite requests.

Important: Performance Analysis and Reporting displays the Composite Request Indicator only for home requests, and not for the other participating requests/transactions. For details on participating requests and home requests, see "The scope of composite requests" on page 188

The following procedure describes how to locate composite requests using Performance Analysis and Reporting.

To locate composite requests using the Performance Analysis and Reporting:

1. View a Request/Transaction Analysis Trend Report for servers that you believe might 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.

The results display: Request/Transaction Name, Request/Transaction Type, Response Time, CPU Time, Server Name, Timestamp, and Number of Records. ITCAM for Application Diagnostics identifies that a request was a home request of a composite request by displaying the composite request icon next to its Request/Transaction Name.

Viewing composite requests

To access the composite request details of requests/transactions, click the Composite Request Indicator for that request/transaction.

Composite requests have the following features:

- 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.

The availability of method-level data depends upon the configuration of the data collectors. The data collectors must be at L3 monitoring level in order to provide full method-level data. To provide Nested Request data, the data collector needs to be at L2 monitoring level.

Because 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 might vary.

Authorization and composite requests

Authorization is enforced in ITCAM for Application Diagnostics 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 that a user has access to a feature, the server-based authorization might further limit access to data about servers based on which group 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.

The home request of a composite request is on server A (which is in group A) and invokes a participating request on server B (which is in group B). Two users must 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 are different. The results are different because 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 sees only request A, and user B sees only request B.

In both cases, the Composite Request Indicator is displayed next to the request, and links to a similar Composite Request Detail page. However, the contents of the Composite Request Detail page are different for each user.

Both users see the complete composite request, including the Home Request on server A and the Participating Request on server B. However, the users do not have access to the Request Detail pages of all requests: User A has access to the Home Request on server A (the request name is linked), but not to the Participating Request on server B (the request name is not linked). User B does not have access to the Home Request on server A (the request name is not linked) but has access to the Participating Request on server B (the request name is linked).

Viewing a Composite Method Trace - SAD

About this task

The Composite Method Trace page displays the method flow of the composite transaction, including the method traces of each individual request participating in the composite transaction.

To view a Composite Method Trace:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display selection page opens.
2. Select a group from the group box.
3. Select a server from the server box. The Server Activity Display page opens.
   The Server Activity Display page opens.

4. Click the composite transaction indicator next to the request that you want to view. The Composite Request Detail for that composite transaction opens.
5. On the left navigation pane, click Composite Method Trace. The Composite Method Trace page opens.

   For CICS, the line numbers and response codes have been added to the Composite Method Trace. The return code is from the method call and the line number is of the CICS method call. You can find their information in the Event Data field on the Flow View tab.

Results

Related topics

Canceling a request

Changing a thread's priority

Viewing a Composite Stack Trace - SAD
About this task

The Composite Stack Trace page displays the stack traces of each server involved in the composite transactions that are actively processing their request/transaction.

Tip: The Composite Stack Trace is useful for debugging composite transactions that are hanging, because stack trace data is unavailable if a composite transaction has completed by the time you access it.

To view a Composite Stack Trace:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display selection page opens.
2. Select a group from the group box.
3. Select a server from the server box.
4. Click the composite transaction indicator next to the request that you want to view. The Composite Request Detail for that composite transaction opens.
5. On the left navigation pane, click **Composite Stack Trace**. The Composite Stack Trace page opens and displays the stack traces of the servers that are actively executing participating requests.

   Related topics

   Canceling a request

   Changing a thread's priority

Viewing a Composite Request Detail - SAD
About this task

The Composite Request Detail page summarizes a composite transaction, in terms of the individual requests that participate in it.

To view Composite Request Detail:

Procedure

1. From the top navigation, click **Problem Determination > Server Activity Display**. The Server Activity Display selection page opens.

2. Select a group from the group box.
3. Select a server from the server box.
4. Click the composite transaction indicator next to the request that you want to view. The Composite Request Detail for that composite transaction opens.
5. Click the Request Name link to drill down and view the detailed information.

Related topics

Canceling a request

Changing a thread's priority

Viewing a Composite Request Detail - In-flight Request Search

About this task

The Composite Request Detail page summarizes a composite transaction, in terms of the individual requests that participate in it.

To view Composite Request Detail:

Procedure

1. From the top navigation, click **Problem Determination > In-flight Request Search**. The In-flight Request Search page opens displaying active requests participated in composite transactions that are identified by the composite transaction indicator.
2. Click the composite transaction indicator next to the request that you want to view its Composite Request Detail information. The Composite Request Detail for that composite transaction opens.
3. Click the Request Name link to drill down and view the detailed information.

Related topics

Canceling a request

Changing a thread's priority

Viewing a Composite Stack Trace - In-flight Request Search

About this task

The Composite Stack Trace page displays the stack traces of each server involved in the composite transaction that are actively processing their request/transaction.

Tip: The Composite Stack Trace is useful for debugging composite transactions that are hanging, since there is no stack trace data available if a composite transaction has completed by the time you access it.

To view a Composite Stack Trace:

Procedure

1. From the top navigation, click **Problem Determination > In-Flight Request Search**. The In-Flight Request Search page opens displaying active requests/transactions.
2. To view the Composite Method Trace for a request/transaction that is participating in a composite transaction, click that request/transaction composite transaction indicator. The Composite Method Trace page for that composite transaction opens.
3. Click **Composite Stack Trace** in the left navigation pane.

4. The Composite Stack Trace page opens and displays the stack traces of the servers that are actively executing participating requests/transactions.

   Related topics

   Canceling a request

   Changing a thread's priority

Viewing a Composite Method Trace - In-flight Request Search
About this task

The Composite Method Trace page displays the method flow of the composite transaction, including the method traces of each individual request participating in the composite transaction.

To view a Composite Method Trace:

Procedure

1. From the top navigation, click **Problem Determination > In-flight Request Search**. The In-flight Request Search page opens displaying active requests participated in composite transactions that are identified by the composite transaction indicator.
2. Click the composite transaction indicator next to the request that you want to view its Composite Request Detail information. The Composite Request Detail for that composite transaction opens.
3. On the left navigation pane, click **Composite Method Trace**. The Composite Method Trace page opens.

   Related topics

   Canceling a request

   Changing a thread's priority

Viewing a Composite Request Detail - PAR
About this task

The Composite Request Detail page summarizes a composite transaction, in terms of the individual requests that participate in it.

To view Composite Request Detail:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports >** select an application report type. For more information, see "Performance analysis and reporting" on page 160. After creating the report, click View Report and the Trend report will open.
2. Click the data in the graph that you are interested in viewing more information about. The Decomposition report opens.
3. Click the data in the Decomposition report that you want more information about. The Request Report Detail opens.

   Related topics

   Viewing the Detail report

   Canceling a request

   Changing a thread's priority

Viewing a Composite Method Trace - PAR

About this task

The Composite Method Trace page displays the method flow of the composite transaction, including the method traces of each individual request participating in the composite transaction.

To view a Composite Method Trace:

Procedure

1. From the top navigation, click **Performance Analysis> Create Application Reports >** select an application report type. For more information, see "Performance analysis and reporting" on page 160. After creating the report, click View Report and the Trend report will open.
2. Click the data in the graph that you are interested in viewing more information about. The Decomposition report opens.
3. Click the data in the Decomposition report that you want more information about. The Request Report Detail opens.
4. Click the item in the detail of the report that you need more information about. The Trace report opens.

   Related topics

   Viewing the nesting summary

   Viewing the method/component tract - depth drilldown detail

   Viewing the method/component tract - depth drilldown report

   Viewing a method/component trace - flow view

Viewing the Nesting Summary

About this task

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

To view the Nesting Summary page:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports**. Select an application report type. For more information, see "Performance analysis and reporting" on page 160.
2. Click the data in the graph that you are interested in viewing more information about. The **Decomposition report** opens.
3. Click the data in the **Decomposition report** that you want more information about. The **Request Report Detail** opens.
4. Click the item in the detail of the report that you need more information about The **Trace Report** opens.
5. Click the **Nesting Summary** tab. The Nesting Summary page opens. You must set the monitoring level of the data collector to at least L2 or L3 to capture data for the **Nesting Summary**.

   Related topics

   Defining a Capacity Analysis report

   Defining a Request/Transaction Analysis report

   Defining Top reports

Viewing a Method/Component Trace - Depth Drilldown detail

About this task

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

To view the Depth Drilldown detail of a method trace:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports**. Select an application report type. For more information, see "Performance analysis and reporting" on page 160.
2. Click the data in the graph that you are interested in viewing more information about. The **Decomposition report** opens.
3. Click the data in the **Decomposition report** that you want more information about. The **Request Report Detail** opens.
4. Click the item in the detail of the report that you need more information about The **Trace Report** opens.
5. Click the Drilldown View tab. The Drilldown View page opens and displays the **Depth Drilldown Detail**.

Results

To view the Depth Drilldown report, choose Depth Report from the box. See "Viewing a Method/Component Trace - Depth Drilldown report."

Related topics

Canceling a request

Changing a thread's priority

Viewing a Method/Component Trace - Depth Drilldown report

About this task

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.

To view the Depth Drilldown report of a method trace:

Procedure

1. From the top navigation, click **Performance Analysis > Create Application Reports**. Select an application report type. For more information, see "Performance analysis and reporting" on page 160.
2. Click the data in the graph that you are interested in viewing more information about. The **Decomposition report** opens.
3. Click the data in the **Decomposition report** that you want more information about. The **Request Report Detail** opens.
4. Click the item in the detail of the report that you need more information about The **Trace Report** opens.
5. Click the **Drilldown View** tab. The Drilldown View page opens to the **Depth Drilldown Detail.**
6. Choose **Depth Report** from the box. The **Depth Drilldown Report** opens.

Related topics

Canceling a request

Audit trails

Trace user actions.

Audit trails provide a means for tracing user actions in the system. This helps with both accountability and troubleshooting.

User Scenarios

Scenario 1: Verifying high server response time

Upon returning from vacation, you see that response time is higher than usual for one of the servers in group ABC. You notice from the Heap Dump Management page that the server is performing heap dumps regularly that are causing the slowness in the response time. You enter the audit trail to find out who scheduled the heap dump. You contact that person and learn that the heap dumps are scheduled for troubleshooting a suspected memory leak in the application.

Scenario 2: Verifying MOD level change

In your role as a production support engineer, you observe that the MOD level of a data collector in the production environment has been set to L2 instead of the expected MOD L1. You ask the Administrator to search the audit trail and find out who changed the MOD level, and find that an application support engineer is troubleshooting a production issue in the application.

Accessing the user audit trail

About this task

The user audit trail is a text file that contains a record of user activity. Activities include Date, Time, User Name for Login, Failed Login, Log Out, Time Out, Authentication, and Account Status changes.

To open the user audit trail:

Procedure

1. Depending on your platform, navigate to the logs directory where you installed the managing server, for example, /var/ibm/tivoli/common/CYN/logs.
2. In a text editor appropriate to your platform, open the **audit-ms-Compound.log** file.

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 Original Request String (ORS)

The application you monitor uses a distinct URI to represent each specific application function, such as log in, check out, or log out. You must 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 monitor 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.

Note: This feature is not available for IMS.

Data used by the Request Mapper

Request name

The Request Name enables 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 data includes the Request Object and the Session Object in the case of a URL-based request.

Application Name

The Application Name enables 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 enables you to analyze their historical data from an application perspective.

For example, requests can be mapped to the following names:

- Account Management
- Web Trading
- Order Management

User IDs

The Application Monitor can 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 is required. User IDs are defined as Web-side identifiers of who initiated the transaction/request.

To capture the user ID from the session, you must 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.http.userid.source=session
```

To capture the attribute called account name 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, the two major types of requests are: JSP and Servlet. These calls come either from a Web server, or from an application server other than itself.

This request, usually expressed in the form of a string, is called the ORS. The ORS is composed of the URI plus the query string.

A unique ORS can be used to represent a specific application function such as log in, check out, and log out, this might not always be the case. Other styles of application design use 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 might sometimes find that it is impossible or ineffective 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 enables the ORS to be rewritten into two other strings before it is passed on to ITCAM for Application Diagnostics:

- Request Name
- Application Name

If no request mapper is used, the Application Monitor maps the incoming ORS onto a Request Name and an Application Name using the following rule:

```
Request Name     = ORS without the host nameApplication 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 and Reporting performs decomposition by Application Name.

Configuring 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.

WebSphere 6.1 uses a different class loading mechanism than WebSphere 6.0 or WebSphere 5.1.1, therefore complete the following steps to configure the Request Mapper for WebSphere 6.1:

1. Stop the WebSphere server.
2. To configure a Request Mapper, complete the following steps:
   a. Assuming the request mapper classes are packaged in requestmapper.jar, create the Request Mapper class plug-ins and package them into the JAR file.
   b. In the datacollector_custom.properties file located in the *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/ directory, set the am.requestmapper property as follows:

      ```
      am.requestmapper= <fully qualified requestmapper class name>
      ```

      where *fully qualified requestmapper class name* is the Request Mapper class that implements the ITCAM Request Mapper interface and is packaged in requestmapper.jar.
   c. Put the library requestmapper.jar in *DC_home*/itcamdc/lib/ext
3. (Optional) The following steps are optional. They provide an example of how to configure a Request Mapper and avoid mixing the Request Mapper-specific properties with other JVM system properties. This is done by creating a separate Request Mapper properties file and including all the Request Mapper properties in this file. In this way, if you have to add additional Request Mapper properties, you can do so without exposing them to other code either in the data collector or in the application server. The following steps provide an example of this optional approach:
   a. Create a property file called requestmapper.properties and put all the Request Mapper-specific properties in this file. Put the requestmapper.properties file in *DC_home*/runtime/*DC_specific dir*.
   b. In the datacollector_custom.properties file located in the *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/ directory, set the customer.reqestmapper.file property as follows:

      ```
      customer.requestmapper.file=
       DC_home/runtime/DC_specific dir/requestmapper.properties
      ```
   c. In the RequestMapper code, get the location of requestmapper.properties file by doing System.getProperty("customer.requestmapper.file")
4. Restart the WebSphere server.

Java docs and an example follow:

Package com.cyanea.mapper

Table 9. Interface Summary

Interface Summary	
MappedRequest	Interface used for providing the ITCAM for Application Diagnostics system with a Distinguishable Request String (DRS) and a Collapsible Request String (CRS) about a particular Servlet request.
RequestMapper	ITCAM for Application Diagnostics recognizes JSP and Servlet requests on an application server.

Interface mapped request

public interface MappedRequest

Interface used for providing the ITCAM for Application Diagnostics system with a DRS and a CRS about a particular servlet request.

Table 10. Method Summary

Method Summary	
java.lang.String	getCRS ( )
java.lang.String	getDRS( )

Interface Request Mapper

public interface RequestMapper

ITCAM for Application Diagnostics recognizes JSP and servlet requests on an application server. These requests are normally identified throughout the ITCAM for Application Diagnostics system using the URI of the request. In some situations, such as when a Struts design paradigm is used, a particular URI is used to handle different types of business requests.

ITCAM for Application Diagnostics provides this interface as a mechanism for modifying ITCAM for Application Diagnostics 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 "**am.requestmapper**" with the implementing class as the value.

For example:

```
-Dam.requestmapper=com.cyanea.mapper.RequestMapperExample
```

Table 11. Method Summary

Method Summary	
MappedRequest	**mapRequest** (java.lang.String servletClassName, javax.servlet.http.HttpServletRequest request) This stateless method translates 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 translates a servlet classname and a URL into a MappedRequest object. Any RequestMapper class must attempt to execute this method as quickly as possible, because 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 ITCAM for Application Diagnostics system.

Request Mapper Example (1):

```
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):

```
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

Click one of the following letters to look up terms in the online glossary.

A, **B**, **C**, **D**, **E**, **F**, **G**, **H**, **I**, **J**, **K**, **L**, **M**,

N, **O**, **P**, **Q**, **R**, **S**, **T**, **U**, **V**, **W**, **X**, **Y**, **Z**

A

Accumulated CPU
The approximate CPU time utilized under a thread since the current request started.

Acknowledge Mode
The acknowledge mode as one of the following character strings: AUTO_ACKNOWLEDGE CLIENT_ACKNOWLEDGE DUPS_OK_ACKNOWLEDGE NO_ACKNOWLEDGE.

Action
The activity the system takes when a trap is met, currently SNMP, trace, or email.

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 setup or a synchronous receive 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, that is the number of methods called at the same time. 2) The average number of invocations being processed concurrently for all the methods.

Active Requests
The requests that are being serviced.

Active Sessions
1) The number of communication sessions active during the interval. 2) The number of concurrently active sessions. 3) The current number of HTTP sessions actively referenced in the server at the end of the interval.

Active Thread
A thread that is servicing a request.

Active Thread Count
The number of activated threads.

Active Thread Group Count
The number of activated thread groups.

Active Threads
The number of concurrently active threads.

Additional Detail
1) A dynamically generated list based on the selections made by the user when creating the report. 2) A box 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 hostname address on which admin server is listening for connections.

Admin Server Listen Port
The port on which admin server is listening for connections.

Affinity Breaks
The number of HTTP session affinities broken, not counting WAS intentional breaks of session affinity.

Alert Condition
The definition of when to perform the selected action.

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 the 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 passed 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 on data from a user application, as opposed to application server data.

Archive Agent
Accepts the aggregated data from a publish server and performs fast data archiving into the database for reporting purposes.

Attempt to Activate Nonexistent Session
The number of requests for a session that no longer exists (presumably because the session timed out).

Authentication
Verifies the identity of a user who is logging in to the Application Monitor with UNIX user names and passwords via the WebSphere authentication layer.

Authoritative Date/Time Stamp
The authoritative date/time when the data was frozen.

Authoritative Only
The file exists only on the authoritative server.

Authoritative Server
The operator selects this server from an application group to perform a comparison of installed binaries with up to 10 additional servers (the Comparison Servers).

Authoritative Size
The size of the file found on the authoritative server.

Average Active Usage
The running average usage of created connections that are active in the Connector Pool since the pool was last shrunk.

Average CPU Usage
The average percentage that the CPU is busy since the server was started.

Average Create Time
The average method response time for create in milliseconds.

Average Drain Size
The average number of objects discarded in each drain. It applies to entity and stateless beans.

Average Execution Time
The average amount of time in milliseconds all invocations of the servlet have executed since created.

Average Garbage Collection Duration
The average duration of a garbage collection call.

Average Invalidation Time
The average time required to process the invalidation of HTTP sessions.

Average Method Response Time
The average response time, in milliseconds, on all methods of the remote interface for this bean.

Average Method Response Time for Create
The average time in milliseconds a bean create call takes including the time for the load.

Average Method Response Time for Remove
The average time in milliseconds a beanRemove call takes including the time at the database.

Average Pool Size
The average number of objects in the pool. It applies to entity and stateless beans.

Average Remove Time
The average method response time for remove in milliseconds.

Average Sessions Lifetime
The average lifetime of invalidated HTTP sessions.

Average time between Garbage Collection Calls
The average time (in seconds) between two garbage collection calls.

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 waiting time in milliseconds until a connection is granted.

Average Waiting Threads
The average number of threads concurrently waiting for a connection.

Avg. Contention Time
The average time (in milliseconds) spent on each monitor contention. The valid format is a positive integer.

Avg. CPU Time per Nested Request Invocations
Response time is defined as (CPU Clock at the time of the nested request finish minus CPU Clock at time of the nested request start). This field is the average of this value for all of the nested requests in the sample.

Avg. CPU Time per Request/Transaction
Response time is defined as (CPU Clock at the time of request finish minus C PU Clock at time of request start). This field is the average of this value for all of the requests in the sample.

Avg. Heap Size after Garbage Collection
The average dynamic storage for a procedure after inactive data is deleted.

Avg. Number of Methods Called per Request/Transaction
Average (over the sample shown on the Detail tab) of the number of method and nested request invocations occurring between the request start and end.

Avg. Response Time
The average elapsed time between entering a request and receiving a response.

Avg. Response Time per Nested Request Invocations
Response time is defined as (Wall Clock at the time of the nested request finish minus Wall Clock at time of the nested request start). This field is the average of this value for all of the nested requests in the sample.

Avg. Response Time per Request/Transaction
Response time is defined as (Wall Clock at the time of request finish minus Wall Clock at time of request start). This field is the average of this value for all of the requests in the sample.

Avg. Time in Use
The average time of the connection pool that is in use.

Avg. Time to Acquire Lock
The average time (in milliseconds) spent on each monitor locking. The valid format is a positive integer.

of Locks Acquired
The number of monitor locks acquired by the method. The valid format is a positive integer.

of Objects in Heap
The number of instance data in storage.

of Online / Total
The total number of CPU currently enabled.

B

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

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% "very slow response" means that 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 that the response time has exceeded Indicator 2 and the system is unhealthy.

Baseline Response Time
This is the historical response time displayed on the Application Overview page in comparison to the current response time.

Baseline Response Time Sample Duration
The number of days included when the system collects and averages the Baseline Response Time.

Bytes Received
The number of bytes transferred to the server from all attached clients.

Bytes Sent
The number of bytes sent from the server to all attached 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 that loop or abuse resources. Cancel Request terminates the request by throwing a runtime exception, and all necessary cleanup occurs accordingly.

Capacitive Increment
The initial capacity configured for this Connector connection pool.

Change Priority
If you see there is a need to raise or lower the priority of a thread, change it by selecting a priority number. Priority 1 is the lowest and priority 10 is the highest.

Change Thread Status
If you must freeze the execution of a thread, suspend the thread to freeze it while investigating the problem further; then reactivate it when the problem is resolved.

CICS Transaction
Gateway page lists all the CTGs that the selected J2EE server has contacted.

Class A collection of data and methods (operations) on the data which share a common structure and behavior.

Class Acquiring Locks
The name of the class that accessed a monitor. The valid format is an alphanumeric string, maximum 128 characters.

Class Path
The path name 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 comparison date/time when the data was frozen.

Comparison Only
The file exists only on the Comparison Server.

Comparison Servers
The operator selects up to 10 servers to compare installed binaries to on the authoritative server.

Component ID
An ID assigned by the ITCAM system for identification.

Component Trace
A list of event flow of the nested requests. It displays a path of execution for a nested request at the component level.

Composite Requests
The requests that conform to an Enterprise Application Integration architecture. The requests for web-enabled preexisting applications.

Composite Transactions
It allows the user to monitor transactions that use 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 the run time, whether active or pooled objects (the system instantiated but did not destroy). 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
The user-defined criteria that combines with the selected resource to set a trap.

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 configured data collector to 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 summary 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 ConnectionFactory Name for the Connection Factory using this Connector connection pool.

Connection Pool Faults
The number of faults (for example, timeout) in the connection pool.

Connector Pool Name
1) The configured Logical Name for the Connection Factory using this Connector connection pool. 2) The name of the connection pool this SQL statement belongs to. 3) The name of the connection pool for the leaked connection.

Container Thread Pool
The current number of threads in the container.

Context Root
The context root (context path) for the Web application.

Contrast Options
A second data set used for comparative analysis.

Cookies & Attributes
The name of the cookies associated with this session and the contents of the cookies.

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 spent so far by the current request.

Current Total Elapsed Time
The total time that has elapsed since the request began executing.

Custom Request Type
The user defines a custom request in a configuration file that allows instrumentation of a class.

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
An Application Monitor software that runs on the 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, that is, month, date of the month, day of the week, and hour of the day.

Data Interval
The type of data provided; the distance between points on the X axis of the 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) The database connection pool on the selected application server has an indicator that displays the number of connections in use and the total number of connections in the pool.

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
This report provides a breakdown of the Trend report by the criteria selected.

Default Data Collector Configuration
The setup configuration assigned to new data collectors to capture information regarding the applications running inside the application server.

Default Monitoring Level
The currently set default monitoring level for all servers connected to the Application Monitor. This is the case when configuring a server for the first time and bringing up the server under the management of the Application Monitor. The default monitoring level for the non-z platform is L2 (Problem Determination). As for the z/OS platform, the default monitoring level is L1 (Production Mode).

Destination Name
The destination for the consumer.

Destination Type
The name or address of the computer being sent SNMP messages.

Drains from Pool
The number of times the daemon found the pool was idle and attempted to clean it. 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 provides 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 might involve developing a new total view of the business and applications of an enterprise, seeing how existing applications fit into the new model, and then devising ways to efficiently reuse what exists while adding new applications and data.

EAR File
1) Enterprise Archive File. 2) 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 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.

Enterprise Overview
It displays the availability for all the applications available in the assigned server groups.

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.

Errors The number of errors encountered during the Servlets execution.

Exclude (Classname)
A list of classes that are not monitored unless they are part of the Exclude Override (Classname).

Exclude Override (Classname)
A subset of classes in the Exclude (Classname) that are not monitored.

Execute Queue
As request enters a WebLogic Server, it is placed on an execute queue while waiting to be performed. This request is then assigned to a thread to process it.

Execution Time High
The amount of time in milliseconds the single longest invocation of the servlet has executed since created.

Execution Time Low
The amount of time in milliseconds the single shortest invocation of the servlet has executed since created.

Execution Time Total
The amount of time in milliseconds all invocations of the servlet have executed since creation.

Existing Sessions
The number of existing communication sessions of the end of the interval.

External Read Size
The size of session data read from persistent store. Applicable only for (serialized) persistent sessions.

External Read Time
The time (in milliseconds) taken in reading the session data from 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. When using a JMS persistent store, if the user chooses not to serialize the data, the counter is not available.

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 might happen because of database unavailability or broken connection to the database.

Faults The number of faults (for example, timeout) in connection pool.

File Name Match
The file names only matched. They are unlikely to be the same.

File Name/Path/Size Match
The files with matched file name and path, and size, but not timestamp, are likely to be the same.

Finalized Sessions
The number of sessions that were finalized.

First Join Time
The first time the component joined with the Kernel.

Fixed Rate
The average response time per 5 minute increments from between the start

date and end date becomes the baseline against which your current response times on the Application Overview is compared.

Fixed Response Time
The response time entered in this field becomes the response time against which your current response times on the Application Overview is compared.

Force GC Force Garbage Collection
When this option is enabled, JVM completes a garbage collection before taking a heap dump.

Free Memory
1) The free memory in JVM run time. 2) The snapshot of free KB of free memory.

Free Pool Size
The number of free connections in the pool.

Full Match
The file name and path, size, and file system timestamp are matched and likely to be identical to each other.

Full path name / Size Match
Files that are likely to be identical to each other based on matching path name 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 references to it are not longer stored in variables, the fields of any objects, or the elements of an arrays.

Garbage Collection Delay
The amount of time the system must wait before taking the second heap snapshot.

Gets Found
The number of times a retrieve found an object available in the pool. It applies to entity and stateless beans.

Gets from Pool
The number of calls retrieving an object from the pool. It 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 commit for global transactions.

Global Prepare Duration
The average duration of prepare 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 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 displays 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 is unknown until the program is running.

Heap Size
The amount of memory allocated to the JVM.

I

Idle Time
Idle time is time that the request has been idling plus any unaccounted CPU time not captured by the Application Monitor.

Initial Capacity
The initial capacity configured for the 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 path name 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 the Connection Factory using this 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 static elements -- the actual 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 / Region (DSA, EDSA) Memory Usage
The JVM / Region Memory in use.

JVM / Region CPU %
The utilization percentage of JVM / Region CPU.

JVM / Region CPU Delta
The difference between the current JVM/Region CPU and its last refreshed data.

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 since the last refresh 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 percent of JVM CPU usage exceeds the threshold value.

JVM Heap Size
The amount of heap that is available to the JVM code.

JVM ID
The JVMID of the server.

JVM Memory Usage
1) The amount of memory used, in MB, by the JVM of the selected 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.

JVM Thread
The basic unit of program execution in the Java Virtual Machine (JVM). 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.

JVMPI
Java Virtual Machine Profiler Interface. A two-way function call interface between the Java Virtual Machine and an in-process profiler agent.

K

Kernel
A directory service dedicated to the Application Monitor that tracks which components have joined or left the network.

Kernel Code Base
The URL where the Application Monitor component downloads the Kernel binaries.

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)
This monitoring level provides Production level monitoring of advanced request data, including CPU information, and canceling a request without SQL level data. The JVMPI is enabled on the corresponding JVMs making JVMPI function calls possible. 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, and Cancel Request functions are available. Use this level for servers that have been selected for 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 being accessed by the current request.

Last Known Method
The name of the last method being accessed by the current request.

Last Known SQL Statement
The last SQL statement accessed by the current request.

Library Path
The path name where the Library is stored.

Line Numbers (CICS only)
The capability to retrieve the line number from the high-level language source code for each EXEC CICS call made by the program.

Listen Address
The address on which this server is listening for connections.

Listen Port
The port on which this server is listening for connections.

Listening Port
The port that clients of the data collector use to communicate with the data collector.

Live Sessions
The number of concurrently live sessions in the application server cache.

Load Timestamp
The timestamp when the 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 local 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 local 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.

Lock Object Class
The classname of the locked object. The valid form is an alphanumeric string, maximum 128 characters.

Log File Name
The Log File used by the Resource Adapter for the Connector Connection Pool.

Logging Enabled
The Log File used by the Resource Adapter for the Connector Connection Pool.

M

Managed Space
This is a term we use to describe the entire scope of what the product can monitor. Since the product can monitor servers and application servers, along with applications and J2EE components like EJBs, the Managed Space has many dimensions.

Max Capacity
The maximum capacity configured for the Connector Connection Pool.

Max Inactive Interval
The maximum time interval, in seconds, that the servlet container keeps this session open between client accesses.

Max Priority
The highest rank assigned to a thread that determine its precedence in processing a request.

Max. Time to Acquire Lock
The maximum time (in milliseconds) spent on each monitor lock. The valid format is a positive integer.

Maximum Active Sessions
The maximum number of active HTTP sessions during the interval.

Maximum Inactive Interval
The maximum time interval, in seconds, that the servlet container keeps this session open between client accesses.

Maximum Live Sessions
The maximum number of live HTTP sessions during the interval.

Maximum Method Records
The maximum number of method entry/exit records. The records are over written when they reach this value. The default value is 10,000.

Maximum Response Time
The maximum response time measured in milliseconds.

MBean
A custom MBean is a Java object that implements a specific interface and conforms to certain design patterns. These requirements formalize the representation of the resource management interface in the MBean. The management interface of a resource is the set of all necessary information and controls that a management application needs to operate on the resource.

MD5 A unique numeric signature that is different for each file when the contents of the file are different even if the creation date and the file names coincide.

Memory Dump Diagnostic
This feature is only applicable for platforms supported by MDD. Memory Dump Diagnostic analyzes either one single heap dump or analyzes and compares two heap dumps while searching for evidence of a memory leak.

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 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 shows the last 48 hours. Otherwise, your report displays 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 onMessage method. It applies to Message Driven beans.

Message Count
The number of messages delivered to the bean onMessage method. It applies to Message Driven beans.

Message Dispatcher
The Message Dispatcher sends out e-mails of performance reports and trap results from the Performance Analysis & Reporting and the Trap & Alert Management applications.

Messages Threshold Time
The amount of time in the threshold condition since the last reset.

Method
1.) The type of HTTP request with valid values of Get or Post. 2) A

function defined in a class. A class can contain data and methods. Methods are operations that are performed on the data. 3) The number of associated methods.

Method Acquiring Locks
The name of the method that accessed a monitor. The valid format is an alphanumeric string, maximum 128 characters.

Method/Component Records
Number of method or nested request events occurring inside the context of this event.

Method Profiling (MP)
A monitoring level that provides Availability Management, System Resources, basic request-level data, and an option to collect method profiles on demand. This monitoring level least affects the CPU processor usage per transaction and is appropriate for servers that are not malfunctioning.

Method Signature
1) The name of the method including its signature (only the leftmost 512 characters are recorded). 2) Methods might have the same name but accept different arguments. An example of a uniquely "callable" method would be classname+methodname+methodsignature.

Method Trace
A method trace is essentially 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 the entry and exit records including the Method Name, Date, Time, Elapsed Time, and CPU Time. Each indent indicates a different nesting level.

Metric 1) The item that you want to measure: Throughput per Second, Throughput per Minute, Throughput per Hour, Response Time, or CPU Time. 2) The item that you want to measure: Pool Size, Concurrent Waiters, Average Wait Time, Faults, Percentage Pool Usage, Physical Connections, Connections Handles, JVM Free Memory, and JVM Memory Used.

Minimum Active Sessions
The minimum number of active HTTP sessions during the interval.

Minimum Life Sessions
The minimum number of live HTTP sessions during the 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 can select between three levels of monitoring for a server or set of servers: L1 (Production mode), L2 (Problem Determination mode), and L3 (Tracing mode).

MP (Method Profiling)
A monitoring level that provides Availability Management, System Resources, basic request-level data, and an option to collect method profiles on demand. This monitoring level least affects the CPU processor usage per transaction and is appropriate for servers that are not malfunctioning.

MQBACK
An MQ API to back out an MQ transaction.

MQBEGIN
An MQ API to begin an MQ transaction.

MQCLOSE
An MQ API to close a queue.

MQCOMIT
An MQ API to commit a transaction.

MQCONN(X) Average Response Time
An MQ API to make queue manager connection.

MQDISC
An MQ API to disconnect a queue manager connection.

MQGET Average Response Time
The average response time of an MQ API to get a message from a queue.

MQINQ
An MQ API to inquire queue attributes.

MQOPEN
An MQ API to open a queue.

MQPUT(1) Average Response Time
An MQ API to put a message in a queue.

MQSET
An MQ API to set a queue attribute.

N

Name A concatenated string. 1) for non-z/OS platform: the name of the WebSphere Node; the name of the application server; the process ID in decimal of the server; the current monitoring level. 2) for z/OS platform: the name of the Sysplex Node; the name of the Application Instance; the address Space ID in decimal of the server region; the current monitoring level.

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 cannot 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 this Connector Pool since the pool was instantiated.

Number of Active Servers
The current total number of alive servers in this cluster.

Number of Active Transactions
The number of active transactions on the 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
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 is instantiated.

Number of Connections Current
The current number of connections to the WebLogic Server.

Number of Connections Destroyed Total
The total number of Connector connections destroyed in this Connector Pool since the pool is instantiated.

Number of Connections High
The peak number of connections to the WebLogic Server since the last reset.

Number of Connections Matched Total
The total number of times a request for a Connector connection 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 connection in this Connector Pool since the pool is instantiated.

Number of Connections Total
1) The total number of JDBC connections in the JDBCConnectionPoolRuntimeMBean since the pool was instantiated. 2) The total number of connections made to the WebLogic Server since the last reset.

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 servlet/JSP.

Number of Foreign Fragments Dropped
The number of fragments that originated in foreign domains/clusters 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 this server into the cluster.

Number of Free Connections Current
The current total free connections.

Number of Free Connections High
The peak number of free connections in this 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 the 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 JMS Servers Current
The current number of JMS servers that are deployed on this WebLogic Server instance.

Number of JMS Servers High
The peak number of JMS servers that were deployed on this WebLogic Server instance since the server was started.

Number of JMS Servers Total
The number of JMS servers that were deployed on this WebLogic Server instance since the server was started.

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 Acquisitions
The number of locks acquired.

Number of Lock Contentions
The number of monitor connections that have occurred. The valid format is a positive integer.

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 on this server attempted 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 Moved
The number of objects moved.

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 sent again 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
1) The total number of connections made to the WebLogic Server since the last reset. 2) The total number of JDBC connections in the JDBCConnectionPoolRuntimeMBean since the pool was instantiated.

Number of Transactions
The total number of transactions processed. This total includes all committed, rolled back and heuristic transactions completed.

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 we have unlocked a user on the server.

Number of User Lockout Total
The cumulative number of user lockouts done on the 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 waiting for a connection.

Number Waiting for Connections High
The peak number of waiters 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 monitoring level selected overrides the default monitoring level until the next schedule occurs.

P

Per Method Concurrent Requests
The number of concurrent calls to invoke the same method.

Percent CPU Usage
The average percentage busy for the CPU since last query.

Percent Maxed
1) The average percent of time that all the connections are in use. 2) The average percent of time that all the threads are in use.

Percent of Total Number
The number of the 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 percent of the 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 must be increased. See WebSphere documentation.

Percent Used
The average percent of the pool that is in use.

Plan Name
The DB2 plan name used by this 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 the 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 computer 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.

Projected Count
The number of requests in the Detail report, after you divide by the Sampling Rate for each request. For example, if your Sampling Rate is 2%, each request in the Detail report counts as 50 in the Projected Count. Since Sampling Rates are different for each request in the Detail report, this value is computed line by line and then the sum is displayed.

Publish Server
Accepts data from a data collector and aggregates it based on different needs.

R

Recent Activity
A diagnostic tool provides server activity analysis reports regarding memory.

Recent Requests
The requests that are recently completed.

Recycle Total
The total number of Connector connections that have been recycled in this Connector Pool since the pool was instantiated.

Reentrant
The bean 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 the application server.

Registered Servlets
The number of registered servlets on the application server.

Remote Active Sessions
The number of active remote communication sessions attached and active within the 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 the user 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. NAME=Request Object Attributes The attributes bound to the request object.

Request Sampling Rate
The percentage of requests stored in the database for reporting and analysis.

Request Type
The type of request either EJB, JSP, or servlet.

Request URL
The URL associated with this 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 up to now.

Resource
1) The full name of the EJB. 2) The resource selected for trapping, that is, Occurrence, CPU time, Resident time, Wait time, SQL Resident time, HTTP request parameters, or SQL statements. 3) The full name of the servlet or JSP on the application server.

Resource Adapter Link Ref
The Resource Adapter Link Reference for cases where this Connection Factory refers to an existing Resource Adapter deployment.

Response Time (ms)
The response time (in milliseconds) a servlet request is finished.

Return Codes (CICS only)
The value of the response code field in the CICS EXEC interface block (the value of RESP in an EXEC CICS call).

Returns Discarded
The number of times the 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 (RPC), 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
The number you place in this field represents the days over which the average response is calculated for the baseline. The response time on the Application Overview page is compared to this baseline.

Run-time Environment
The specifics regarding the setup 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

SAD Server Activity Display. Tracks transactions and requests and provides detailed thread data for an application server at a specific point in time.

Sample Count
The number of requests in the Detail report.

Sample End Time
The time that the last sample arrived.

Sample Start Time
The time that the system receives the first sample.

Sample Sum
The total number of samples collected for the report period.

Sampling Frequency
The percentage of requests stored in the database for reporting and analysis.

Schedule
A time defined by the user when the system switches to a new monitoring level.

Secondary Distribution Names
The names of the remote servers (for example, 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 path name where the Security Policy file is stored.

Selector
1) The selector associated with this consumer. 2) The selector for the 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 the 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 the cluster.

Server Overview
This page displays comprehensive server information, activity, statistics, and resource data for the selected server.

Server Region Name
The name of the Server Region which belongs to the server instance.

Server Resource Trap
A trap on system resource activity, as opposed to the user application behavior.

Server Scope
The server or set of servers on which the report is being 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 plug-ins written in C and C++ for specific Web servers (ISAPI, NSAPI).

Servlet Name
The name of the Servlet or JSP on the application server.

Servlet Volume
The number of times that servlet requests were sent to the application server.

Servlet/JSP Activity
The amount of servlet/JSP calls made for 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 the session object.

Session Create Time
The time the server created the 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 Object
The session object is used to share information for one user across multiple pages while visiting a website. In other words, a session object is a way of retaining state for a normally stateless HTTP website. 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.

Sessions 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 is made.

Shrink Period Minutes
The Shrink Period (in minutes) of this Connector connection pool.

Shrinking Enabled
The shrinking of the 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 belonging to the same Java class as the heap size.

SMF Data
This feature provides detailed information from SMF records published periodically by WebSphere.

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 the component source.

SQL Call
The SQL operation performed on the Table.

SQL Statement
The SQL statement that is currently 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 being executed when the stack trace was requested, in Last in First Out order. For each method, the printout includes the class name, method name, and (optionally) a line number.

Start Date/Time
An ID assigned to a thread. The ID cannot be modified.

State The state of the 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 the transaction status. 2) The component status.

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.

Subscription Name
The subscription name for the durable subscriber.

Suspended
A user paused the thread and can reactivate it when ready.

System Paging Rate
A paging file is a space on a hard disk used as the virtual memory extension of the real memory (RAM) of a computer. Having a paging file allows the operating system of a computer 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
It displays the summary for all the system resource usage information with a 5 minute refresh rate.

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

T

Table Name
The name of the table affected by the SQL call.

Target Type
Maps to the metric for a trap, for example, DB Pool Size or CPU Time.

Thread
A thread allows multiple streams of execution concurrently and independently in the same program. A thread is a thread of control.

Thread Create
The total number of threads created.

Thread Destroy
The total number of threads destroyed.

Thread Dump
A detailed information of memory allocation of threads in a JVM.

Thread ID
ID assigned to a thread by the Application Monitor 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 request to a free thread. 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 different types include, JSP, EJB, or Servlet.

Thread's Priority
The priority of the thread when processing requests.

Threshold
The threshold monitors the action or behavior of the server activity. The system alerts the user when the server exceeds the threshold.

Throughput (response/min, Last Hour)
The number of transactions / requests being transmitted in a given 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
This report displays the top unique methods that, during the report period, took the most cumulative CPU time and the sum total CPU time. It displays up to 100 records.

Top CPU Intensive Requests Report
This report displays top unique requests that, during the report period, took the most cumulative CPU time and the sum total CPU time. It displays up to 100 records.

Top Method Used Report
This report displays the top unique methods used during the report period and how often each request was used. It displays up to 100 records.

Top Request Used Report
This report displays the top unique requests used during the report period and how often each request was used. It displays up to 100 records.

Top Slowest Methods Report
This report 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
This report displays the top unique requests that took the longest time to complete and the average completion time. It displays up to 100 records.

Top SQL Intensive Methods Report
This report displays the top unique methods that made the highest sum total of SQL calls during the report period. It displays up to 100 records.

Top SQL Intensive Requests Report
This report displays the top unique requests that made the highest sum total of SQL calls during the report period. It displays up to 100 records.

Top SQL Used Report
This report displays the top five SQL call types that were most often called, as well as the number of calls during the report period.

Top Tables Used Report
This report displays the tables that were called most often. It also displays the number of times each table was called during the report period. It displays up to 100 records.

Total Acquisition Time
The amount of time spent acquiring a lock.

Total Contention Time
The total Time (in milliseconds) spent on monitor contention. The valid format is a positive integer.

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 percent of the entire platform 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 run time.

Total Method Calls
1) The number of calls to the remote methods of the bean. 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 send to the 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. This includes both active and inactive sessions.

Total SQL Used
The top five SQL call types that were most often called and the number of calls.

Total Thread Count
The total number of active requests being serviced by the selected application server.

Total Time to Acquire Locks
The total time (in milliseconds) spent on monitor lock. The valid format is a positive integer.

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 are not successfully completed.

Transaction Policy
The bean method 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 this connector connection pool.

Transaction Volume
The number of times that transactions were executed on the application server.

Trap A set of conditions, thresholds, and criterion set by the user, when the system meets the settings, an action occurs.

Trap Action History
When a trap meets any condition and action, a record is replaced in this page.

Trap Condition
The user-defined criteria that combine with selected resource to set a trap.

Trap Type
A way of categorizing two different types of metrics for the application and server resource trap.

Trend Report
This report 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 box, and then click the data points.

U

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 JVM run time.

User Name
The database user name that is used for creating the connection.

UUID Universally Unique Identifier. An identifier for each symbol in an activity diagram.

V

Volume Delta
1) The number of completed requests since the page 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 Session Browser
This feature provides session information for one server or multiple servers and allows you to find a specific session across one or more servers.

Web Server Overview
It displays the information for the performance of Web servers.

Web Services Request Type
This is the Web Services transactions in and out of the application server. Support includes JAX-RPC over SOAP/HTTP and SOAP/JMS.

WebLogic Version
The version of the server.

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.

Chapter 4. ITCAM Agent for WebSphere Applications

IBM Tivoli Composite Application Manager Agent for WebSphere Applications provides a Systems Management solution for the WebSphere application server Versions 7, 8 and 8.5. Using the agent, you can monitor multiple WebSphere application servers running on the same physical node. Each application server must have been configured with its own ITCAM Data Collector for WebSphere.

IBM Tivoli Composite Application Manager Agent for WebSphere Applications is a component of ITCAM for Application Diagnostics Version 7.1 and later. It is also a component of ITCAM for Applications Version 7.2. If you are using ITCAM for Applications, the managing server (deep dive) functionality is not available; ignore all references to this functionality in this document.

The Tivoli Enterprise Monitoring agent collects four types of data through the data collector embedded in the WebSphere Application server process:

- Data for application server requests from the ITCAM Data Collector for WebSphere
- Resource data from WebSphere Performance Monitoring Infrastructure (PMI)
- Data from WebSphere log files
- Process data from the operating system

Initiating data collection and the reporting of data

Because of high processor usage, some data items are not automatically collected and reported. The collection of some data and statistics depends upon the setting of instrumentation levels for certain attributes. If the instrumentation levels are not set appropriately, certain information is not collected and displayed in the workspaces. Similarly, those attributes that collect request and application trace data require you to complete several configuration steps. If you need to collect these data, use one of these methods to reconfigure data collection:

- Complete configuration steps (as explained in the ITCAM Agent for WebSphere Applications installation and customization guide).
- Issue Take Action commands to take specific action against your WebSphere application server or the monitoring product using the Tivoli Enterprise Portal.
- Use Manage Tivoli Enterprise Services (as explained in the various IBM Tivoli Monitoring installation manuals and the ITCAM Agent for WebSphere Applications installation and customization guide).

Automatic baselining

To display application health status, ITCAM monitors request response times (averaged over a sampling interval, by default 60 seconds) for every application. Every top-level request available in an application is monitored separately.

For every request, two *thresholds* are set, known as *fair* and *bad*. Response time below both thresholds is considered *good*. When at least one average request response time for an application rises over the fair threshold, a health warning (indicated with an exclamation mark symbol) for this application is reported. In the same way, when at least one average request response time rises over the bad threshold, an application health alarm (indicated with an x symbol) is reported.

© Copyright IBM Corp. 2006, 2015

ITCAM also monitors the "nested" requests (for example, database calls) within every top-level request. In the event of a warning or alarm, it checks which of the nested requests is taking more than its usual share of time. Depending on the type of such nested requests, ITCAM shows whether the client, application, or backend tier is the likely cause of the warning/alarm. Servlet and Portal request types are assigned to the client tier; EJB and User (Custom) request types, to the application tier; all other request types (JNDI, JDBC, JCA, JMS) to the backend tier.

Important: Baselining is currently not supported for SCA requests.

When ITCAM starts to monitor a new application, it automatically starts a *baselining process*. This process normally runs for seven days and provides updated information every hour from the beginning. During the process, ITCAM collects statistical data for all requests in this application. Once the data is collected, ITCAM sets the thresholds automatically; it also records the typical share of response time for each nested request type.

In most cases, this automatic setting is adequate. During the baselining process, the baselines get updated periodically, and the alarms/warnings start to correspond to real problems. There is no need to adjust baselining settings when things are working normally. (The automatic thresholds usually become usable earlier, after the application has been observed through its typical load patterns). If you need to acquire thresholds, based on whatever data is available, before the hourly automatic update, you can manually update baselining.

However, in some situations the threshold levels can become inadequate. This results in either too many false alarms/warnings, or in real problems going undetected. Such situations can be broadly split into two categories:

- If some time has passed since the baselining process for an application, its response times might have changed because of configuration alteration, database growth, changing load patterns, and so on. In this case, you might need to run the baselining process again. It is good practice to do it after any configuration or infrastructure change.
- If the thresholds are incorrect immediately after the baselining process has been completed, you might have to adjust the auto threshold settings.

As a last resort, you can also override the thresholds with fixed values. However, do not do this unless you know a lot about the monitored application, or unless instructed by IBM Level 3 Support.

If you need to have the thresholds set before they are updated automatically for the first time, you can trigger a baseline update. This immediately sets the thresholds based on the request data collected so far.

Additional information

For additional usage information about this agent, see:

- Workspaces
- Attributes
- Situations
- Take Action commands

Workspaces for ITCAM Agent for WebSphere Applications

As part of the IBM Tivoli Composite Application Manager Agent for WebSphere Applications product integration with the Tivoli Enterprise Portal, workspaces offer views of monitoring data that provide detailed current data about WebSphere application servers running on Linux, UNIX, and Windows and z/OS platforms on your site. In addition to reports and graphs, a workspace can contain other views (that is, windows), such as a Notepad editor session, a browser session, a telnet session, an event console, or a Take Action view from which you can issue commands.

Several views of high-level information

Several workspaces provide high-level information to help you meet the monitoring and administrative needs of your site. These workspaces report on status and availability for both the WebSphere administrative server and its application server instances. They allow you to easily monitor the availability of your enterprise, the WebSphere application server, and application server instances.

Primary and secondary workspaces

The workspaces listed in the Navigator are directly accessible and are thus termed *primary workspaces*. Some of these also contain *secondary workspaces*, which are not accessible directly from the Navigator. Instead, you must select and display the primary workspace. Then, use either a menu option or a special link icon in the primary workspace views to reach the secondary workspaces (sometimes called subsidiary workspaces).

Workspaces with historical data links

Several workspaces provide secondary workspaces that display historical data. You can specify a time span over which to collect historical data, which accumulates and summarizes the data in the primary workspaces that generate them. (The default setting is 15 minutes; you can modify this setting to suit your needs.) The descriptions of the historical workspaces follow the descriptions of the primary workspaces that generate them in the workspace helps.

Available Tivoli Enterprise Portal workspaces

For an overview of the organization of the available workspaces, see Organization of the predefined workspaces.

Organization of the predefined workspaces

The IBM Tivoli Composite Application Manager Agent for WebSphere Applications workspaces for the Tivoli Enterprise Portal define data displays that appear in the Navigator Physical view. In addition to the workspaces that the Navigator lists, you can reach their subsidiary (that is, secondary) workspaces from the primary workspaces (those listed in the Navigator).

Accessing the subsidiary workspaces

You can access the subsidiary workspaces of a primary workspace by using one or more of the following methods:

From the Navigator:

1. Select the primary workspace.
2. Right-click the name of the selected workspace in the Navigator.
3. Select **Workspaces** from the menu.
4. Select the desired subsidiary workspace.

From the View menu:

1. Select the primary workspace.
2. From the menu bar, select **View > Workspaces**.
3. Select the desired subsidiary workspace.

From a report:

1. Select the primary workspace.
2. If the workspace report contains a link icon, click the icon to navigate to the default subsidiary workspace pertaining to the selected row or right-click the icon and select a subsidiary workspace from the menu.



From a chart view:

The data displayed in some bar charts and plot charts is linked to subsidiary workspaces. To search for a link, right-click a bar or data point in the chart. If **Link to** displays in the menu, you can select a subsidiary workspace pertaining to the data in the chart.

Workspace organization

The hierarchy levels shown in the Navigator depend on how your enterprise customizes the Tivoli Enterprise Portal. However, ITCAM Agent for WebSphere Applications does provide a set of predefined workspaces, which do not require customization. The following list shows the order and hierarchy of the predefined workspaces provided by the ITCAM Agent for WebSphere Applications. It is a representation of how the predefined workspaces are organized in the Navigator. For more detailed information about a workspace, click its name in the table.

Important: Some workspaces are only available when running a monitoring agent in a z/OS environment. For more information, see "Region workspaces in a z/OS environment" on page 346.

operating system [for example, Windows]

- *system* [that is, node name]
  - WebSphere Agent
    - "WebSphere agent summary workspace" on page 259
    - "WebSphere agent summary status workspace" on page 259
    - "Application Server Summary and Business Process Manager Summary workspaces" on page 260
    - "Configuration workspaces" on page 267

- "WebSphere application server workspace" on page 339
- "Resources and Applications workspaces" on page 263
- "High Availability Manager workspace" on page 302
- "DCS Stacks workspace" on page 294
- "Configuration workspaces" on page 267

-

- Application Health
  - Application Registry
  - Application Trend at L1
  - Application Trend at L2/L3
  - OS Stack
  - JVM Stack Trend
  - "Web Tier Analysis workspace" on page 323
  - "Backend Tier Analysis workspace" on page 322
  - "Request Baseline workspace" on page 319
  - "Application Configuration workspace" on page 321
  - "EJB Tier Analysis workspace" on page 320
  - "Application Health History workspace" on page 323
  - "Selected Request - Service Components workspace" on page 327
- Request Analysis
  - Selected Request - Datasources
  - Selected Request JMS Queues
  - Selected Request Resource Adapters
  - Selected Request - History
  - "Selected Request - Service Components workspace" on page 327
  - "Service Components workspace" on page 329
- Garbage Collection Analysis
  - Allocation Failures
  - Garbage Collections - Selected Allocation Failure
- Log Analysis
- Pool Analysis
- Datasources
  - "Data sources workspace" on page 290
  - "Selected Datasources - Datasource Trend workspace" on page 324
- JMS Summary
- Web Applications
  - Sessions
  - Servlets / JSPs - Selected Enterprise Application
- EJB Containers
  - Container Object Pools
  - Container Transactions
  - Enterprise Java Beans
- DB Connection Pools
  - Selected DB Connection Pool - History
- J2C Connection Pools

    - Thread Pools
      - "Thread Pool Trend workspace" on page 334
      - Alarm Manager
    - Cache Analysis
      - "Thread Pool Trend workspace" on page 334
      -
    - Workload Management
    - Scheduler
    - Web Services
      - Selected Web Services - History
    - Messaging Engines
      - Client Communications
      - Messaging Engine Communications
      - WMQ Client Link Communications
      - WMQ Link Communications
      - Destinations
        - Durable Subscriptions
- WebSphere Portal Server
    - "Application Server Summary and Business Process Manager Summary workspaces" on page 260
    - "Configuration workspaces" on page 267
      - "WebSphere application server workspace" on page 339
      - "Resources and Applications workspaces" on page 263
      - "High Availability Manager workspace" on page 302
      - "DCS Stacks workspace" on page 294
      - "Configuration workspaces" on page 267
    - Portal Summary
    -
      - Portlet Summary
        - Selected Portlet - History
      - Portal Pages Summary
        - Selected Portal Page - History
    - Request Analysis
      - Selected Request - Datasources
      - Selected Request JMS Queues
      - Selected Request Resource Adapters
      - Selected Request - History
    - Garbage Collection Analysis
      - Allocation Failures
      - Garbage Collections - Selected Allocation Failure
      -
    - Log Analysis
    - Pool Analysis
    - Datasources
      - "Data sources workspace" on page 290

        - "Selected Datasources - Datasource Trend workspace" on page 324
    - JMS Summary
    - Web Applications
        - Sessions
        - Servlets / JSPs - Selected Enterprise Application
    - EJB Containers
        - Container Object Pools
        - Container Transactions
        - Enterprise Java Beans
    - DB Connection Pools
        - Selected DB Connection Pool - History
    - J2C Connection Pools
    - Thread Pools
        - Alarm Manager
    - Cache Analysis
    - Workload Management
    - Scheduler
    - Web Services
        - Selected Web Services - History
- WebSphere ESB Server
    - Application Health Summary
        - Selected Application - Application Tier Analysis
        - Selected Application - Configuration
        - Selected Application - Backend Tier Analysis
        - Selected Application - Health History
        - Selected Application - Client Tier Analysis
        - Application Registry
            - Selected Request - Baseline
    - High Availability Manager
    - DCS Stacks
    - Request Analysis
        - Selected Request - Datasources
        - Selected Request JMS Queues
        - Selected Request Resource Adapters
        - Selected Request - History
        - "Selected Request - Service Components workspace" on page 327
        - "Service Components workspace" on page 329
    - Garbage Collection Analysis
        - Allocation Failures
        - Garbage Collections - Selected Allocation Failure
    - Log Analysis
    - Pool Analysis
    - Datasources
        - Selected Datasource - History
    - JMS Summary

- Web Applications
  - Sessions
  - Servlets / JSPs - Selected Enterprise Application
- EJB Containers
  - Container Object Pools
  - Container Transactions
  - Enterprise Java Beans
- DB Connection Pools
  - Selected DB Connection Pool - History
- J2C Connection Pools
- Thread Pools
  - Alarm Manager
- Cache Analysis
- Workload Management
- Scheduler
- Web Services
  - Selected Web Services - History
- Messaging Engines
  - Client Communications
  - Messaging Engine Communications
  - WMQ Client Link Communications
  - WMQ Link Communications
  - Destinations
    - Durable Subscriptions
- Service Components
  - Service Component Elements

- WebSphere Process Server
  - Application Health Summary
    - Selected Application - Application Tier Analysis
    - Selected Application - Configuration
    - Selected Application - Backend Tier Analysis
    - Selected Application - Health History
    - Selected Application - Client Tier Analysis
    - Application Registry
      - Selected Request - Baseline
  - High Availability Manager
  - DCS Stacks
  - Request Analysis
    - Selected Request - Datasources
    - Selected Request JMS Queues
    - Selected Request Resource Adapters
    - Selected Request - History
    - “Selected Request - Service Components workspace” on page 327
    - “Service Components workspace” on page 329
  - Garbage Collection Analysis

        - Allocation Failures
        - Garbage Collections - Selected Allocation Failure
      - Log Analysis
      - Pool Analysis
      - Datasources
        - Selected Datasource - History
      - JMS Summary
      - Web Applications
        - Sessions
        - Servlets / JSPs - Selected Enterprise Application
      - EJB Containers
        - Container Object Pools
        - Container Transactions
        - Enterprise Java Beans
      - DB Connection Pools
        - Selected DB Connection Pool - History
      - J2C Connection Pools
      - Thread Pools
        - Alarm Manager
      - Cache Analysis
      - Workload Management
      - Scheduler
      - Web Services
        - Selected Web Services - History
      - Messaging Engines
        - Client Communications
        - Messaging Engine Communications
        - WMQ Client Link Communications
        - WMQ Link Communications
        - Destinations
          - Durable Subscriptions
      - Service Components
        - Service Component Elements
  - Lotus® Workplace Server
    - Application Health Summary
      - Selected Application - Application Tier Analysis
      - Selected Application - Configuration
      - Selected Application - Backend Tier Analysis
      - Selected Application - Health History
      - Selected Application - Client Tier Analysis
      - Application Registry
        - Selected Request - Baseline
    - High Availability Manager
    - DCS Stacks
    - Workplace Mail

  - IMAP/POP
  - Messages Queues
- Portal Summary
  - Portlet Summary
    - Selected Portlet - History
  - Portal Pages Summary
    - Selected Portal Page - History
- Request Analysis
  - Selected Request - Datasources
  - Selected Request JMS Queues
  - Selected Request Resource Adapters
  - Selected Request - History
- Garbage Collection Analysis
  - Allocation Failures
  - Garbage Collections - Selected Allocation Failure
- Log Analysis
- Pool Analysis
- Datasources
  - Selected Datasource - History
- JMS Summary
- Web Applications
  - Sessions
  - Servlets / JSPs - Selected Enterprise Application
- EJB Containers
  - Container Object Pools
  - Container Transactions
  - Enterprise Java Beans
- DB Connection Pools
  - Selected DB Connection Pool - History
- J2C Connection Pools
- Thread Pools
  - Alarm Manager
- Cache Analysis
- Workload Management
- Scheduler
- Web Services
  - Selected Web Services - History

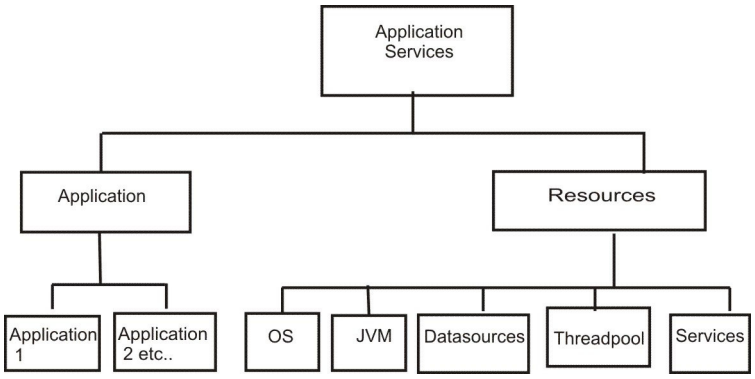
For additional information, see: Attribute groups used by the predefined workspaces

Summary workspaces

You can use summary workspaces to quickly see the status of WebSphere application servers and applications in your enterprise.

About Summary Workspaces

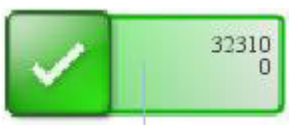
Summary workspaces provide a way to quickly monitor the status of application servers and applications. ITCAM for Application Diagnostics provides predefined situations that you can use to monitor WebSphere application servers in your enterprise. Summary workspaces enable you to quickly determine the status of these situations. User-defined and predefined situations are mapped to various colored icons in the summary workspaces. The icon color indicates status which enables you to quickly determine the overall health of applications servers and applications. The following organization chart shows the structure of the icons in the summary workspaces:



Summary Workspace Icons

In summary workspaces, each icon displays as one of the following statuses: Critical, Warning, Normal, or Unknown. The status is calculated based on the status of the underlying situations being monitored. Each icon also displays the metrics for the first two situations shown on the flyover. The following table shows the possible status of icons:

Table 12. Status Icons

Status Icon	Status
	Critical
	Warning
	Normal
	Unknown or Application Stopped

Summary Workspace Flyovers

The icons indicate the status of the WebSphere application servers and applications in your enterprise. To access more detailed information from the summary workspaces, point to the icon and a flyover is displayed. The flyover provides relevant metrics pertaining to the icon. It also shows the top 10 situations that are linked with the icon. You can go directly to the situation event result workspace by clicking the situation link icon in the flyover.

Drill Down on Summary Workspaces

You can drill down on the icons to see more information. When you double-click an icon, further workspace views showing more detailed monitoring data are displayed.

See also

"Situations for ITCAM Agent for WebSphere Applications" on page 488

Summary Workspace Views

You can use summary workspaces to quickly see the status of WebSphere application servers and applications in your enterprise. Each workspace can contain one or more views.

There are five summary workspaces. The following table describes the summary workspaces:

Table 13. Summary Workspace Table

Workspace Name	Level in TEP	Views Available
WebSphere Agent Summary	Agent level	• Application Servers Status Table • Application Servers Summary
WebSphere Agent Summary Status	Agent level	• Application Servers Status • Application Servers Summary
application server Summary	application server Level	• application server - Resources • application server - Applications
Resources	application server Level	• application server - Resources • Situation Event Console
Applications	application server Level	• application server - Applications • Situation Event Console

WebSphere agent summary workspace

The WebSphere Agent Summary Workspace provides summary monitoring information for WebSphere application servers in your enterprise. It contains two views, the Application Servers Status Table view and the Application Servers Summary view.

Application Servers Status Table view

Two icons are displayed in this view: the Application and the Resources icons.

The flyover for the Applications icon displays the following metrics:

- application server Name
- Average Response Time (ms)
- Request Rate (Req/Sec)
- Error Rate (Errors/sec)
- Number of problem Situations
- List of top 10 situations

The flyover for the Resources icon displays the following metrics:

- application server Name
- JVM CPU%
- GC Active Time (ms)
- Number of problem Situations
- List of top 10 situations

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WebSphere agent summary status workspace

The WebSphere Agent Summary Status workspace is an alternative summary workspace available at the WebSphere agent level. The WebSphere Agent Summary Status workspace provides summary monitoring information for WebSphere Application servers.

To switch to the WebSphere Agent Summary Status workspace, right-click the WebSphere Agent in the Tivoli Enterprise Portal navigator and select **Workspace** and then select **WebSphere Agent Summary Status**.

The WebSphere Agent Summary Status workspace contains two views, the Application Servers Status view and Applications Servers Summary view.

Application Servers Status View

In this view, the Server icon indicates the status of both applications and resources. The flyover for the Server icon displays the following metrics:

- Average Response Time (ms)
- Request Rate (Req/Sec)
- Error Rate (Errors/sec)
- JVM CPU%
- GC Active Time (ms)

- WAS Node Name
- WAS Cell Name
- WAS Cluster Name
- Number of Problem Situations
- List of top 10 situations

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Server Summary and Business Process Manager Summary workspaces

The Application Server Summary workspace provides summary monitoring information for a WebSphere Application Server. The Business Process Manager Summary workspace provides summary monitoring information for a Business Process Manager Server. These workspaces are identical and contain two views: the Application Server - Resources view and the Applications Server - Applications view.

Application server - Resources View

The Application server - Resources view contains the following icons:

- OS
- JVM
- Datasource
- Threadpool
- Services

If ITCAM for SOA is installed on the monitored Business Process Manager server, right-click any resource and select **Operation Flow for Application Server** to view the Operation Flow for Application Server workspace, provided by ITCAM for SOA. In this workspace, you can see the topology for all components running on the server.

OS icon

The OS icon provides summary metrics for machine CPU and paging metrics. The flyover for the OS icon displays the following metrics:

- System CPU (ms)
- System Paging Rate (Kbytes/sec)
- Number of problem situations
- List of top 10 situations

When you double-click the OS icon, the following subsidiary views display in the OS Stack workspace:

- Current OS stack summary
- CPU used
- System Paging

For further information about the OS Stack workspace, see OS Stack

JVM icon

The JVM icon flyover provides summary metrics for: CPU for the JVM process, garbage collection, and heap metrics. The flyover information for the JVM icon displays the following metrics:

- JVM CPU%
- GC Active Time (ms)
- List of top 10 situations

When you double-click the JVM icon, the following subsidiary views display in the JVM Stack Trend workspace:

- JVM CPU Trend
- Percent GC time used
- Heap Usage trend

For further information about the JVM Stack Trend workspace, see JVM Stack Trend

Datasources icon

The Datasources icon flyover provides summary metrics for JDBC, JMS, JCA, and JTA, it also indicates the number of problem situations. When you double-click the Datasource icon, the following subsidiary views display in the Datasources workspace:

- Worst Datasource Query Times
- Worst Datasource Update Times
- Datasources - Current Interval

Threadpool icon

The Threadpool icon flyover provides summary metrics for threadpool information including the number of problem situations. When you double-click the Threadpool icon, the following subsidiary views are displayed in the Threadpools workspace:

- Highest Average Pool Sizes
- Average thread pool usage
- Threadpools table

Services icon

The Services icon flyover provides summary metrics for web services, Workload Management, and System Integration Bus. When you double-click the Services icon:

- for Portal, Services icon takes you to web Services which has the following views:
  - Worst Response Times
  - Most Popular
  - Web Services
  - Web Service Gateway
- for Lotus, the Services icon takes you to Workload Management which has these views:
  - WLM Server Incoming Requests

– WLM Client Outgoing Requests
– Workload Management Server
– Workload Management Client

Application server - Applications View

In this view, there is an icon per application. The flyover for an Application icon displays the following metrics:

- Average Response Time (ms)
- Request Rate (Req/Sec)
- Error Rate (Errors/sec)
- Monitoring Level
- Number of problem Situations
- List of top 10 situations

If ITCAM for SOA version 7.2 is installed on the monitored Business Process Manager server, right-click any application and select **Selected Application - Group Summary** to view the Group Summary workspace, provided by ITCAM for SOA. In this workspace, you can see the status of all process groups that include any components or operations that are a part of the selected application.

When you double click an Application icon, the following subsidiary views are displayed in either the Application Trend at L1 or Application Trend at L2/L3 workspace:

- Response Time Trend
- Error Rate Trend
- Request Rate Trend
- Selected Application Summary: Application Name

For further information about the Application Trend workspace, see "Selected Application - Application Trend at L1 workspace" on page 282, and "Selected Application - Application Trend at L2/L3 workspace" on page 283

Application Server Summary and Business Process Manager Summary workspaces

To access the workspaces, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, click a **WebSphere Application Server** or **Business Process Manager** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Resources and Applications workspaces

The Resources and Applications workspaces provide monitoring data for your application server.

Resources Workspace

To access the Resources workspace, right-click the application server node in the Tivoli Enterprise Portal navigator and select **Workspace** and then select **Resources**. The Resources workspace contains the following views:

- Application Servers Resources
- Situation Event Console

For information about the Application Servers Resources view, see "Application Server Summary and Business Process Manager Summary workspaces" on page 260. The Situation Event Console displays additional detail for all open situations. For details on how to perform filtering on open situations, see Tivoli Monitoring help.

If ITCAM for SOA is installed on the monitored Business Process Manager server, right-click any resource and select **Operation Flow for Application Server** to view the Operation Flow for Application Server workspace, provided by ITCAM for SOA. In this workspace, you can see the topology for all components running on the server.

Applications Workspaces

To access the Applications workspace, right-click the application server node in the Tivoli Enterprise Portal navigator and select **Workspace** and then select **Applications**. The Applications workspace contains the following views:

- Application Servers Applications
- Situation Event Console

For information about the Application Servers Applications view, see "Application Server Summary and Business Process Manager Summary workspaces" on page 260. The Situation Event Console displays additional detail for all open situations. For details on how to perform filtering on open situations, see Tivoli Monitoring help.

If ITCAM for SOA is installed on the monitored Business Process Manager server, right-click any application and select **Selected Application - Group Summary** to view the Group Summary workspace, provided by ITCAM for SOA. In this workspace, you can see the status of all process groups that include any components or operations that are a part of the selected application.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Situation Mapping and Summary Workspaces

User-defined situations are mapped to icons in summary workspaces. When you create a situation, if the situation then triggers an alert, detail of the situation is displayed in one of the summary workspaces icon flyovers.

When you create a situation, the attribute group you base the situation on determines which summary workspace icon flyover the situation displays in. The following table shows which attribute groups map to which icons and predefined situations :

Table 14. Mapping of attribute groups to predefined situations

Tivoli Enterprise Monitoring Agent Attribute Group Name	**Icons**	**Predefined Situations**
Applications Monitoring Configuration	N/A	WASAppDiscovered
Requests Monitoring Configuration	N/A	
Baseline	N/A	
Applications Health Status	Applications	WASApplicationBad, WASApplicationFair, WASApplicationGood
Applications Server Status	JVM	
Log Analysis	JVM	WASError
KYN Command	N/A	
WebSphere Agent Events	N/A	
DC Messages WebSphere	JVM	
Dynamic Cache	Services	
Dynamic Cache Templates	Services	
Workload Management Client	Services	
Workload Management Server	Services	
DB Connection Pools	Datasources	WASDBConnectionPoolThrdTimeout WASDBConnectionPoolUsageMaxed WASDBConPAverageUsageTimeHigh WASDBConPAvgWaitTimeHigh
Container Object Pools	Threadpools	
Enterprise Java beans	Applications	WASEJBCreateTimeHigh WASEJBMethodResponseTimeHigh WASEJBRemoveTimeHigh
Web Applications	Applications	WASWebApplicationError
Web Applications - Sessions	Applications	WASSrvlSessAvgActiveSessionHigh WASSrvlSessExtReadTimeHigh WASSrvlSessExtWriteTimeHigh
Applications Server	JVM, OS	WASHighCPUPercentUsed
EJB Containers	Applications	
Servlets JSPs	Applications	WASServletsJSPError
Servlet Sessions	Applications	
Thread Pools	Threadpools	WASThreadPoolPercentMaxed WASThreadFreeLow

Table 14. Mapping of attribute groups to predefined situations (continued)

Tivoli Enterprise Monitoring Agent Attribute Group Name	Icons	Predefined Situations
Container Transactions	Datasources	WASContainerTransactionRollback WASCTGlbTransDurationHigh WASCTLclTransDurationHigh
J2C Connection Pools	Datasources	WASJ2CConnectionPoolUsageMaxed WASJ2CCPAverageUsageTimeHigh WASJ2CCPAvgWaitTimeHigh
DCS Stack		
High Availability Manager		
Web Services Gateway		
Web Services		
Alarm Manager		
Scheduler	Services	
Client Communications	Services	
Durable Subscriptions	Services	
Messaging Engine Communications	Services	
Messaging Engines	Services	
Queue	Services	
Service Component Elements	Services	
Service Components	Services	
Topic Spaces	Services	
WMQ Client Link Communications	Services	
WMQ Link Communications	Services	
Workplace Mail Service	Datasources	
Workplace Mail Queues	Datasources	
Workplace Mail IMAP/POP	Services	
Portal Summary	Services	
Portal Page Summary	Services	WASPortalPageResponseTime
Portlet Summary	Services	WASPortletResponseTime
Datasources	Services	WASDataSrcConWaitTimeHigh
Request Times and Rates	Applications	WASHighResponseTime
Request Analysis	Applications	WASReqSQLExecuteTimePercentHigh WASReqSQLQueryTimePercentHigh WASReqSQLUpdateTimePercentHigh
JMS Summary	Datasources	
Selected Request	Applications	

Table 14. Mapping of attribute groups to predefined situations (continued)

Tivoli Enterprise Monitoring Agent Attribute Group Name	Icons	Predefined Situations
Garbage Collection Analysis	JVM	WASHighGCTimePercent WASAvgHeapSizeAfterGCHigh
Allocation Failure	JVM	WASOutofHeapSpace
Garbage Collection Cycle	JVM	
WebSphere Agent		WASNotConnected
WebSphere App Server		WASHighCPUPercentUsed WASHighResponseTime

Where NA is indicated for the icon, it means that situations created based on these attribute groups are not reported in the summary workspaces. This is because these tables are strictly related to Tivoli Enterprise Monitoring Agent configuration parameters that do not reflect the application or application server health.

For more information, see:

WebSphere Agent situations
WebSphere Agent attributes

Summary Workspaces error messages

Four possible error messages can be displayed in the summary workspace status bar.

The following table lists and explains the error messages:

Table 15. Summary Workspaces error messages

Error Message	Explanation
Internal Communication Error	This message indicates a communication problem between the summary workspaces front end and the summary workspaces back-end code (called evaluator) running inside the embedded WebSphere server on the Tivoli Enterprise Portal Server server. This error displays for two reasons: 1. If this message is displayed for all the summary workspaces, there is an installation error. Summary workspaces require code to run inside Tivoli Monitoring eWAS. Ensure that Tivoli Monitoring 6.2, including eWAS, is installed. Install ITCAM Agent for WebSphere Applications support files. After installation, reconfigure the Tivoli Enterprise Portal Server. If you do not reconfigure the Tivoli Enterprise Portal Server, the Summary Workspace status bar displays "Internal Communication Error". 2. If this error message is displayed for only some of the summary workspaces, turn up tracing and send to IBM support for review.
Invalid Data	This message indicates that data being sent from the summary workspaces back-end code (called evaluator) running inside the embedded WebSphere server on the Tivoli Enterprise Portal Server server is malformed. Turn on the tracing, collect the logs, and send to IBM support for further analysis.

Table 15. Summary Workspaces error messages (continued)

Error Message	Explanation
No Applications Configured	This message indicates that no applications are configured on the WebSphere server. This message also displays if the WebSphere server is not connected to the Tivoli Enterprise Monitoring Agent. The message no longer displays when the WebSphere server is back online. No user action is needed.
No Applications Servers Connected	This message indicates that no WebSphere servers are connected to the Tivoli Enterprise Monitoring Agent. The message is no longer displayed when the WebSphere server is back online. No user action is needed.

Configuration workspaces

Monitoring Agent configuration and tuning is facilitated in the Tivoli Enterprise Portal through *configuration workspaces*. There are two configuration workspaces: one for the Monitoring Agent level and one for the server level:

- WebSphere Agent Configuration workspace
- application server Configuration workspace

Both configuration workspaces have the same tabs and configuration settings.

The WebSphere Agent Configuration workspace settings are initial settings applied to all data collectors managed by the Monitoring Agent. For example, when a data collector connects to the Monitoring Agent for the first time or if the data collector configuration was deleted. In both these examples, the configuration settings specified in the WebSphere Agent Configuration workspace are applied.

The application server Configuration workspace contains individual server settings and the settings in this workspace override the settings in the WebSphere Agent Configuration workspace.

Settings for the WebSphere Agent Configuration and application server Configuration workspaces are saved in the following files:

Table 16. Workspace configuration files

Workspace	Configuration file
WebSphere Agent Configuration workspace in windows:	*CANDLE_HOME*\TMAITM6*hostname_productcode*.xml
application server Configuration workspace in windows:	*CANDLE_HOME*\TMAITM6\ *hostname_productcode_servervendor.servernode.server name*.xml
WebSphere Agent Configuration workspace in other platforms:	*CANDLE_HOME*/config/*hostname_productcode*.xml

Table 16. Workspace configuration files (continued)

Workspace	Configuration file
application server Configuration workspace in other platforms:	*CANDLE_HOME*/config/ *hostname_productcode_servervendor.servernode.server name*.xml

The WebSphere Agent Configuration workspace configuration settings are initial settings which are applied to all data collectors managed by the selected WebSphere agent. The application server Configuration workspace configuration settings override agent level configuration settings.

Both configuration workspaces have the following two views:

- **Application Diagnostics Configuration view** - this view has the following four tabs
  - Collection (Basic)
  - Collection (Advanced)
  - Application Dashboard (Basic)
  - Application Dashboard (Auto Threshold)
- **Application Servers view** - this view has one tab which lists all application servers monitored by the selected WebSphere agent, for each application server, there is a link to application server Configuration workspace.

Application Diagnostics Configuration - Basic Tab

The Basic tab contains the following fields:

- **Request Data Monitoring** - in this field, specify the monitoring level for requesting data from connected data collectors. The following values can be entered into this field:
  - **Disable** - request data is not monitored and is not displayed
  - **Level 1** - only monitors edge request data, for example, servlets, JSPs, EJBs, SCA requests, and web service requests
  - **Level 2** - monitors nested request data in addition to edge request data, for example, JNDI, JMS, JDBC, and JCA requests

  Important: This monitoring level is applied by default to newly discovered applications. However, you can set the monitoring level for any operation separately, using the "Set_Application_Monitoring: Set monitoring" on page 504 Take Action command.
- **Request Data Monitoring Method** - in this field, specify the monitoring method used by the Monitoring Agent to govern when it uploads request and garbage collection data from connected data collectors. The following values can be entered into this field:
  - **On Demand** - when the monitoring method is set to On Demand, data is uploaded only when requested by the user. The exception to this is if the cache of data is still current. Whether this data is current is determined by the Request Data On Demand Maximum Sample Age (sec) field in the Collection Advanced tab. With On Demand monitoring, the Tivoli Enterprise Portal response time is slower because the data is collected as requested. However, the retrieved data is the most current available. CPU and memory usage on the monitored systems is lower for On Demand monitoring.

  - **Fixed Interval** - when the monitoring method is set to Fixed Interval, the Monitoring Agent will upload sample data from the data collectors at regular fixed intervals, and respond to a user request using the latest cache of sample data gathered at the last interval. With Fixed Interval monitoring, CPU and memory usage can be higher because more data is collected more frequently but Tivoli Enterprise Portal response time is faster as the data is more readily available. Data samples are calculated for the same time interval and as a result are more consistent.
- **Resource Data Monitoring** - in this field, specify whether resource data is monitored from connected data collectors. The following values can be entered in this field:
  - **Disable** - resource data is not collected by the Monitoring Agent and is not displayed in the Tivoli Enterprise Portal.
  - **Enable** - resource data is collected by the Monitoring Agent and is displayed in the Tivoli Enterprise Portal.
- **Resource Data Monitoring Method** - in this field, specify the monitoring methodology used by the agent to govern when it uploads resource data from connected data collectors. The following values can be entered into this field
  - **On Demand** - when the monitoring method is set to On Demand, data is uploaded only when requested by the user. The exception to this is if the cache of data is still current. Whether this data is current is determined by the Resource Data On Demand Maximum Sample Age (sec) field in the Collection Advanced tab. With On Demand monitoring, the Tivoli Enterprise Portal response time is slower as the data is collected as requested. However, the retrieved data is the most current available. CPU and memory usage on the monitored systems is lower for On Demand monitoring.
  - **Fixed Interval** - when the monitoring method is set to Fixed Interval, the Monitoring Agent will upload sample data from the data collectors at regular fixed intervals, and respond to a user request using the latest cache of sample data gathered at the last interval. With Fixed Interval monitoring, CPU and memory usage can be higher because more data is collected more frequently but Tivoli Enterprise Portal response time is faster as the data is more readily available. Data samples are calculated for the same time interval and as a result are more consistent.
- **Garbage Collection Monitoring** - in this field, specify if verbose garbage collection output monitoring is enabled. The following values can be entered in this field:
  - **Disable** - verbose garbage collection output data is not collected by the Monitoring Agent and is not displayed in the Tivoli Enterprise Portal.
  - **Enable** - verbose garbage collection output data is collected by the Monitoring Agent and is displayed in the Tivoli Enterprise Portal.

Application Dashboard (Basic) tab

The Application Dashboard (Basic) tab contains the following fields:

- **Application Fair Completion Rate Threshold (%)** - Defines the default completion percentage for application fair availability threshold.
- **Application Bad Completion Rate Threshold (%)** - Defines the default completion percentage for application bad availability threshold.
- **Application Fair Resource Usage Threshold (%)** - Defines the threshold percentage for fair usage level of an application resource.
- **Application Bad Resource Usage Threshold (%)** - Defines the threshold percentage for bad usage level of an application resource.

- **Application Resource Usage Monitoring Cutoff Threshold (%)** - Defines the cutoff threshold percentage for application resources usage monitoring.
- **Request Monitoring Control Level** - Specifies the request monitoring control level for the server. The following entries can be entered to this field:
  - **Application** Request monitoring settings are defined for each application independently.
  - **Server** Request monitoring settings are defined on the server level

The Request Monitoring Control Level option gives you more control over request monitoring settings. In certain sequences, you can benefit from locking the request monitoring control on the server level, because you can change data collector monitoring level in one place.

In ITCAM For Application Diagnostics, the request data monitoring level (Level1 or Level2) that displays on the Tivoli Enterprise Portal depends on the **Request Monitoring Control Level** setting you choose.

- If you select the **Request Monitoring Control Level** as **Application** (the default setting), the monitoring level is set separately for every application. For a newly discovered application, the current monitoring level for the agent is applied. Therefore, to display Level2 data for an application in the Tivoli Enterprise Portal, you must set the monitoring level for the application using the "Set_Application_Monitoring: Set monitoring" on page 504 Take Action command.
- If you select the **Request Monitoring Control Level** as **Server**, the level is set for the server and affects all applications. If you set the Request Data Monitoring Level at Level2 using the "Start_Request_Monitoring : Begin reporting request data" on page 506 Take Action command, the Request Data Monitoring Level for all the applications in this server is Level2. While you can still use the **Set_Application_Monitoring** Take Action command to set the monitoring level for an individual application, the level does not apply as a server-wide level is in operation. To apply the levels set for individual applications, change the **Request Monitoring Control Level** setting to **Application**.

Collection Advanced tab

The Collection Advanced tab contains the following fields:

- **Request Data On Demand Maximum Sample Age (sec)** - The maximum allowed age of sample request data in seconds before collecting a new sample of data. If the monitoring method is set to On Demand, when a user request is received, and the current sample cache is older than the value specified, then the Monitoring Agent uploads a new sample before servicing the request. Two successive on-demand requests received from users within the period specified by the maximum sample age return the same results without incurring the CPU and memory cost of a new data sample.
- **Request Data Fixed Interval between Collections (sec)** - The amount of time in seconds between uploads of sample request data from the data collectors to the Monitoring Agent when the monitoring method is set to Fixed Interval. When a user request is received, it is serviced from the latest uploaded sample.
- **Request Data Sampling Rate (%)** - The percentage of requests that are sampled for request data monitoring.
- **Resource Data On Demand Maximum Sample Age (sec)** - The maximum allowed age of sample resource data in seconds before collecting a new sample of data. If the monitoring method is set to On Demand, when a user request is received and the current sample cache is older than the value specified in this field, then the Monitoring Agent uploads a new sample before servicing the

request. Two successive on-demand requests received from a user within the period specified by the maximum sample age return the same results without incurring the CPU and memory cost of a new data sample.

- **Resource Data Fixed Interval between Collections (sec)** - The amount of time in seconds between uploads of sample resource data from the data collectors to the agent. When a Tivoli Enterprise Portal request is received, it is serviced from the latest uploaded sample.
- **Garbage Collection Polling Interval (sec)** - The interval in seconds between the Monitoring Agent scanning the verbose Garbage Collection output.
- **Log Scan Polling Interval (sec)** - The interval in seconds between the Agent scanning the application server standard output log for changes.

Application Dashboard (Auto Threshold) tab

The Application Dashboard (Auto Threshold) tab contains the fields:

- **Response Time Selection (%)** - Defines the percentage from baseline to be used for response time auto-thresholding.
- **Response Time Deviation (%)** - Defines the deviation for baseline selection to be used for response times auto-thresholding.
- **Fair Response Time Projection (%)** - Defines the percentage to derive the fair response time threshold from the baseline selection.
- **Bad Response Time Projection (%)** - Defines the percentage to derive the bad response time threshold from the baseline selection.

Workspace link to managing server visualization engine

With the appropriate configuration and permissions, you can access the managing server visualization engine from specific workspaces in the WebSphere Tivoli Enterprise Portal. Instead of opening another browser and clicking the relevant link, you can access the managing server visualization engine from Tivoli Enterprise Portal using a link called **Diagnostic Server Activity Display**. Some of the workspaces have additional links you can use to access the managing server visualization engine. All these links begin with the word **Diagnostic**. When you log in to the **Welcome to the Application Monitor** page, the information displayed is specific to the content in the Tivoli Enterprise Portal workspace you selected. You also can manually create your own links to the managing server visualization engine using the **Link Wizard**, which is available on all workspaces.

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

You can access the managing server visualization engine from the following workspaces.

- WebSphere Agent
- Request Analysis
- Garbage Collection Analysis
- Datasources
- JMS Summary
- Web Applications
- EJB Containers
- DB Connection Pools
- J2C Connection Pools
- Thread Pools

Here is a list of the managing server visualization engine links available from the workspaces:

- **Diagnostic Server Activity display**. Use this link to diagnose application problems, for example, slow transactions or high response times. This link is available from all workspaces except Request Analysis and Garbage Collection Analysis.
- **Diagnostic In-Flight Request Search.** Use this link to identify any hanging transactions. This feature is only available from the WebSphere Agent and Request Analysis workspaces.
- **Diagnostic Recent Completed Requests**. The Recent Requests tab displays data regarding recently completed server requests. This feature is only available from the Request Analysis workspace.
- **Diagnostic SMF Data**. This feature is only available you are using a z/OS data collector and in the Request Analysis workspace. The SMF Overview displays summary information for all the resources on the selected application server.
- **Diagnostic JVM Thread Display** Use this link to diagnose application problems, for example, slow transactions by examining threads running in JVM. This link is available from the Thread Pools workspace.
- **Diagnostic Memory Leak** Use this link to diagnose memory leak problems. This link is available from the Garbage Collection Analysis workspace.

For information about accessing workspaces, see "Accessing the managing server visualization engine from Tivoli Enterprise Portal workspaces" on page 274.

Prerequisites for access

The following conditions must be met to gain access to the managing server visualization engine through the Tivoli Enterprise Portal.

- ITCAM for Application Diagnostics managing server and ITCAM Data Collector for WebSphere must be installed in your environment.
- Tivoli Enterprise Portal users must be members of the **Diagnostic Users Group** within the Tivoli Enterprise Portal. For more information, see "Granting users access to managing server visualization engine from Tivoli Enterprise Portal" on page 273.
- During the installation of the managing server, kernel properties in the managing server must be set up accordingly with the correct host name and port number. This action is completed by the user installing the managing server.

You can also create your own links to the managing server visualization engine using the **Link Wizard**.

For more information, see "Creating links to the managing server visualization engine using the Link Wizard" on page 278

Kernel Settings to access the managing server visualization engine through the Tivoli Enterprise Portal

If users access the managing server visualization engine through the Tivoli Enterprise Portal, the kernel properties in the managing server must be set up accordingly with the correct host name and port number.

The following properties must be added to `kl1.properties` and `kl2.properties` (By default, the managing server installer replaces *@{HOST_VE}* and *@{PORT_VE_HTTP}* at managing server installation time) :

- *ve.host=@{HOST_VE}*
- *ve.port=@{PORT_VE_HTTP}*

At kernel startup time, the kernel must read these two properties and set them as part of properties in *PROBE_CONFIG.PROPS*. If a user changes the VE host name or port number, then the kernel needs to be restarted. Use the following steps to start and stop the kernel In `$MS_HOME/bin`:

1. To start kernel, issue:`./amctl.sh wd<kernel count> start`
2. To stop kernel, issue: `./amctl.sh wd<kernel count> stop`

where <*kernel count*> is 1 by default.

Granting users access to managing server visualization engine from Tivoli Enterprise Portal

As a user, you must be set up as a member of the **DIAGNOSTIC USERS** group in the Tivoli Enterprise Portal, otherwise you cannot access the visualization engine from the Tivoli Enterprise Portal. The default administration user **Sysadmin** is automatically a member of this group. Any user with administrator permissions can add or remove additional users to the **DIAGNOSTIC USERS** group.

Before you begin

To complete this task you must be a user with administrator permissions to add or remove additional users to the **DIAGNOSTIC USERS** group. For more information about access, see "Prerequisites for access" on page 272.

Procedure

1. From the Tivoli Enterprise Portal main menu, click **Edit > Administer Users**.
2. In the **Administer Users** window, click the **Users Groups** tab.
3. Click the group name, in this case **DIAGNOSTIC USERS**.
4. Click the **Members** tab to view existing members of this group and to assign additional users.
5. To add users, in the **Available Members** section, select the users that you want to assign to the Group.
6. Click the left arrow to move the selected users to the **Assigned Members** section of the window.
7. Click **Apply** and **OK** to implement the changes.
8. To remove a user from the group, click the **Assigned Members** tab, and select the users that you want to remove.
9. Click the right arrow to move the selected users to the **Available Members tab**. Then click **Apply** and **OK**.

What to do next

Users who are members of this group can access the managing server visualization engine from the Tivoli Enterprise Portal. See "Accessing the managing server visualization engine from Tivoli Enterprise Portal workspaces" on page 274.

Adding the LDAP user to Tivoli Enterprise Portal user accounts:

About this task

To add the LDAP user to Tivoli Enterprise Portal user accounts, use Tivoli Enterprise Portal user administration.

1. In the Tivoli Enterprise Portal main menu, select **Administer Users**.
2. Click **Create New User** to create a user profile from defaults, or **Create Another User** to create a user profile as a copy of an existing one.
3. In the **Modify User** window, enter the user name for the new user in the **User ID** field.
4. In the **Distinguished Name** field, enter the following string:

   `uid=username,cn=users,dc=ibm,dc=com`

   This string registers the LDAP user with Tivoli Enterprise Portal. If you are using an existing LDAP configuration, use the applicable distinguished name.

Tip: For more information about Single sign-on, see **Appendix N Setting Up single sign on for Tivoli Enterprise Portal Users** in the *ITCAM for Application Diagnostics managing server Installation and Customization Guide.*

Accessing the managing server visualization engine from Tivoli Enterprise Portal workspaces

Use the following links for information about how to access the managing server visualization engine from the Tivoli Enterprise Portal workspaces.

To ensure that you have access to the managing server visualization agent, see "Prerequisites for access" on page 272 before you begin.

- "Accessing the managing server visualization engine from the WebSphere Agent workspace" on page 339
- "Accessing the managing server visualization engine from the Request Analysis workspace" on page 318
- "Accessing the managing server visualization engine from the Garbage Collection Analysis workspace" on page 301
- "Accessing the managing server visualization engine from the Datasources workspace" on page 291
- "Accessing the managing server visualization engine from the JMS Summary workspace" on page 306
- "Accessing the managing server visualization engine from the Web Applications workspace" on page 336
- "Accessing the managing server visualization engine from the EJB Containers workspace " on page 298
- "Accessing the managing server visualization engine from the DB Connection Pools workspace" on page 294
- "Accessing the managing server visualization engine from the J2C Connection Pools workspace" on page 305
- "Accessing the managing server visualization engine from the Thread Pools workspace" on page 334

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271

- "Access the managing server visualization engine from the Tivoli Enterprise Portal"

Access the managing server visualization engine from the Tivoli Enterprise Portal

You can access the managing server visualization engine from links in the ITCAM Agent for WebSphere Applications workspaces.

When you access the managing server visualization engine in this way, the managing server visualization engine displays in a browser view in a workspace. The Tivoli Enterprise Portal navigation tree is automatically hidden in the workspace. To show or hide the Tivoli Enterprise Portal navigation tree, click the small black arrow.

The following table displays a list of Tivoli Enterprise Portal workspaces that have links to the managing server visualization engine.

Table 17. Tivoli Enterprise Portal workspaces that have links to managing server visualization agent

Workspace: Table View Name	**Number of links to managing server visualization agent**	**Link Name**	**Link Target Pages managing server visualization agent**	**Pre-populated information in the link page to managing server visualization engine**
WebSphere Agent Summary Status > Application Servers WebSphere Agent Summary Status > Application Servers WebSphere Agent Configuration > Application Servers	2	• Diagnostic Server Activity Display • Diagnostic In-Flight Request Search	• Server Activity Display – Active Requests • In-Flight Request Search	The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace.
OS Stack >Current OS Stack Summary	3	• Diagnostic Server Activity Display • Diagnostic In-Flight Request Search • <platform OS> <platform> is one of the following operating systems: Linux, UNIX, Windows or z/OS	• Server Activity Display – Active Requests • In-Flight Request Search • Using the dynamic workspace link to the corresponding OS agent workspace. For z/OS, the link is to OMEGAMON XE for z/OS.	The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace.

Table 17. Tivoli Enterprise Portal workspaces that have links to managing server visualization agent (continued)

Workspace: Table View Name	Number of links to managing server visualization agent	Link Name	Link Target Pages managing server visualization agent	Pre-populated information in the link page to managing server visualization engine
JVM Stack Trend >JVM Stack Trend	1	Diagnostic Memory Leak	Memory Leak Analysis	The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace.
Request Analysis > Requests – Current Interval	3	• Diagnostic Recent Completed Requests • Diagnostic In-Flight Request Search • Diagnostic SMF Data (z/OS only)	• Server Activity Display – Recent Requests • In-Flight Requests • SMF Data (for z/OS data collectors only)	• The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace. • Content in Request Detail column of Requests table view in Tivoli Enterprise Portal is pre-populated in the following fields: – Recent Requests: Client Request – In-Flight Request Search: Search Request/ Transaction field
Garbage Collection Analysis >Garbage Collection Analysis	1	Diagnostic Memory Leak	Memory Leak Analysis	The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace.

Table 17. Tivoli Enterprise Portal workspaces that have links to managing server visualization agent (continued)

Workspace: Table View Name	Number of links to managing server visualization agent	Link Name	Link Target Pages managing server visualization agent	Pre-populated information in the link page to managing server visualization engine
Thread Pools >Thread Pools	1	Diagnostic JVM Thread Display	JVM Thread Display	The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace.
Datasources > Datasources – Current Interval Web Applications >Web Applications EJB Containers >EJB Containers JMS Summary >JMS Summary – Current Interval DB Connection Pools > DB Connection Pools J2C Connection Pools > J2C Connection Pools	1	Diagnostic Server Activity Display	Server Activity Display – Active Requests	The Group Server drop-down menu is pre-populated based on data collector information from Tivoli Enterprise workspace.

The **Server Group** feature that displays applies to the managing server visualization engine. When you access any of these pages from the Tivoli Enterprise Portal, the information in the **Groups** and **Servers** fields is automatically populated based on the data collector associated with the link and workspace you selected in the Tivoli Enterprise Portal.

Groups are a set of application servers which have similar functionality. All configured data collectors are automatically in the **Unassigned Servers Group**. The relationship between Server Group and data collector is many to many. A data collector can belong to one or more server groups. A server group can have one or more data collectors. You can add data collectors to groups using the **Server Management** functionality in the managing server visualization engine. For more information about adding data collectors to Server Groups, refer to the Composite Application Manager Help in the managing server visualization engine interface.

The **Server Activity Display** section has three tabs.

- **Active Requests**: provides real-time request or transaction data for an application server at the time the page displays.

- **Recent Requests**: displays the last 100 or less completed request or transaction data for an application server.
- **Lock Contentions**: displays requests that are hanging because they are waiting on a lock. The data shows data that is currently locked and the item that is waiting to be locked.

The **Active Requests** tab and the **Recent Requests** tab have a toolbox icon  . You can click this icon to access direct links to the following features:

- JVM Display
- System Resources
- Monitoring On Demand
- data collector Properties
- Trap and Alert Management

When you access the managing server visualization engine in this way, the managing server visualization engine displays in a browser view in a workspace. The Tivoli Enterprise Portal navigation tree is automatically hidden in the workspace. To show or hide the Tivoli Enterprise Portal navigation tree, click the small black arrow.

Creating links to the managing server visualization engine using the Link Wizard

Throughout the Tivoli Enterprise Portal, you can use the **Link Wizard** to manually create links to the managing server visualization engine.

Creating links

You can access the Link Wizard feature from other views and workspaces in the Tivoli Enterprise Portal.

1. To access the **Link Wizard**, from any of the tables or views, right-click the link icon and click **Link Wizard**.
2. Follow the steps in the wizard to do one of the following actions:
   - Create a link.
   - Modify an existing link.
   - Delete one or more links.
3. Type the name and description of the link.
4. Choose one to the following options with the aid of the descriptions in the **Link Wizard**.
   - Dynamic
   - Absolute
   - Relative
5. Choose the option that you want to use. Follow the instructions in the help within the application until you get to the **Workspace Link Wizard Parameters** page.

Adding parameters to the Link Wizard

When you get to the **Workspace Link Wizard Parameters** page in the Link Wizard you must manually add these two symbols **VEHOSTPORT** and **VEPATH** and add information to these parameters. There are two predefined workspaces which have an embedded browser as its only view.

- Diagnostic Link for Agent. Choose this workspace if the link is from the agent level workspaces.
- Diagnostic Link. Choose this workspace if the link is from the server level workspace.

To add Symbols to the Link Wizard, complete the following steps:

1. Click **Symbol** and type **VEHOSTPORT**.
2. Click **OK** to add the Symbol.
3. To add an expression, select **VEHOSTPORT**, and click **Modify Expression**.
4. **Basic Setup:** In the text field, type the expression that you want to add. The value for the **VEHOSTPORT** is **<your ve host>:<your ve port>.** For example, if in your environment, the managing server visualization engine is installed on host1, with port 9080, then the **VEHOSTPORT** value is: *"host1:9080"*.

   Important:
   - Double quotation marks are required in the expression.
   - If your managing server visualization engine host or port information changes, then you must update the link you defined and correct the **VEHOSTPORT** information manually.

   Advanced Setup: In the text field, type the expression that you want to add. The value for the **VEHOSTPORT** value is *CALL(candle.kwj.ve.ITCAMLinkHelper, getVehostportForWASServer, null, null, $kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$)*. To get the expression: *$kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$,* in the Expression Editor, click **Symbol**.
5. Click **Origin Node** and click **OK** to get *$kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$.*

   When using the Advanced set up option, if your managing server visualization agent host or port information changes, ITCAM for Application Diagnostics custom code updates the changes automatically, you do not have to modify the links you defined manually to correct the information.
6. To add an expression, **Basic Setup:** the **VEPATH** value can be set as *"am/ve/sad/threadList?mappingTEPUrl=true"*

   Important:
   - Double quotation marks are required in the expression.
   - When you use the Basic Setup option, you must select the server group and the server in the managing server visualization engine yourself when you log in to managing server visualization engine.

   To add an expression, **Advanced Setup:** Set the **VEPATH** value in the Expression Editor as:*"am/ve/sad/threadList?mappingTEPUrl=true&server=" + CALL(candle.kwj.ve.ITCAMLinkHelper, getServerIdForWASServer, null, null, $kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$)*

   Where the correct expression of *$kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$* can be found by using Symbols view under Expression Editor: *$kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$ (Symbol: Origin Node)*.
7. Review the details in the **Workspace Link Wizard - Summary**. Click **Finish**.
8. If you use the **Advanced Setup** options in Step 4 and Step 6, modify the **linkIsEnabled** parameter in the Workspace Link Wizard Parameters view with expression: *CALL(candle.kwj.ve.ITCAMLinkHelper, isEnabledForWASServer, null,*

null, $kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$) && $kfw.TableRow:ATTRIBUTE.<table_name>.ASID$!= -3. In the **Expression Editor**, use the Symbols view to find expressions of *$kfw.TableRow:ATTRIBUTE.<table_name>.ORIGINNODE$ (Symbol: Origin Node)*and *$kfw.TableRow:ATTRIBUTE.<table_name>.ASID$ (Symbol: ASID)*.

Manually adding your own links to the Link Wizard

The minimum settings are **VEHOSTPORT** and **VEPATH**. If you need to link to a different page, you must change this value to corresponding URL path. Here is a list of frequently used managing server visualization engine pages and URL paths:

- Enterprise Overview: `am/home`
- Server Overview: `am/ve/serverOverview`
- Server Statistics Overview : `am/avm/main`
- Server Activity Display: `am/ve/sad/threadList`
- Memory Leak: `am/ve/memory/leakReport`
- JVM Thread Display: `am/ve/jtd/threadGroupList`
- Monitoring On Demand: `am/ve/mod/console`
- Trap and Alert Management: `am/trap`
- System Resources:`am/ve/jmxbrowser`

Important: If you change the host port number at any point, you must modify these two properties again and the kernel properties on the Managing Server installation must be restarted to identify the changes.

Link anchor

You can use the **Link anchor** option to access the **Link anchor properties** window. You can use the **Link anchor properties** window to display visual indicators on tables where customized links have been defined, and to establish a default link that opens when the user clicks the indicator.

1. From any of the tables, right-click the choose link icon and click **Link Anchor** to display the **Link anchor properties** window.
2. Depending on the area of the table item you select, the window displays the following information:
   - Default no link
   - Show Link indicator
   - Link indicator always enabled.
3. For more information about the uses of these items, refer to the help within the **Link anchor properties** window.

"Workspace link to managing server visualization engine" on page 271

Workspace link to ITCAM for SOA

You can navigate from ITCAM Agent for WebSphere Applicatins workspaces to ITCAM for SOA.

If ITCAM for SOA is installed and configured for the same application server, you can navigate from the "Application Server Summary and Business Process Manager Summary workspaces" on page 260 and from the "Application Health

workspace" on page 283 to workspaces provided by ITCAM for SOA. ITCAM for SOA is available with ITCAM for Applications. For more information, see the *ITCAM for SOA User Guide*.

A *service group* is a set of related service operations that collectively might represent or encompass a business function or application in your enterprise. This can consist of a service flow, a subset of a flow, or any collection of operation aggregates that represent something meaningful to you in your monitored environment. A *process group* is a Business Process Management (BPM)-specific group that aggregates elements that are part of a BPM solution, such as applications, modules, components, or operations. A process group allows the monitoring of interactions between the components of the solution. This monitoring provides BPM-specific context information.

Alarm Manager workspace

This workspace displays aggregated information about the alarms for each work manager.

This workspace displays data provided by the Alarm Manager attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains:

- Highest Alarm Rate bar chart, which displays the five highest number of alarms fired per second
- Work Manager Alarms report, which displays detailed information about the alarms for each work manager

Accessing the Alarm Manager workspace

To access this workspace from the Thread Pools workspace, use one of the following procedures:

- Within the Navigator, right-click the **Thread Pools** entry; then from the menu select **Workspace** > **Alarm Manager**.
- From the primary Tivoli Enterprise Portal menu, expand the **View** menu, and select **Workspace** > **Alarm Manager**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Allocation Failures workspace

This workspace summarizes all the heap-allocation failures that occurred within the Java Virtual Machine (JVM) over the current interval and that caused the JVM to initiate garbage collection.

This workspace displays data provided by the Allocation Failure attributes.

Note to Solaris and HP-UX users: Allocation-failure information is not recorded on these platforms; hence this workspace is always empty.

The predefined workspace contains the following items:

- Allocation Failure Elapsed Times bar chart, which displays the number of allocation failures during the current interval
- Heap Usage bar chart, which displays the heap usage for this JVM. The bar's fail over gives the allocation-failure ID number followed by a range of recording times. This allocation-failure number displays in the Allocation Failures report and associates each bar with that particular row within the report
- Allocation Failures report, which displays information about the heap-allocation failure that caused the Java Virtual Machine hosting the application server to start its garbage-collection routine. The Allocation Failures report includes the ASID field.

Accessing the Allocation Failures workspace

To access this workspace from the Garbage Collector Activity workspace, complete the following steps:

- From the Garbage Collection Analysis report, right-click the link icon and from the menu, click **Allocation Failures**. Note that in the Allocation Failure workspace, the ASID is displayed in the Allocation Failures report.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Application Trend at L1 workspace

This workspace displays access trends for individual applications. It provides trend information for request and response times. It also provides trend information for application errors. This workspace provides a broad overview of the health of an application and draws data from multiple attribute groups.

This workspace displays data provided by the Request Analysis and Request Times and Rates attribute groups.

The predefined workspace contains:

- Selected Application Summary report displays application name, average request response time, average request completion rate, error rate, and ASID. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.
- Response Time Trend chart displays summary trend times for the overall response time for the selected application.
- Error Rate Trend chart displays the error rate for the application.

- Request Rate Trend chart displays the number of requests completed per second for the application.

Accessing the Selected Application - Configuration workspace

Access this workspace using one of the following methods:

- Double click any application in the application server Summary workspace.
- Right-click on application icon in the application server Summary workspace and select **Link To**, then select **Application Trend at L1**.

For additional information, see:

- application server Summary Workspace
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Application Trend at L2/L3 workspace

This workspace displays access trends for individual applications. It provides trend information for request and response times. It also provides trend information for application errors. This workspace provides a broad overview of the health of an application and draws data from multiple attribute groups.

This workspace displays data provided by the Request Analysis and Request Times and Rates attribute groups.

The predefined workspace contains:

- Selected Application Summary report displays application name, average request response time, average request completion rate, error rate, and ASID. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.
- Response Time Trend chart displays trend times for the following different elements in milliseconds: JNDI, JMS, Application, JCA, SCA, Servlet, EJB, and web service requests.
- Error Rate Trend chart displays the error rate for the application.
- Request Rate Trend chart displays the number of requests completed per second for the application.

Accessing the Selected Application - Configuration workspace

Access this workspace using one of the following methods:

- Double click any application in the application server Summary workspace.
- Right-click on application icon in the application server Summary workspace and select **Link To**, then select **Application Trend at L2/L3**.

For additional information, see:

- application server Summary Workspace
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Health workspace

The workspace displays the information about the real-time health status of applications monitored by the Tivoli Enterprise Monitoring Agent.

The health status information is collected from the following sources.

- Request Metrics - performance data that measures request execution time collected from the ITCAM instrumentation points in the application code.
- Resource Metrics - pool usage and container performance statistics collected from the corresponding PMI modules or MBeans.
- GC Metrics - metrics on garbage collection frequency and performance collected from parsing of the GC verbose log file when it is enabled for the application server JVM.
- OS metrics - metrics collected about the JVM process and the whole system execution, such as CPU used percentage, paging rate, and so on.

Additionally, the monitoring agent uses thresholds, called Application Health Indicators, to determine the quality of the application service. For request response times, thresholds are assigned automatically during baselining. You can also manually customize the thresholds. There are three monitored application tiers evaluated for health status.

- Client Tier provides performance data and status of application execution in servlets/JSPs or portal containers as well as corresponding thread pools servicing these containers.
- Application Tier provides application execution metrics of EJB containers and custom requests.
- Backend Tier provides application execution in JDBC, JCA, JMS, JNDI API calls.

This workspace displays data provided by the Application Health Status attributes.

By default, the predefined workspace has the following views:

- Situation Event console view, which shows the event console with activity associated with the Application Health Summary Navigator item and any other workspaces in the group, as well as linked workspaces. The Navigator displays an event icon overlaid on the Application Health Summary node when a situation becomes true. The report is useful when multiple alerts are raised as you can see them all in a single filtered view.
- Application Health Summary report, which shows the report of the application name, status, and health indicator for client, application, and backend tiers health status.

Important: Due to the runtime MBeans configuration, the Tivoli Enterprise Monitoring Agent (Tivoli Enterprise Monitoring Agent) can find only composition units of business-level applications that associate with either web or EJB modules.

If any applications are running on the Business Process Manager server and ITCAM for SOA is installed on the server, an additional link is available. In the Application Health Summary view, right-click any application and select **Selected Application - Group Summary** to view the Group Summary workspace, provided by ITCAM for SOA. In this workspace, you can see the status of all process groups that include any components or operations that are a part of the selected application.

Accessing the Application Health Summary workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems or UNIX Systems, as appropriate for the node you are monitoring.

2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere Application server entry of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Application Health** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Registry workspace

This workspace displays the information about the server configuration for the application.

This workspace displays data provided by the Application Monitoring Configuration attributes.

The predefined workspace contains:

- Situation Event Console report, which shows the event activity for situations associated with the current Navigator item. The Navigator alerts you when a situation becomes true by overlaying the Navigator item with an event indicator. This report is useful when multiple alerts are raised and you might not know newly arrived alerts just by looking at the indicator.
- Application Configuration report, which shows the configurations that are discovered, stored and managed for WebSphere applications running within that application server.

Accessing the Application Registry workspace

To access this workspace from the Application Health Summary workspace, use one of the following procedures:

- Within the Navigator, right-click the **Application Health** entry; then, from the menu, click **Workspace** > **Application Registry**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Application Registry**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Business Process Manager workspace

The Business Process Manager workspace displays overall statistics for a Business Process Manager server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the WebSphere application server heap over time. The chart flyovers display the exact values.

This view displays data provided by the Garbage Collection Analysis attributes.

- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes.
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes.
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the application server attributes.
- Application server summary report, which displays overall information about this WebSphere application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the application server attributes.

Accessing the Business Process Manager workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, right-click a **Business Process Manager** entry, and select **Workspaces** > **Business Process Manager**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Cache Analysis workspace

This workspace reports information about the dynamic cache.

WebSphere application server consolidates several caching activities, including servlets, web services, and WebSphere commands, into one service called the *dynamic cache*. These caching activities work together to improve application performance. The activities share many configuration parameters, which are set in a *dynamic cache service* in an application server. The dynamic cache works within a Java Virtual Machine (JVM) within an application server, intercepting calls to cacheable objects, for example, through a servlet service method or an execute method on a command. The dynamic cache either stores the object output to or serves the object content from the dynamic cache.

This workspace displays data provided by both the Dynamic Cache attributes and the Dynamic Cache Templates attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report

resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.

- Because of high overhead, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- In-memory Cache Sizes - History graph, which shows the sizes of in-memory cache entries for the specified interval
- Highest Miss Rates bar chart, which shows the most frequent cache entry misses (per second). The Y-axis headings correspond to the row number of the Dynamic Cache Templates report
- Dynamic Cache report, which displays information about the dynamic cache, including cache sizes and timeout rates
- Dynamic Cache Templates report, which displays information about the cache template data. A cache template is an object type defined by a cache policy specified in WebSphere application server file cachespec.xml. A cache policy specifies the caching rules and indicates what is cached, the invalidation and timeout conditions, and other data

Accessing the Cache Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Cache Analysis** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Client Communications workspace

This workspace provides overall Service Integration Bus communication performance data and counters for all clients connected to this application server. WebSphere application server 5.1 based products do not support this workspace.

This workspace displays data provided by the Client Communications attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report

resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.

- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Connection Count - History graph, which shows the number of API sessions used by clients that were network connected to this application server
- Error Count - History graph, which shows the communication errors that occurred and that resulted in the disconnection of a network connection to a client
- Communication Statistics report, which displays information about client communications, including API connections, errors, reads, writes, sent priority, received priority, MessageSent priority, and MessageReceived priority

Accessing the Client Communications workspace

To access this workspace from the Platform Messaging workspace, use one of the following procedures:

- Within the Navigator, right-click the **Platform Messaging** entry; then from the menu, click **Workspace** > **Client Communications**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Client Communications**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Container Object Pools workspace

This workspace displays aggregate information about the object pools associated with Enterprise Java Beans (EJBs). It provides a view of pool performance for all Enterprise Java Beans deployed to each container.

This workspace displays data provided by the Container Object Pools attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Object Pool Rates - History graphs, which show the retrieval, return, discard, and drain rates for the EJBs in each EJB container
- Container Object Pools report, which displays:
  - Aggregated information for each defined EJB container that aggregates bean object pool performance for all Enterprise beans deployed to that container
  - Aggregated information for the application server that aggregates bean object pool performance data for all Enterprise beans deployed to the application server

Accessing the Container Object Pools workspace

To access this workspace from the EJB Containers workspace, use one of the following procedures:

- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Container Object Pools**.
- Within the Navigator, right-click the **EJB Containers** entry; then, from the menu, click **Workspace** > **Container Object Pools**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Container Transactions workspace

This workspace displays data about the activities and transactions running in each application server.

This workspace displays data provided by the Container Transactions attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Transaction Rates - History graph, which shows the per-second begin, commit, and rollback rates over time for local and global transactions
- Transaction Durations - History graph, which shows the amount of time it takes to complete local and global transactions
- Container Transactions report, which displays performance information for global and local transactions that run in each defined EJB container and an aggregated value for all transactions that run in the application server

Accessing the Container Transactions workspace

To access this workspace from the EJB Containers workspace, use one of the following procedures:

- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Container Transactions**.
- Within the Navigator, right-click the **EJB Containers** entry; then, from the menu, click **Workspace** > **Container Transactions**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Data sources workspace

The data sources workspace displays statistical data for the data sources that your applications reference when accessing databases.

This workspace displays data provided by the Datasources attributes.

The predefined workspace contains the following items:

- Worst Datasource Query Times bar chart, which shows the longest times (in milliseconds) the application spent waiting to retrieve data from the database during the specified interval
- Worst Datasource Update Times bar chart, which shows the longest times (in milliseconds) the application spent updating data within the database during the specified interval
- Datasources - Current Interval report, which displays database usage information. For example, this report shows traffic information such as the time the application spent trying to connect to the database, and the total and average processing times for database queries and updates.

Accessing the Data sources workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Datasources** entry.

Selected Data source - History workspace

The Selected Datasource - History workspace displays the historical information that corresponds to the information in the Datasource workspace for a selected data source. Historical information is collected over a specific time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected Data source - History workspace

To access this workspace from the Datasource workspace, use one of the following procedures:

- From the Datasources - Current Interval report, right-click the link icon and from the menu, click **Selected Datasource - History**.
- From Worst Datasource Query Times bar chart or the Worst Datasource Update Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Datasource - History**.

Selected Data source - Datasource Trend workspace

The Datasources Trend workspace displays information about datasource connections. This workspace displays data provided by the Datasources attributes.

This predefined workspace contains the following items:

- Current Datasources report displays datasource name, total wait time, connection rate, connection average wait time and connection max wait time. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.
- Total Wait Time chart displays the total time that applications had to wait for a connection to the data source.
- Connection Rate Trend chart displays the number of connection requests created for the data source per second.
- Connection Average and Max Time Trend chart display the average time in milliseconds and the worst-case time in milliseconds that applications had to wait for a connection.

Accessing the Selected Data source - Datasource Trend workspace

Right-click the **Datasources** workspace, select **Workspace** and then **Datasource Trend** workspace

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the Datasources workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal, access the **Datasources** workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **Datasources - Current Interval** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display**.
2. If this is your first time to access the Managing server visualization agent during a session, you see a **Welcome to the Application Monitor** page.

3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need only to log in using your **User Name** and **Password** once per session. Click **OK** to display the **Server Activity Display** page in the managing server Visualization engine. You can use this page to diagnose application problems, for example, slow transactions or high response times.
4. In a z/OS environment, right-click the link icon on a row where the ASID column displays Summary.
   a. Click **Selected Datasource- Servant Regions**.
   b. In the **Selected Datasource - Servant Regions** table, right-click the link icon on a row.
   c. Click **Diagnostic Server Activity Display**.
5. For more information about the options available in a z/OS environment, see "Region workspaces in a z/OS environment" on page 346 and "Accessing a Region workspace" on page 348.
6. The results in this page relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Server Activity Display** page, and additional features refer to the Composite Application Manager help within the managing server visualization agent Interface.
7. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271.
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

DB Connection Pools workspace

This workspace displays information about the database connection pools associated with each application server.

You can use this workspace to monitor Java Database Connectivity (JDBC) performance for WebSphere application server applications. This workspace displays data provided by the DB Connection Pools attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of the high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must

first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Highest Average Pool Sizes bar chart, which shows the largest average size (that is, number of connections) for each database connection pool
- Worst Wait Times bar chart, which shows the worst wait times (in milliseconds) for each database connection pool
- Highest Allocation Rates bar chart, which shows the rate at which database connections are being made for each connection pool
- DB Connection Pools report, which displays information about the database connection pool for each defined data source. The report also displays an aggregated value that aggregates over all data sources. For example, this report displays the number of threads waiting for a connection and the number of connections created and released

Accessing the DB Connection Pools workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server entry of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **DB Connection Pools** entry.

Selected DB Connection Pool - History workspace

The Selected DB Connection Pool - History workspace displays historical information that corresponds to the information in the DB Connection Pools workspace for a selected connection pool. Historical information is collected over a specific time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected DB Connection Pool - History workspace

To access this workspace from the DB Connection Pools workspace, use one of the following procedures:

- From the DB Connection Pools report, right-click the link icon and from the menu, click **Selected DB Connection Pool - History**.
- From the Highest Average Pool Sizes bar chart, the Worst Wait Times bar chart, or the Highest Allocation Rates bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected DB Connection Pool - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the DB Connection Pools workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal access the DB Connection Pools workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **DB Connection Pools** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display**.
2. If this is your first time accessing the managing server visualization agent during a session, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need only log in once per session using your **User Name** and **Password**.
4. Click **OK** to display the **Server Activity Display** page in the managing server Visualization engine. You can use links to diagnose application problems, for example, slow transactions or high response times.
5. The results relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Server Activity Display** page and additional features, refer to the Composite Application Manager help within the managing server visualization agent Interface.
6. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

DCS Stacks workspace

This workspace displays aggregated information about each DCS stack within the entire WebSphere application server domain, including multiple nodes and servers.

This workspace displays data provided by the DCS Stack attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.

- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Highest Message Buffer Reallocations bar chart, which displays the highest number of message buffer reallocations that occurred as a result of inadequate buffer size
- Most Sent Messages bar chart, which shows most frequent number of message buffer reallocations that occurred as a result of inadequate buffer size
- High Severity Congestion Events bar chart, which shows the number of times that a high severity congestion event for outgoing messages was raised
- DCS Statistics report, which displays information for the DCS stack data, including incoming and outgoing message size, sent messages, and high severity congestion events

Accessing the DCS Stacks workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **DCS Stacks** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Destinations workspace

In this workspace, you can view performance data and counters for the destinations of a selected messaging engine.

A destination is a virtual location within a service integration bus, to which applications attach as producers, consumers, or both, to exchange messages. There are two types of destinations, queues and topic spaces. WebSphere application server 5.1 based products do not support this workspace. This workspace displays data provided by both the Topic Spaces attributes and the Queue attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report

resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.

- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains:

- Queue report, which displays information about the queue data. A queue is a destination for point-to-point messaging
- Topic Spaces report, which displays information about the topic space data. A topic space is a destination for publish/subscribe messaging

Accessing the Destinations workspace

To access this workspace from the "Messaging workspace" on page 311, complete one of the following steps:

- From the Messaging Engines report, right-click the link icon and from the menu, click **Destinations**.
- From Average Local Wait Time - History graph, Expired Messages - History graph, Incomplete Topic Publications - History graph, or Total Published - History graph, right-click any bar; then, from the menu, click **Link To** > **Destinations**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Durable Subscriptions workspace

In this workspace, you can view statistic counters for the durable subscriptions of a selected topic.

The default messaging provider supports the use of durable subscriptions to topics. This enables a subscriber to receive a copy of all messages published to a topic, even messages published during periods of time when the subscriber is not connected to the server. WebSphere application server 5.1 based products do not support this workspace. This workspace displays data provided by the Durable Subscriptions attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Durable Subscriptions report, which displays information about durable subscriptions that pertain to a selected topic, including the number of messages consumed, and message wait time

Accessing the Durable Subscriptions workspace

To access this workspace from the Destinations workspace, complete the following step:

- From the Topic Spaces report, right-click the link icon and from the menu, click **Durable Subscriptions**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

EJB Containers workspace

This workspace displays aggregated information about each defined EJB.

This workspace displays aggregated bean performance data for all Enterprise beans deployed to an EJB container. It also displays aggregated information for the application server that aggregates bean performance data for all Enterprise beans deployed on the application server. This workspace displays data provided by the EJB Containers attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Method Average Response Time - History graph, which shows the average response time for methods invoked by the EJBs in the container
- Method Invocation Rate - History graph, which shows the rate at which the methods of an EJB container were invoked
- EJB Containers report, which displays aggregated information for each defined EJB container that aggregates bean performance data for all Enterprise beans deployed to that container. The report also displays aggregated information for the application server that aggregates bean performance data for all Enterprise beans deployed to the application server. For example, this report displays load values, response times, and life cycle activities for Enterprise beans

Accessing the EJB Containers workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **EJB Containers** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the EJB Containers workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal access the EJB Containers workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **EJB Containers** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display**.
2. If this is your first time to access the managing server visualization engine during a session, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need only log in once per session using your **User Name** and **Password**.
4. Click **OK** to display the **Server Activity Display** page in the managing server Visualization engine. You can use link to diagnose application problems, for example, slow transactions or high response times.
5. The results relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Server Activity Display** page and additional features, refer to the Composite Application Manager help within the managing server visualization agent Interface.
6. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271

- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

Enterprise Java Beans workspace

This workspace reports information about each Enterprise Java Bean (EJB) defined for an EJB container.

The workspace provides information about these beans that relates to their identity, instrumentation level settings, creation and destruction of bean objects, response times, invocations, calls, and rates for retrievals, returns, and discards. This workspace displays data provided by the Enterprise Java Beans attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Worst Method Response Times bar chart, which shows the worst response times (in milliseconds) for methods invoked by each bean instance
- Highest Method Invocation Rates bar chart, which shows the methods that are invoked most often by each bean instance
- Enterprise Java Beans report, which shows performance information about each EJB deployed to the application server. This report displays information about bean activity, including the rates at which beans are being instantiated and destroyed

Accessing the Enterprise Java Beans workspace

You access this workspace from the EJB Containers workspace. To list the EJBs for all containers, use one of the following procedures:

- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Enterprise Java Beans**.
- Within the Navigator, right-click the **EJB Containers** entry; then, from the menu, click **Workspace** > **Enterprise Java Beans**.

To see the EJBs referenced by a specific EJB container, complete the following step:

- From the EJB Containers report, right-click the link icon and from the menu, click **Enterprise Java Beans**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collections - Selected Allocation Failure workspace

This workspace provides detailed information about the garbage-collection cycles that occurred in response to a specific heap-allocation failure that occurred within the Java Virtual Machine.

This workspace displays data provided by the Garbage Collection Cycle attributes.

Note to Solaris and HP-UX users: Allocation-failure information is not recorded on these platforms. Consequently, this workspace is always empty.

The predefined workspace contains the following items:

- GC Elapsed Times bar chart, which breaks down the mark, sweep, and compact times (in milliseconds) for each garbage-collection cycle that occurred for the selected allocation failure
- Heap Usage bar chart, which displays the JVM heap usage (kilobytes in use, freed, and free at start of garbage collection) for each garbage-collection cycle
- Garbage Collections - Selected Allocation Failure report, which displays information about a single garbage-collection cycle that the JVM hosting the application server performed. For example, this report displays the free heap space both before and after garbage collection, the heap space freed, and the number of objects moved during garbage collection. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.

Accessing the Garbage Collections - Selected Allocation Failure workspace

To access this workspace from the Allocation Failures workspace, use one of the following procedures:

- From the Allocation Failures report, right-click the link icon and from the menu, click **Garbage Collections - Selected Allocation Failure**.
- From the Allocation Failure Elapsed Times bar chart or the Heap Usage - History bar chart, right-click any bar; then, from the menu, click **Link To** > **Garbage Collections - Selected Allocation Failure**.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collection Analysis workspace

This workspace summarizes all the Java Virtual Machine (JVM) garbage-collector activity over a user-defined interval.

The JVM generates detailed garbage collection logs for an application server when started with the **`verbose:gc`** runtime parameter. This workspace displays data provided by the Garbage Collection Analysis attributes.

The predefined workspace contains the following items:

- Garbage Collection Rate - History graph, which displays the rate at which the garbage-collection algorithm is being invoked
- Heap Usage - History bar chart, which displays the high-water mark of free storage (in kilobytes) available in the heap after each garbage-collector run

- Percentage of Time Garbage Collector Running - History graph, which displays the percentage of real time the garbage collector was running during the current interval, for each server region
- Garbage Collection Analysis report, which displays information about the garbage-collection activities within the Java Virtual Machine that is hosting the application server. For example, this report displays the number of times the collector ran during the interval and the resulting number of objects that the collector freed

Accessing the Garbage Collection Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the name of the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Garbage Collection Analysis** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the Garbage Collection Analysis workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal navigate to the **Garbage Collection Analysis** workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **Garbage Collection Analysis** window, right-click the **choose link** icon  then click **Diagnostic Memory Leak**.
2. If this is the first time you access the managing server visualization agent, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you must log in only once per session using your **User Name** and **Password**.
4. Click **OK** to display the **Memory Leak Confirmation report** page in the managing server visualization engine. You can use this page to diagnose memory leak problems.
5. In a z/OS environment, right-click the link icon on a row where the ASID column displays Summary.
   a. Click **Garbage Collection Analysis Servant Regions**.

b. In the **Garbage Collection Analysis - Servant Regions** table, right-click the link icon on a row.
c. Click **Diagnostic Memory Leak**.

6. For information about Creating a Memory Leak Confirmation report in this page, and additional features refer to the Composite Application Manager help within the managing server visualization engine Interface.
7. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271.
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

High Availability Manager workspace

The High Availability Manager workspace provides aggregated information about high availability managers.

This workspace displays data provided by the High Availability Manager attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set the Resource Data Collection Method configuration value to **On Demand**) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Local Group - History graph, which shows the number of local groups.
- Group State Rebuild Time - History graph, which shows the time taken to rebuild the global group state in milliseconds.
- High Availability Manager report, which displays information about the high availability manager, including group state rebuild time, bulletin-board subjects, bulletin-board subscriptions, bulletin-board rebuild time, and local bulletin-board subjects

Accessing the High Availability Manager workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.

2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **High Availability Manager** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

IMAP/POP workspace

This workspace provides aggregated statistics of the usage information about the IMAP service and the POP3 service connectivity, especially for the performance-related connectivity.

This workspace displays data provided by the Workplace Mail IMAP/POP attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

The predefined workspace contains the following items:

- Active Sessions bar chart, which displays the number of active sessions during the sampling interval
- Authentication Failures bar chart, which displays the number of authentications failures during the sampling interval
- Workplace Mail report, which displays detailed information about the workplace mail for each protocol

Accessing the IMAP/POP workspace

To access this workspace from the Workplace Mail workspace, use one of the following procedures:

- Within the Navigator, right-click the **Workplace Mail** entry; then from the menu, click **Workspace** > **IMAP/POP**.

- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **IMAP/POP**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

J2C Connection Pools workspace

This workspace reports information about resource adapters and connectors that adhere to J2C, the WebSphere application server implementation of the JEE Connector Architecture (JCA).

Data counters for this category contain usage information about the J2C connection pools that enable enterprise beans to connect to and interact with systems such as the Customer Information Control System (CICS) and the Information Management System (IMS). This workspace displays data provided by the J2C Connection Pools attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Highest Average Pool Sizes bar chart, which shows the largest average number of managed connections for each J2C connection pool. The Y-axis headings correspond to the row number of the J2C Connection Pools report
- Worst Wait Times bar chart, which shows the worst wait time (in milliseconds) for each of the J2C connection pools. The y-axis headings correspond to the row number of the J2C Connection Pools report
- Highest Allocation Rates bar chart, which displays the highest managed-connection creation, destruction, and allocation rates (in events per second)
- J2C Connection Pools report, which displays information about connectors that adhere to J2C. For example, this report displays the number of managed connections or physical connections and the total number of connections or connection handles

Accessing the J2C Connection Pools workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.

3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **J2C Connection Pools** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the J2C Connection Pools workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In The Tivoli Enterprise Portal access the J2C connection pools workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements see "Prerequisites for access" on page 272.

1. In the **J2C connection pools** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display**.
2. If this is your first time to access the managing server visualization agent during a session, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need to log in only once per session using your **User Name** and **Password**.
4. Click **OK** to display the **Server Activity Display** page in the managing server Visualization engine. You can use link to diagnose application problems, for example, slow transactions or high response times.
5. The results relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Server Activity Display** page and additional features, refer to the Composite Application Manager help within the managing server visualization agent Interface.
6. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

JMS Summary workspace

The JMS Summary workspace displays information about queues being used by your applications using the Java Message Service (JMS) interface.

The JMS Summary workspace also provides information about how WebSphere application server applications are using WebSphere MQ. It displays such information as the number of messages read and written and which queue managers and queues were used during the interval.

This workspace displays data provided by the JMS Summary attributes.

The predefined workspace contains the following items:

- Worst JMS Send Times bar chart, which displays the longest times (in milliseconds) your application spent putting messages onto a queue during the interval
- Worst JMS Receive Times bar chart, which displays the longest times (in milliseconds) your application spent getting messages from a queue during the interval
- Worst JMS Browse Times bar chart, which displays the longest times (in milliseconds) your application spent browsing messages on a queue during the interval
- JMS Summary - Current Interval report, which displays detailed information about how the WebSphere application server uses messaging middleware (that is, WebSphere MQ) using JMS. Details include the send, receive, browse, and publish times for your application. It also includes such information as, which queue managers and queues are being used and how many messages are being read and written

Accessing the JMS Summary workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **JMS Summary** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the JMS Summary workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In The Tivoli Enterprise Portal access the **JMS Summary** workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **JMS Summary - Current interval** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display**.
2. If this is your first time to access the managing server visualization agent during a session, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need to log only once per session using your **User Name** and **Password**.
4. Click **OK** to display the **Server Activity Display** page in the managing server Visualization engine. You can use this link to diagnose application problems, for example, slow transactions or high response times.
5. In a z/OS environment, right-click the link icon on a row where the ASID column displays Summary.
   a. Click **Selected JMS - Servant Regions**.
   b. In the **Selected JMS - Servant Regions** table, right-click the link icon on a row.
   c. Click **Diagnostic Server Activity Display**.
6. For more information about the options available in a z/OS environment, refer to "Region workspaces in a z/OS environment" on page 346 and "Accessing a Region workspace" on page 348.
7. The results in this page relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Server Activity Display** page, and additional features refer to the Composite Application Manager help within the managing server visualization agent Interface.
8. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271.
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

JVM Stack Trend workspace

This workspace displays trend data regarding JVM CPU usage, JVM garbage collection, and JVM heap usage.

This workspace displays data provided by the application server and Garbage Collection Analysis attribute groups.

The predefined workspace contains:

- JVM CPU Trend chart indicates the percentage of the JVM CPU used.
- Percent GC Time Used chart

- Heap Usage Trend chart

Accessing the Selected Application - Configuration workspace

Access this workspace using one of the following methods:

- Double click the JVM icon in Resources workspace.
- Right-click the JVM icon in the Resources workspace and select **Link To**, then select **JVM Stack Trend**.

For additional information, see:

- Resources and Applications workspaces
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Log Analysis workspace

This workspace reports application server error and exception conditions as recorded in the application server log file.

This workspace displays data provided by both the Log Analysis attributes and DC Messages attributes.

The predefined workspace contains the following items:

- DC Message Events, which display aggregated information about the messages from WebSphere data collector
- Log Analysis report, which displays application server error and exception conditions as recorded in the application server log file, SystemOut.log. This information includes the exception severity as well as the ID and text of the associated message. In the Log Analysis report, if the PID value is displayed as `-1`, it indicates that the data collector is disconnected. If a WebSphere server shutdown occurs, the connection between the data collector and Tivoli Enterprise Monitoring Agent is closed but the data collector continues to write to log files. Tivoli Enterprise Monitoring Agent processes these records but sets the PID value to `-1`.

Accessing the Log Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Log Analysis** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Lotus Workplace Server workspace

The Lotus Workplace Server workspace displays overall statistics for each Lotus Workplace application server being monitored by the Tivoli Enterprise Monitoring Agent.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the WebSphere application server heap over time. The chart flyovers display the exact values

  This view displays data provided by the Garbage Collection Analysis attributes.
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes.
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes.
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the application server attributes.
- Application server Summary report, which displays overall information about this WebSphere application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the application server attributes.

Accessing the Lotus Workplace Server workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, click the **Lotus Workplace Server** entry of your choice.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Messages Queues workspace

This workspace provides aggregated statistics about the usage information about the message delivery on Lotus Workplace servers.

This workspace displays data provided by the Workplace Mail Queues attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

The predefined workspace contains the following items:

- Queue Messages bar chart, which displays the number of messages in the ready, retry, unprocessed, and dead state in the queue during the sampling interval
- Workplace Mail Message Queues report, which displays detailed information about the state of messages in each queue

Accessing the Messages Queues workspace

To access this workspace from the Workplace Mail workspace, use one of the following procedures:

- Within the Navigator, right-click the **Workplace Mail** entry; then from the menu, click **Workspace** > **Messages Queues**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Messages Queues**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Messaging Engine Communications workspace

This workspace provides aggregated counter statistics for all the messaging engines being hosted by the current application server. WebSphere application server 5.1 based products do not support this workspace.

This workspace displays data provided by the Messaging Engine Communications attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must

first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Situation Event Console report, which shows the event activity for situations associated with the current Navigator item and any items within the branch. The Navigator alerts you when a situation becomes true by overlaying the Navigator item with an event indicator. This report is useful when multiple alerts are raised and you might not know newly arrived alerts just by looking at the indicator.
- Messaging Engine Communications report, which displays information about the messaging engine communications, including API connections, errors, reads, writes, message written and message read.

Accessing the Messaging Engine Communications workspace

To access this workspace from the Platform Messaging workspace, use one of the following procedures:

- Within the Navigator, right-click the **Platform Messaging** entry; then, from the menu, click **Workspace** > **Messaging Engine Communications**.
- From the primary Tivoli Enterprise Portal menu, click **View** >**Workspace** > **Messaging Engine Communications**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Messaging workspace

In this workspace, you can view of performance counters of the Messaging Engines supported by a server. WebSphere application server 5.1 based products do not support this workspace.

This workspace displays data provided by the Messaging Engines attributes.

Note:

- This workspace reports zeros for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Average Local Wait Time - History graph, which shows the historical time spent by messages on this durable subscription on consumption
- Expired Messages - History graph, which shows the number of report-enabled messages that expired while on this queue

- Incomplete Topic Publications - History graph, which shows the number of publications not yet received by all historical subscribers.
- Total Published - History graph, which shows the historical number of publications to the messaging engines.
- Messaging Engines report, which displays the aggregated information about each messaging engine. A messaging engine is a server component that provides the core messaging functionality of a service integration bus. A messaging engine manages bus resources and provides a connection point for applications

Accessing the Messaging workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Platform Messaging** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

OS Stack workspace

This workspace displays information about the operating system performance.

This workspace displays data provided by the application server attribute group.

The predefined workspace contains:

- Current OS Stack Summary report which displays:
  - Server name
  - Platform CPU Used (ms)
  - System Paging

  Important: The feature Platform CPU Used (ms) does not apply the z/OS platform.
- CPU Used chart
- System Paging chart

Accessing the Selected Application - Configuration workspace

Access this workspace using one of the following methods:

- Double click the OS icon in Resources workspace.
- Right-click the OS icon in the Resources workspace and select **Link To**, then select **OS Stack Trend**.

For additional information, see:

- Resources and Applications workspaces
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Pool Analysis workspace

This workspace displays information about the usage of several types of pools associated with each application server, including web container pools, ORB pools, J2C connection pools, and database connection pools. This workspace helps you detect resource constraints and potential performance congestion.

Important:

- This workspace reports zeros for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Web Container Pool Usage - History graph, which shows the configured maximum number of web container pooled threads and the average active threads in the web container pool

  This view displays data provided by the Thread Pools attributes.
- ORB Pool Usage - History graph, which shows the configured maximum number of ORB pooled threads and the average active threads in the ORB pool

  This view displays data provided by the Thread Pools attributes.
- Web Container Pool % at Max - History bar chart, which shows the maximum usage percentage for the web container pooled threads over time

  This view displays data provided by the Thread Pools attributes.
- ORB Pool % at Max - History bar chart, which shows the maximum usage percentage for the ORB pooled threads over time

  This view displays data provided by the Thread Pools attributes.
- Percent CPU Used - History graph, which shows the percentage of the CPU used over time

  This view displays data provided by the application server attributes.
- DB Connection % at Max - Current Interval bar chart, which shows the maximum usage percentage for a database connection pool over time

  This view displays data provided by the DB Connection Pools attributes.
- J2C Connection % at Max - Current Interval bar chart, which shows the maximum usage percentage for a J2C connection pool over time

  This view displays data provided by the J2C Connection Pools attributes.

Accessing the Pool Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Pool Analysis** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Portal Pages Summary workspace

This workspace reports performances statistics about WebSphere Portal page response times completed on the interval. A historical version of this workspace provides a long-term view of a single portal page that you select.

The following workspace displays only if the request monitoring control level, monitoring level is set to Level2. For more information about the request monitoring control level, see "Application Dashboard (Basic) tab" on page 269.

This workspace displays data provided by the Portal Page Summary attributes.

The predefined workspace contains the following items:

- Worst Response Times bar chart, which displays the worst average response times (in milliseconds) for portlet during the current interval
- Most Popular Portal Pages bar chart, which shows the number of requests for portlet
- Portal Pages report, which displays aggregated information about portal pages, including average response time and request count

Accessing the Portal Pages Summary workspace

To access this workspace from the Portal Summary workspace, use one of the following procedures:

- Within the Navigator, right-click the **Portal Summary** entry; then, from the menu, click **Workspace** > **Portal Pages Summary**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Portal Pages Summary**.

Selected Portal Page - History workspace

The Selected Portal Page - History workspace displays the historical information that corresponds to the information in the Portal Pages Summary workspace for a selected portal page. Historical information is collected over a specific time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected Portal Page - History workspace

To access this workspace from the Portal Pages Summary workspace, use one of the following procedures:

- From the Portal Pages report, right-click the link icon and from the menu, click **Selected Portal Page - History**.
- From the Worst Response Times bar chart, or the Most Popular Portal Pages bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Portal Page - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Portal Summary workspace

The workspace reports summarize statistics about response times and functional decomposition of additional requests collected for WebSphere Portal applications. These include statistics about Portal Pages/Gateway Servlet aggregated response times collected on the interval. Others find grained statistics about Portlet, Model Building, Page Loading, Authentication, and Authorization requests response times collected on the same interval. By default, this workspace is configured for long-term historical interval reporting.

The following workspace displays only if the request monitoring control level, monitoring level is set to Level2. For more information about the request monitoring control level, see "Application Dashboard (Basic) tab" on page 269.

This workspace displays data provided by the Portal Summary attributes.

The predefined workspace contains the following items:

- Portal Pages/Gateway Servlet - History graph, which shows the historical average response time (in milliseconds) of portal pages/Gateway Servlet
- Portlet - History graph, which shows the historical average response time (in milliseconds) of portlets
- Model Building - History graph, which shows the historical response time (in milliseconds) of model building
- Page Loading - History graph, which shows the historical response time (in milliseconds) of page loading
- Authentication - History graph, which shows the historical response time (in milliseconds) of authentication
- Authorization - History graph, which shows the historical response time (in Milliseconds) of authorization

Accessing the Portal Summary workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.

4. Within the list of available agents, expand the WebSphere Portal server of your choice.
5. Within that server list of available WebSphere Portal Server workspaces, click the **Portal Summary** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Portlet Summary and Selected Portlet - History workspaces

These workspaces reports performances statistics about WebSphere Portal portlet response times completed on the interval.

Portlet Summary workspace

The Portlet Summary workspace displays only if the request monitoring control level, monitoring level is set to Level2. For more information about the request monitoring control level, see "Application Dashboard (Basic) tab" on page 269.

A historical version of this workspace provides a long-term view of a single portlet that you select. This workspace displays data provided by the Portlet Summary attributes.

The predefined workspace contains the following items:

- Worst Response Times bar chart, which displays the worst average response times (in milliseconds) for portlet in the current interval
- Most Popular Portlets bar chart, which shows the exception and request rates (in events per second) for portlet
- Portlets report, which displays aggregated information about portlets, including average response time, request count and request rate

Accessing the Portlet Summary workspace

To access this workspace from the Portal Summary workspace, use one of the following procedures:

- Within the Navigator, right-click the **Portal Summary** entry; then, from the menu, click **Workspace** > **Portlet Summary**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Portlet Summary**.

Selected Portlet - History workspace

The Selected Portlet - History workspace displays the historical information that corresponds to the information in the Portal Summary workspace for a selected portlet. Historical information is collected over a particular measured time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected Portlet - History workspace

To access this workspace from the Portlet Summary workspace, use one of the following procedures:

- From the Portlets report, right-click the link icon and from the menu, click **Selected Portlet - History**.
- From the Worst Response Time bar chart, or the Most Popular Portlets bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Portlet - History**.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Request Analysis workspace

The workspace reports response times and functional decomposition information about requests (including servlets, JSPs, and EJB methods) that completed during the interval.

A historical version of this workspace provides a long-term view of a single request that you select. This workspace displays data provided by the Request Analysis attributes.

The predefined workspace contains the following items:

- Worst Average Response Times bar chart, which displays the five worst response times for requests processed during the current interval
- Worst Completion Rates bar chart, which displays the 11 requests that have the worst completion rates
- Requests - Current Interval report, which displays detailed information about the response times recorded for each request

Accessing the Request Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Request Analysis** entry.

Selected Request - History workspace

The Selected Request - History workspace displays the historical information that corresponds to the information in the Request Analysis workspace for a single request type that you select. Historical information is collected over a specific time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected Request - History workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon and from the menu, click **Selected Request - History**.
- From the Worst Average Response Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Request - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

The following workspaces are displayed only if the request monitoring control level, monitoring level is set to Level2. For more information about the request monitoring control level, see "Application Dashboard (Basic) tab" on page 269.

- Selected Request - Datasource
- Selected Request - JMS Queues
- Selected Request - Resource Adapters
- Selected Request - Service Components

Accessing the managing server visualization engine from the Request Analysis workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal access the request analysis workspace, see "Accessing the Request Analysis workspace" on page 317. You can use you can use the following links to access the managing server visualization engine.

- **Diagnostic Recent Completed Requests**
- **Diagnostic In-flight Request Search**
- **Diagnostic SMF Data** (This option is only available if the Tivoli Enterprise Portal is connected to a z/OS data controller.)

For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **Requests - Current Interval** window, right-click the **choose link** icon  then click one of the following options:
   - **Diagnostic Recent Completed Requests**
   - **Diagnostic In-flight Request Search**
2. To view Diagnostic Recent Completed Requests or Diagnostic In-flight Request Search in a z/OS environment, right click the link icon on a row where the ASID column displays Summary.
   a. Click **Selected Request Servant Regions** .
   b. In the **Selected Requests - Servant Regions Current Interval** table, right-click the link icon on a row.
   c. Click **Diagnostic Recent Completed Requests** or **Diagnostic In-flight Request Search**.
3. To view **Diagnostic SMF Data** (z/OS data collector only) right-click the link icon on a row where the ASID column displays Summary.
   a. Click **Selected Request Servant Regions**.
   b. In the **Selected Requests - Servant Regions Current Interval** table, right-click the link icon on a row.
   c. Click **Diagnostic SMF**.

For information about the options available in a z/OS environment, refer to "Region workspaces in a z/OS environment" on page 346 and "Accessing a Region workspace" on page 348

4. If this is your first time to access the managing server visualization agent during the session, you see a **Welcome to the Application Monitor** page.
5. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need to log in only once per session using your **User Name** and **Password**.
6. Click **OK** to display the relevant page in the managing server visualization engine.
   - If you click **Diagnostic Recent Completed Requests** you see the **Server Activity Display** page.
   - If you click **Diagnostic In-flight Request Search** you see the **In-Flight Request Search** page.
   - If you click **Diagnostic SMF Data** you see the **SMF Overview** page.
7. The information in these pages relates directly to the context from where you launched the link in the Tivoli Enterprise Portal.
8. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271.
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

Request Baseline workspace

This workspace displays aggregated information about the request baseline.

The baselining collects statistical information about an application requests completion times and uses this information to assign fair and bad thresholds on the application requests. The product divides the whole request response times into buckets and collects individual hits into each bucket. Use these attributes to get statistics from individual requests collected during baselining interval.

This workspace displays data provided by the Baseline attributes.

The predefined workspace contains:

- Baseline Data report, which shows lower and upper boundaries for each bucket request as well as the breakdown of nested request types in percentage.
- Request Label report, which shows the monitoring configuration settings for selected requests, including auto-threshold settings and actual thresholds calculated from the baseline data.
- Nested Delays Distribution bar chart, which displays a bar for each bucket of response times across the different nested types (JDBC, JCA, JMS, and so on). This chart provides you with additional hints and insight about how to interpret response times distribution displayed in the distribution chart.

- Response Time Distribution bar chart, which displays the distribution of the servlet response times on the baselining interval, also called zones.

You can use the bar charts to customize automatic request time thresholds. See "Enable_Auto_Threshold: set threshold parameters" on page 500.

Accessing the Request Baseline workspace

Complete the following steps to access this workspace: :

1. Access the "Application Configuration workspace" on page 321.
2. In the Application Requests table, right-click the link icon and click **Request Baseline**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

EJB Tier Analysis workspace

The workspace displays detailed information about application tier health for a selected WebSphere application.

The application tier health is derived from the following performance statistics:

- Calculated application request delays in EJB container or custom requests delays compared against corresponding thresholds assigned in application configuration.
- Completion rates for application edge EJB requests.
- Application server ORB thread pool utilization level.
- PMI statistics for application EJB container transactions begin, commit, and rollback rates.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Application Tier Analysis report, which shows the overall health status of the Application tier for a selected application. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.
- Container Pool Usage bar chart, which displays the average number of concurrently active threads and the average number of free threads in the pool. This view displays data provided by the Thread Pools attributes.
- Worst Application Tier Delays bar chart, which displays the top 10 delayed requests in the application tier. This view displays data provided by the Request Analysis attributes.
- Worst Application Tier Completion Rates bar chart, which displays the top 10 worst requests in the application tier. This view displays data provided by the Request Analysis attributes.
- Container Transactions - Rates bar chart, which displays the counts of global and local transactions that were started, committed, and undone during the interval. This view displays data provided by the Container Transactions attributes.
- JVM Health - CPU Used % graph, which displays the percentage of the CPU used by the Java Virtual Machine (JVM) during the interval. This view displays data provided by the application server attributes.

- JVM Health - Heap Used % graph, which displays the current heap usage for the monitored JVM. This view displays data provided by the Garbage Collection Analysis attributes.
- JVM Health - GC Active Time % graph, which displays the percentage of real time that the garbage collector was active during the interval. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the EJB Tier Analysis workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon and, from the menu, click **Selected Application - EJB Tier Analysis**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Configuration workspace

This workspace displays the information about the configuration details of the selected application.

The workspace contains information about application requests and the corresponding thresholds assigned to them, as well as about status and configuration for application baselining activity. An entry is created for each application in the configuration report when a WebSphere application is discovered by the monitoring agent. The data is also stored in a context file local to monitoring agent where it can persist between monitoring agent restarts.

This workspace displays data provided by the Application Monitoring Configuration attributes.

The predefined workspace contains:

- Longest Request Thresholds - Top 10 bar chart, which displays the 10 longest (in time) request thresholds configured for the given application (Servlet/JSP URL or EJB class/method call).
- Application Requests report, which shows the discovered application requests and thresholds assigned to them. Click the link in the Application Request Configuration report or right-click and select Selected Request - Baseline to go to the Request Baseline Workspace. The link to the Selected Request - Baseline is disabled when the baseline request count is less than or equal to 0.
- Application report, which shows the common details about application configuration, including custom requests monitoring levels for application and current baselining status.

Accessing the Application Configuration workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon and from the menu, click **Application Configuration**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Backend Tier Analysis workspace

This workspace displays information about the details of the backend tier for a selected application.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Backend Tier Analysis report, which shows the overall health status of the backend tier for a selected application. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.
- Worst Backend Tier Delays bar chart, which displays the top 10 delayed requests in the backend tier. This view displays data provided by the Request Analysis attributes.
- Most Used JDBC resources bar chart, which displays the average time per request used by queries and updates to the data source. This view displays data provided by the Datasources attributes.
- Most Used JMS Resources bar chart, which displays the longest times your application spent in getting messages from a queue, putting messages onto a queue, publishing messages to a queue, or browsing messages on a queue during the interval. This view displays data provided by the JMS Summary attributes.
- Most Used JDBC Pools bar chart, which displays the average percentage of the connection pool in use during the interval. This view displays data provided by the DB Connection Pools attributes.
- Most Used JCA Pools bar chart, which displays the average percent of the pool that is in use for the interval. This view displays data provided by the J2C Connection Pools attributes.
- JVM Health - CPU Used % graph, which displays the percentage of the CPU used by the Java Virtual Machine (JVM) during the interval. This view displays data provided by the application server attributes.
- JVM Health - Heap Used % graph, which displays the current heap usage for the monitored JVM. This view displays data provided by the Garbage Collection Analysis attributes.
- JVM Health - GC Active Time % graph, which displays the percentage of real time that the garbage collector was active during the interval. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Backend Tier Analysis workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon and from the pop-up menu, click **Backend Tier Analysis**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Health History workspace

The workspace displays the information about the historical health status of a selected application. By default, the history data is collected for the last 24 hours.

The workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Availability/Throughput - History graph, which displays average request processing rate by application over the time. This view displays data provided by the Request Times and Rates attributes.
- Availability/Completion Rate - History graph, which displays the average request completion rate by application over the time; Completion rate is defined as ratio of successfully completed requests count to the total count of requests processed by application on the interval. This view displays data provided by the Request Times and Rates attributes.
- Availability/Average Load- History graph, which displays the average number of concurrent application requests over the time. This view displays data provided by the Request Times and Rates attributes.
- Response Time - History graph, which displays the average application response time over the time. This view displays data provided by the Request Times and Rates attributes.
- Server Resources/CPU Used - History graph, which displays the percent of CPU time used by the application JVM process over the time. This view displays data provided by the application server attributes.
- Server Resources/Paging Rate - History graph, which displays the system paging rate in kilobytes per second over the time. This view displays data provided by the application server attributes.
- Server Resources/GC Active Time - History graph, which displays the percentage of total CPU time for which the garbage collector was active over the time. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Application Health History workspace

Complete the following steps to access this workspace from the Application Health Summary:

1. Click **Application Health Summary report**.
2. Right-click the link icon to display the pop-up menu.
3. Click **Application Health history**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Tier Analysis workspace

This workspace displays detailed information about the client tier health for a selected WebSphere application.

The client tier health indicator is derived from the following performance statistics:

- Calculated application request delays inside Servlet/JSP or Portal container compared against corresponding thresholds assigned in application configuration.

- Completion rates for edge Servlet/JSP and Portal application requests.
- Application server WebContainer thread pool utilization level.
- PMI statistics for HTTP session counts by application.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Web Tier Analysis report, which shows the overall health status of application execution in web or portal containers. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.
- Web Contain bar chart, which displays the current utilization level of the web Container thread pool. This view displays data provided by the Thread Pools attributes.
- HTTP Sessions - Average bar chart, which displays the average number of concurrently active and live HTTP session numbers for the selected application during the interval. This view displays data provided by the Servlet Sessions attributes.
- Worst Web Tier Delays bar chart, which displays the top 10 requests with biggest delays (threshold violations) in the client tier. This view displays data provided by the Request Analysis attributes.
- Worst Web Tier Completion Rates bar chart, which displays the top 10 Servlet/JSP/Portal edge requests with the worst completion rates. This view displays data provided by the Request Analysis attributes.
- JVM Health - CPU Used % graph, which displays the percentage of the host CPU used by the Java Virtual Machine (JVM) during the interval. This view displays data provided by the application server attributes.
- JVM Health - Heap Used % graph, which displays the current heap usage for the monitored JVM. This view displays data provided by the Garbage Collection Analysis attributes.
- JVM Health - GC Time % graph, which displays the percentage of real time that the garbage collector was active during the interval. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Web Tier Analysis workspace

Complete the following steps to access this workspace from the Application Health Summary:

1. Click **Application Health Summary report**.
2. Right-click the link icon and from the menu click **Web Tier Analysis**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Datasources - Datasource Trend workspace

The Datasources Trend workspace displays information about datasource connections.

This workspace displays data provided by the Datasources attributes.

The predefined workspace contains the following items:

- Current Datasources report displays datasource name, total wait time, connection rate, connection average wait time and connection max wait time.
- Total Wait Time Trend line chart displays the total time that applications had to wait for a connection to the data source.
- Connection Rate Trend bar chart displays the number of connection requests (per second) created for the data source.
- Connection Average and Max Time Trend line chart displays the average time (in milliseconds) and the worst-case time (in milliseconds) that applications had to wait for a connection.

Selected Request - Datasources workspace

The Selected Request - Datasources workspace displays information about JDBC activity performed by the request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes.

The predefined workspace contains the following items:

- Worst Datasources Response Times bar chart, which shows the worst response times (in milliseconds) for data sources accessed by this request
- Selected Request - Datasources report, which displays detailed information about the data sources accessed for the selected request. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.

Accessing the Selected Request - Datasources workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon and from the menu, click **Selected Request - Datasources**.
- From Worst Average Response Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Request - Datasources**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - JMS Queues workspace

The Selected Request - JMS Queues workspace displays information about message queues owned by messaging middleware and accessed by the request that you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes.

The predefined workspace contains the following items:

- Worst JMS Queues Response Times bar chart, which shows the worst response times (in milliseconds) for JMS resources accessed by this request
- Selected Request - JMS Queues report, which displays detailed information about the JMS resources accessed by the selected request. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.

Accessing the Selected Request - JMS Queues workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon and from the menu, click **Selected Request - JMS Queues**.
- From Worst Average Response Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Request - JMS Queues**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - Portal Processing workspace

The Selected Request - Portal Processing workspace displays information about portlet, or portal page response times, or both, referenced by the request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes.

Note: This workspace appears under all WebSphere application servers, but it has data only under the WebSphere Portal Server.

The predefined workspace contains the following items:

- Worst Portal Processing Response Times bar chart, which shows the worst response times (in milliseconds) for portal sources accessed by this request
- Selected Request - Portal Processing report, which displays detailed information about the portal sources accessed for the selected request. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.

Accessing the Selected Request - Portal Processing workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link and from the menu, click **Selected Request - Portal Processing**.
- From Worst Average Response Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Request - Portal Processing**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - Resource Adapters workspace

The Selected Request - Resource Adapters workspace displays response-time information about the J2C resources adapters referenced by the request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes.

The predefined workspace contains the following items:

- Worst Average Response Times bar chart, which shows the worst-performing J2C resource adapter nested requests, in milliseconds
- Selected Request - Resource Adapters report, which displays detailed information about each J2C resource adapter that was accessed by the selected request. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.

Accessing the Selected Request - Resource Adapters workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon and from the menu, click **Selected Request - Resource Adapters**.
- From Worst Average Response Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Request - Resource Adapters**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - Service Components workspace

The Selected Request - Service Components workspace displays information about Service Component Architecture (SCA) activity performed by the request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes.

The predefined workspace contains the following items:

- Worst Service Components Response Times bar chart, which shows the worst response times (in milliseconds) for data sources accessed by this request
- Selected Request - Service Components report, which displays detailed information about the data sources accessed for the selected request. For Tivoli Enterprise Monitoring Agent running on z/OS, region ID can be found in the ASID column.

Accessing the Selected Request - Service Components workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon and from the menu, click **Selected Request - Service Components**.
- From Worst Average Response Times bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Request - Service Components**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Service Component Elements workspace

This workspace lists performance metrics for all the service components and their elements. Service components contain one or more elements, which are sets of

different steps processed in each service component. In turn, each element has its own set of event natures, which are key points that are reached when processing a service component element.

This workspace displays data provided by the Service Component Elements attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Worst Service Times bar chart, which shows the numbers of the Average Response Time in milliseconds
- Most Invocations bar chart, which shows the numbers of the invocations per second
- Most Failures bar chart, which shows the numbers of the failed invocation counts
- Service Component Elements report, which shows aggregated data about the average response time, failed count, success count, error rate, and request rate.

Accessing the Service Component Elements workspace

To access this workspace from the Service Components workspace, use one of the following procedures:

- Within the Navigator, right-click the **Service Components** entry; then, from the menu, click **Workspace** > **Service Component Elements**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Service Component Elements**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - History workspace

The Selected Request - History workspace displays information about historical request instances for the named request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes.

The predefined workspace contains the following items:

- Selected Request Response Times - History bar chart, which shows the average response times (in milliseconds) spent within various request types in instances of this named request.
- Selected Request - History sources report, which displays detailed information about historical request instances for the selected request.

Click the  icon to select a time span. The item displays the data for the selected time span.

Accessing the Selected Request - data sources workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon and select **Selected Request - History**.
- From Worst Average Response Times bar chart, right-click any bar and select **Link To > Selected Request - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Service Components workspace

This workspace provides overview performance of the key service components. WebSphere servers feature their own service components, and each of these components has its own set of event points that can be monitored.

This workspace displays data provided by both the Service Components attributes. This workspace is available only for WebSphere Process Server and WebSphere ESB Server.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Service Component Architecture - History graph, which shows historical bad request count, the instrumentation level, set instrumentation level type, and summary, when the component name of the service is Service Component Architecture
- Business Rules - History graph, which shows historical bad request count, the instrumentation level, set instrumentation level type, and summary, when the name is Business Rules

- Map - History graph, which shows historical bad request count, the instrumentation level, set instrumentation level type, and summary, when the name is Map
- Mediation - History graph, which shows historical bad request count, the instrumentation level, set instrumentation level type, and summary, when the name is Mediation
- Business State Machine - History graph, which shows historical bad request count, the instrumentation level, set instrumentation level type, and summary, when the name is Business State Machine
- Selector - History graph, which shows historical bad request count, the instrumentation level, set instrumentation level type, and summary, when the name is Selector
- Bad Requests report, which displays a summary of the bad request counts

Accessing the Service Components workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Service Components** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Servlets/JSPs - Selected Web Application workspace

This workspace displays statistical data regarding the servlets and JSPs invoked by a single Web application.

This workspace displays data provided by the Servlets JSPs attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Worst Servlet/JSP Response Times bar chart, which displays the worst average response times (in milliseconds) for servlets and JSP invoked by the selected web application
- Most Popular Servlet/JSP bar chart, which shows the servlet and JSP exception and request rates (in events per second) for the selected application
- Worst Servlet/JSP Error Rates bar chart, which shows the worst servlet and JSP error rates for the selected application during the interval
- Servlets/JSPs - Selected Web Application report, which displays performance information about the servlets and JSPs invoked by the application. For example, this report displays the average number of concurrent requests for a servlet and the amount of time it takes a servlet to respond to a request

Accessing the Servlets/JSPs - Selected Web Application workspace

To access this workspace from the Web Applications workspace, use one of the following procedures:

- From the Web Applications report, right-click the link icon and from the menu, click **Servlets/JSPs - Selected Web Application**.
- From the Worst Response Times bar chart, the Most Popular Web Applications bar chart, or the Worst Error Rates bar chart, right-click any bar; then, from the menu, click **Link To** > **Servlets/JSPs - Selected Enterprise Application**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Scheduler workspace

The Scheduler workspace contains data for the Scheduler service. The scheduler service schedules and tracks the starting and stopping of applications.

This workspace displays data provided by the Scheduler attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Highest Task Failures bar chart, which shows the number of the task failure count. The y-axis headings correspond to the row number of the Scheduler report
- Highest Completed Tasks bar chart, which shows the number of the task finished count. The y-axis headings correspond to the row number of the Scheduler report

- Schedulers report, which displays information about the scheduler data, including task finish count, task failure count, task expiration rate, task finish rate, and task run rate

Accessing the Scheduler workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Scheduler** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Sessions workspace

This workspace displays information about servlet sessions.

A session is a series of requests to a servlet, originating from the same user at the same browser. Applications running in a web container can use these sessions to track individual users. This workspace displays data provided by the Servlet Sessions attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Most Active Sessions bar chart, which shows the most frequently referenced servlet sessions for each listed web application
- Largest Total Session Object Sizes bar chart, which shows the total session object sizes (in bytes) and the associated web applications
- Servlet Sessions report, which shows usage data about the servlet sessions, including the rates as which sessions are created and destroyed and their read and write times

Accessing the Sessions workspace

To access this workspace from the Web Applications workspace, use one of the following procedures:

- Within the Navigator, right-click the **Web Applications** entry; then, from the menu, click **Workspace** > **Servlet Sessions**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **Servlet Sessions**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Thread Pools workspace

This workspace reports information about the various thread pools that support the applications running in your Java Virtual Machine (JVM).

This workspace displays data provided by the Thread Pools attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Highest Average Pool Sizes bar chart, which shows the largest thread pools in the JVM
- Average Thread Pool Usage bar chart, which shows the average active and free threads for each thread pool
- Thread Pools report, which shows information about the usage statistics for thread pools that belong to a WebSphere application server, such as average and maximum pool sizes and creation and destruction rates

Accessing the Thread Pools workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.

5. Within that server list of available WebSphere application server workspaces, click the **Thread Pools** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the Thread Pools workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In The Tivoli Enterprise Portal access the Thread Pools workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements see "Prerequisites for access" on page 272.

1. In the **Thread Pools** window, right-click the **choose link** icon  then click **Diagnostic JVM Thread Display**.
2. If this is your first time to access the managing server visualization agent during a session, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need only to log in using your **User Name** and **Password** once per session.
4. Click **OK** to display the **Diagnostic JVM Thread Display** page in the managing server Visualization engine. You can use link to diagnose application problems, for example, slow transactions or high response times.
5. The results relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Diagnostic JVM Thread Display** page and additional features, refer to the Composite Application Manager help within the managing server visualization engine Interface.
6. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

Thread Pool Trend workspace

The Thread Pool Trend workspace displays information about thread pool size and usage.

This workspace displays data provided by the Selected Request attributes.

This predefined workspace contains the following items:

- Current Thread Pool report displays thread pool name, average active threads, average pool size, percentage of time pool at max and average free threads.
- Average Pool Size Trend bar chart displays the average number of threads in the pool.
- Average Thread Pool Usage Trend bar chart displays the average percentage of time that all threads were in use during the sampling interval.
- Percent of Time Pool Size at Max Trend bar chart displays the percentage of time the pool size is running at the maximum value.

Accessing the Threadpool Trend workspace

Select the **Thread Pools** workspace, in the **Thread Pools** report, click the link icon, and select **Thread Pool Trend**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Applications workspace

This workspace displays information about the web applications running in JEE application servers.

This workspace displays data provided by the Web Applications attributes.

Note:

- The **Web Applications** workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- Worse Response Times bar chart, which shows the worst servlet response times (in milliseconds) during the interval
- Most Popular Web Applications bar chart, which shows the servlet exception and request rates (in events per second)
- Worse Error Rates bar chart, which shows the worst servlet error rates during the interval
- Web Applications report, which displays aggregated performance data for each web application, about all servlets and JSPs deployed to that web application, including response and error rates, and response times.

Accessing the Web Applications workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Web Applications** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the Web Applications workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal access the **WebSphere Agent** workspace. You can use the **Diagnostic Server Activity Display** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **Web Applications** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display**.
2. If this is your first time to access the managing server visualization agent during a session, you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need to log in only once per session using your **User Name** and **Password**.
4. Click **OK** to display the **Server Activity Display** page in the managing server Visualization engine. You can use link to diagnose application problems, for example, slow transactions or high response times.
5. The results in this page relate directly to the context from where you launched the link in the Tivoli Enterprise Portal. For more information about using the **Server Activity Display** page and additional features, refer to the Composite Application Manager help within the managing server visualization agent Interface.
6. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271

- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

Web Services workspace

The Web Services workspace displays information about the data counters of the web services.

The examples of the information include the number of loaded web services, the number of requests delivered and processed, the request response time, and the average size of requests. This workspace displays data provided by both the Web Services attributes and the Web Services Gateway attributes.

Note:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains:

- Worst Response Times bar chart, which displays the worst average response times (in milliseconds) for the selected web services
- Most Popular bar chart, which shows the exception and request rates (in events per second) for the selected services
- Web Services report, which displays aggregated performance data for each web service, including requests, response times, and payload sizes
- Web Service Gateway report, which displays aggregated performance data for each Web service gateway, including the number of synchronous and asynchronous requests and responses

Accessing the Web Services workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Web Services** entry.

Selected Web Services - History workspace

The Selected Web Services - History workspace displays the historical information that corresponds to the information in the Web Services workspace for a selected web service. Historical information is collected over a particular measured time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected Web Services - History workspace

To access this workspace from the Web Services workspace, use one of the following procedures:

- From the Web Services report, right-click the link icon and from the menu, click **Selected Web Services - History**.
- From the Worst Response Times bar chart, or the Most Popular bar chart, right-click any bar; then, from the menu, click **Link To** > **Selected Web Services - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WebSphere Agent workspace

This workspace displays product events that affect the ability of the WebSphere application server agent to collect data.

This workspace displays events occurring within the WebSphere application server agent and WebSphere application servers that are installed on the host computer. It also displays the status of the Tivoli Enterprise Monitoring Agent. The predefined workspace contains the following items:

- Agent Events report, which displays information about agent-level events that affect the ability of the Tivoli Enterprise Monitoring Agent to collect data for the WebSphere application server. You can use this view to see exception and error messages, their IDs, and their severity.

  The Agent Events report also shows the result of issuing a Take Action command. Place your cursor over a truncated message to display the text of the complete message

  This report displays data reported by the WebSphere Agent Events attributes.
- Application Servers Summary report displays information about status of the WebSphere server.

  This report displays data reported by the application server Status attributes.

Accessing the WebSphere Agent workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.

3. Within that node list of available Tivoli Enterprise Monitoring Agents, if the **WebSphere Agent** entry has a name next to it (for example, **WebSphere Agent - Primary**), click the entry. Otherwise, expand the entry and click one of the agent names.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Accessing the managing server visualization engine from the WebSphere Agent workspace

Important: To use this functionality, you need ITCAM for Application Diagnostics Managing Server.

In the Tivoli Enterprise Portal access the WebSphere agent workspace. You can use the **Diagnostic Server Activity Display** or the **Diagnostic In-flight Request Search** link to access the managing server visualization engine. For information about access requirements, see "Prerequisites for access" on page 272.

1. In the **Applications Server Summary** window, right-click the **choose link** icon  then click **Diagnostic Server Activity Display** or **Diagnostic In-flight Request Search**.
2. If this is your first time to access the managing server visualization agent, during the session you see a **Welcome to the Application Monitor** page.
3. Depending on the user setup configuration in your environment, you might have to type your managing server visualization engine **User Name** and **Password**. If you do, you need to log in only once per session using your **User Name** and **Password**.
4. Click **OK** to display the relevant page in the managing server Visualization engine.
   - If you click **Diagnostic In-flight Request Search** you see the **In-Flight Request Search** page in managing server visualization engine.
   - If you click **Diagnostic Server Activity Display** you see the **Server Activity Display** page in the managing server visualization engine.
5. The information in both of these pages relates directly to the context from where you launched the link in the Tivoli Enterprise Portal.
6. To return to the previous workspace in the Tivoli Enterprise Portal interface at any time choose from the following options:
   - From the Tivoli Enterprise Portal desktop client, including WebStart, click the back arrow on the Portal toolbar.
   - From the Tivoli Enterprise Portal browser client, click the browser back arrow.

For additional information, see:

- "Workspace link to managing server visualization engine" on page 271
- "Access the managing server visualization engine from the Tivoli Enterprise Portal" on page 275

WebSphere application server workspace

The WebSphere application server workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilobytes) within the WebSphere application server heap over time. The chart flyovers display the exact values

  This view displays data provided by the Garbage Collection Analysis attributes.
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes.
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes.
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server used over time

  This view displays data provided by the application server attributes.
- Application server Summary report, which displays overall information about this WebSphere application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the application server attributes. In the application server summary report, each row represents a different region. When you right-click the link for a row, you can choose to go to Selected Region - application server Summary, Selected Region - Request Analysis, Selected Region - Application Health Status, Selected Region - Datasources, Selected Region - Log Analysis or Selected Region - JMS Summary. All these links are disabled when Tivoli Enterprise Monitoring Agent is running on Distributed data collector and enabled when Tivoli Enterprise Monitoring Agent is running on z/OS data collector.

Accessing the WebSphere application server workspace

Complete the following steps to access this workspace:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, click the **WebSphere App Server** entry of your choice.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WebSphere ESB Server workspace

The WebSphere ESB Server workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the WebSphere application server heap over time. The chart flyovers display the exact values

This view displays data provided by the Garbage Collection Analysis attributes.

- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes.
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes.
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the application server attributes.
- Application server summary report, which displays overall information about this WebSphere application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the application server attributes.

Accessing the WebSphere ESB Server workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, click the **WebSphere ESB Server** entry of your choice.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WebSphere Portal Server workspace

The WebSphere Portal Server workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the WebSphere application server heap over time. The chart flyovers display the exact values

  This view displays data provided by the Garbage Collection Analysis attributes.
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes.
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes.
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the application server attributes.

- Application server summary report, which displays overall information about this WebSphere application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the application server attributes. In the application server summary report, each row represents a different region. When you right-click the link for a row, you can choose to go to Selected Region - application server Summary, Selected Region - Request Analysis, Selected Region - Application Health Status, Selected Region - Datasources, Selected Region - Log Analysis, Selected Region - JMS Summary, Selected Region - Portal Summary, Selected Region - Portlet Summary. All these links are disabled when Tivoli Enterprise Monitoring Agent is running on Distributed data collector and enabled when Tivoli Enterprise Monitoring Agent is running on z/OS data collector.

Accessing the WebSphere Portal Server workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, click the **WebSphere Portal Server** entry of your choice.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WMQ Client Link Communications workspace

This workspace provides aggregated counter statistics for all of the clients of WMQ Queue Managers that are or have been connected to this application server. WebSphere application server 5.1 based products do not support this workspace.

This workspace displays data provided by the WMQ Client Link Communications attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- WMQ Client Link Communication Errors - History graph, which shows the number of errors that have caused connections to WMQ JMS clients to be dropped
- WMQ Client Link Statistics report, which displays information about the messaging engine communications, including batch sent, message sent, message received, comm errors, writes blocked, and reads blocked

Accessing the WMQ Client Link Communications workspace

To access this workspace from the Platform Messaging workspace, use one of the following procedures:

- Within the Navigator, right-click the **Platform Messaging** entry; then, from the menu, click **Workspace** > **WMQ Client Link Communications**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **WMQ Client Link Communications**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WMQ Link Communications workspace

This workspace provides aggregated counter statistics for all of the WMQ Queue Managers that are or have been connected to this application server. WebSphere application server 5.1 based products do not support this workspace.

This workspace displays data provided by the WMQ Link Communications attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- WMQ Link Communication Errors - History graph, which shows the historical number of communication errors that resulted in the disconnection of a network connection to a WMQ Queue Manager
- WMQ Link Statistics report, which displays information about the messaging engine communications, including batches sent, batches received, message sent, message received, and comm errors

Accessing the WMQ Link Communications workspace

To access this workspace from the Platform Messaging workspace, use one of the following procedures:

- Within the Navigator, right-click the **Platform Messaging** entry; then, from the menu, click **Workspace** > **WMQ Link Communications**.
- From the primary Tivoli Enterprise Portal menu, click **View** > **Workspace** > **WMQ Link Communications**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workload Management workspace

This workspace displays information about the Workload Management (WLM) server and about the WLM client that initiates workload requests to that server.

Workload management optimizes the distribution of client processing tasks. Incoming work requests are distributed to the application servers, enterprise beans, servlets, and other objects that can most effectively process the requests. Workload management also provides failover protection when servers are not available, improving application availability. In a WebSphere application server environment, you implement workload management using clusters, transports, and replication domains.

This workspace displays data provided by both the Workload Management Server attributes and the Workload Management Client attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

The predefined workspace contains the following items:

- WLM Server Incoming Requests bar chart, which shows the incoming strong affinity requests and the incoming nonaffinity requests (per second)
- WLM Client Outgoing Requests bar chart, which shows the outgoing requests (per second)
- Workload Management Server report, which shows detailed information about the WLM server, such as incoming requests and clients served
- Workload Management Client report, which shows information about the clients that initiate workload requests, such as outgoing requests and response times

Accessing the Workload Management workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.

2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.
4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Workload Management** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workplace Mail workspace

This workspace provides aggregated statistics of the usage information about the incoming message traffic.

This workspace displays data provided by the Workplace Mail Service attributes.

Important:

- This workspace reports blanks for resource data on the first invocation if PMI data collection is configured for on-demand sampling (that is, if your site set configuration value Resource Data Collection Method to On Demand) or if you have not yet run applications that generate PMI resource data. To report resource data in this workspace after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- Because of high processor usage, the Tivoli Enterprise Monitoring Agent provides on-demand sampling by default. To activate PMI monitoring, you must first select this workspace and then select it again later. Each time you select the workspace, it displays the data collected during the interval between selections.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

The predefined workspace contains the following items:

- Connections bar chart, which displays the number of connections to the SMTP server, SMTP client connections, and the maximum number of concurrent LDAP connections during the sampling interval
- Workplace Mail report, which displays detailed information about the workplace mail connections

Accessing the Workplace Mail workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to the node that you want to select.
3. Within that node list of monitored applications, expand the list of WebSphere agents.

4. Within the list of available agents, expand the WebSphere application server of your choice.
5. Within that server list of available WebSphere application server workspaces, click the **Workplace Mail** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Region workspaces in a z/OS environment

For z/OS installations, workspace tables report data at both a region and server instance level.

Important: ITCAM Agent for WebSphere Applications on z/OS is available in separate product packages.

Note: ITCAM Agent for WebSphere Applications 7.2 is not available on z/OS The advantage is that you can view metrics collected at both levels and switch between server instance level and region level. The following table lists the workspaces that show information at both a region and server level.

All workspaces except the Garbage Collection Analysis workspace report data on both levels. The Garbage collection analysis workspace does not have links to the top-level workspaces.

Table 18. Workspaces and the Servant and Selected Regions in a z/OS environment

Workspace Table Name	Link Name	Description
WebSphere App Server - application server Summary	WebSphere App Server - application server Summary - Selected Region - application server Summary Selected Region - Application Health Status Selected Region - Request Analysis Selected Region - Log Analysis Selected Region - Data Sources Selected Region - JMS - Summary	This workspace displays data at regional level. Click the Selected Region links to access region-specific links to other top-level workspaces. When a workspace is linked from the application server summary you can view specific drill-down metrics. To view a report for an individual region, see “Accessing a Region workspace” on page 348

Table 18. Workspaces and the Servant and Selected Regions in a z/OS environment (continued)

Workspace Table Name	Link Name	Description
Application Health - Application Health Summary	Selected Application - Servant regions Selected Application - Health History Selected Application- Web Tier Analysis Selected Application - EJB Tier Analysis Selected Application - Backend Tier Analysis Selected Application - Request Analysis Selected Application - Configuration	In an z/OS environment, the Application Health Summary report displays the total results for the server instances. To view report results by region, click the WebSphere App Server - application server Summary table and right-click a link icon in the table to view the available options.
Request Analysis - Requests Current Interval	Selected Request - Datasource Selected Request - JMS Queues Selected Request - Resource Adaptors Selected Request - History Selected Request - Servant Regions	In an z/OS environment, when you select Request Analysis - Requests Current Interval, this report displays the total results for the server instances. To view a report for an individual Request Analysis region, see "Accessing a Region workspace" on page 348.
Garbage Collection Analysis - Garbage Collection Analysis	Selected Region - History Garbage Collection Analysis - Servant Region (only available when you click a **[Summary]** row)	For Garbage Collection Analysis, there is a summary report of all regions and there are also reports by individual region.
Log Analysis - Log Analysis	Selected Region - Log Analysis	Log Analysis workspace. In a z/OS environment, the log analysis workspace reports data in two ways. When you select the Log analysis workspace the report displays JVM Log Analysis and DC message events from all regions. To view a report for an individual log analysis region, see "Accessing a Region workspace" on page 348.
Data sources - Data sources - Current Interval	Selected Datasource Selected Datasource - History	In an z/OS environment, this report displays the total results for the server instances. To view a report for an individual Data source region, see "Accessing a Region workspace" on page 348.

Table 18. Workspaces and the Servant and Selected Regions in a z/OS environment (continued)

Workspace Table Name	Link Name	Description
JMS Summary - JMS Summary - Current Interval JMS	Selected JMS - Servant Regions	In an z/OS environment, this report displays the total results for the server instances. To view a report for an individual JMS Summary region, see“Accessing a Region workspace.”
WebSphere Portal Server	Selected Region - Portal Server Summary	In an z/OS environment, this report displays the total results for the server instances. To view a report for an individual Portal Server summary, see “Accessing a Region workspace”
Portal Summary	Selected Region - Portal Summary Selected Region - Portlet Summary Selected Portal Page - History Selected Portlet - History Selected Region - Portal Page Summary	In an z/OS environment report displays the total results for the server instances. To view a report for an individual region see “Accessing a Region workspace”

Accessing a Region workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand z/OS system, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored agents, expand the list of the servers.
4. In the list of available servers, click the WebSphere agent of your choice.
5. In the list of available servers, select the Server of your choice.
6. Right-click the selected Server node and select **WebSphere App Server** workspace.
7. Right-click a link icon in the **application server Summary** table to display workspaces connected with the current region.
8. Select the workspace of your choice from the following list:
   - Selected Region application server Summary
   - Selected Region - Application Health Status
   - Selected Region - Request Analysis
   - Selected Region - Log Analysis
   - Selected Region - Data Sources
   - Selected Region - JMS - Summary

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Attributes for ITCAM Agent for WebSphere Applications

IBM Tivoli Composite Application Manager Agent for WebSphere Applications is a Tivoli Enterprise Monitoring Agent that resides within your distributed systems. This agent gathers data about running WebSphere application server processes that have been collected and stored by the ITCAM Data Collector for WebSphere, and stores those data in elements called attributes. Each attribute is a characteristic of an object. For example, the Receive Count attribute in the JMS Summary attribute group counts the number of messages your applications have retrieved from JMS messages queues.

Attribute groups

The IBM Tivoli Composite Application Manager Agent for WebSphere Applications attributes are organized into groups of related items. These attribute groups comprise the attribute tables for this agent. For example, the Garbage Collection Analysis attribute group provides information about the frequency with which the Java Virtual Machine (JVM) invokes its garbage collector.

Attributes and workspaces

Within the Tivoli Enterprise Portal workspaces, these attributes are displayed in, and correspond to, the columns in the reports and the items in the graphic displays for charts and graphs. You can use the collected data to analyze and monitor the performance of your WebSphere application servers and the applications running within them. For an overview of the correlations between the predefined workspaces and the attribute groups, see Attribute Groups Used by the Predefined Workspaces.

Attributes and situations

Various attributes are referenced by the predefined situations of the product. You can also use the IBM Tivoli Composite Application Manager Agent for WebSphere Applications attributes to create your own situations to monitor the performance of your WebSphere application servers and their applications. These situations can monitor your WebSphere application server resources or correlate multiple conditions to alert you to problems that might have occurred when attribute values exceed defined thresholds.

Attribute groups used by the predefined workspaces

A workspace contains graphical data or report columns that correspond directly to particular attributes in an attribute group. The following table shows the correlations between the predefined workspaces and the attribute groups. The primary and secondary workspaces, are listed alphabetically, not in the order in which they appear in the Navigator.

Table 19. Workspaces and the attribute groups they reference

Workspace	Related Attribute Groups
Application Health Summary	Application Health Status
Application Registry	Application Monitoring Configuration
Allocation Failures	Allocation Failure

Table 19. Workspaces and the attribute groups they reference (continued)

Workspace	Related Attribute Groups
Cache Analysis	Dynamic Cache Dynamic Cache Templates
Client Communications	Client Communications
Container Object Pools	Container Object Pools
Container Transactions	Container Transactions
Datasources Selected Datasource - History	Datasources
DB Connection Pools Selected DB Connection Pool - History	DB Connection Pools
DCS Stacks	DCS Stack Counter
Destinations	Topic Spaces Queue
Durable Subscriptions	Durable Subscriptions
EJB Containers	EJB Containers
Enterprise Java Beans	Enterprise Java Beans
Garbage Collections - Selected Allocation Failure	Garbage Collection Cycle
Garbage Collector Analysis	Garbage Collection Analysis
High Availability Manager	High Availability Manager
IMAP/POP	Workplace Mail IMAP/POP
J2C Connection Pools	J2C Connection Pools
JMS Summary	JMS Summary
Log Analysis	Log Analysis
Lotus	application server Request Times and Rates Garbage Collection Analysis
Messages Queues	Workplace Mail Queues
Messaging Engine Communications	Messaging Engine Communications
Platform Messaging	Messaging Engines
Pool Analysis	Thread Pools DB Connection Pools J2C Connection Pools application server
Portal Pages Summary Selected Portal Page - History	Portal Page Summary
Portal Summary	Portal Summary
Portlet Summary Selected Portlet - History	Portlet Summary
Request Analysis Selected Request - History	Request Analysis
Selected Request - Baseline	Baseline
“Selected Request - Service Components workspace” on page 327	Selected Request
Scheduler	Scheduler

Table 19. Workspaces and the attribute groups they reference (continued)

Workspace	Related Attribute Groups
Selected Application - Application Tier Analysis Selected Application - Backend Tier Analysis Selected Application - Health History Selected Application - Client Tier Analysis	Application Health Status
Selected Application - Configuration	Application Monitoring Configuration
Selected Request - Datasources	Selected Request
Selected Request - JMS Queues	Selected Request
Selected Request - Resource Adapters	Selected Request
Servlets/JSPs - Selected Web Application	Servlets JSPs
Sessions	Servlet Sessions
Thread Pools	Thread Pools
Web Applications	Web Applications
Web Services Selected Web Services - History	Web Services Counters Web Services Gateway Counters
WebSphere Agent	WebSphere Agent Events application server Status "Remote Configuration Requests attributes" on page 434
WebSphere App Server	application server Request Times and Rates Garbage Collection Analysis
WebSphere ESB Server	application server Request Times and Rates Garbage Collection Analysis
WebSphere Portal Server	application server Request Times and Rates Garbage Collection Analysis
WebSphere Process Server	application server Request Times and Rates Garbage Collection Analysis
WMQ Client Link Communications	WMQ Client Link Communications
WMQ Link Communications	WMQ Link Communications
Workload Management	Workload Management Client Workload Management Server
Workplace Mail	Workplace Mail Service

Alarm Manager attributes

The **Alarm Manager** attributes provide information for the alarm management. Use these attributes to manage alarms fired by the application for each work manager.

The attributes within this group are used to build the Alarm Manager workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Alarms Canceled The number of alarms canceled by the application. The valid format is a positive integer.

Alarms Created The total number of alarms created by all asynchronous scopes for the current Work Manager. The valid format is a positive integer.

Alarms Fired The number of alarms fired. The valid format is a decimal (formatted to three decimal places).

Alarms Latency Duration The latency of alarms fired in milliseconds. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Alarms Pending Size The number of alarms waiting to fire. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Alarms Rate The number of alarms firing per second. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Instrumentation Level For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 20. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Work Manager Name The name of the work manager. The value format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Allocation Failure attributes

The **Allocation Failure** attribute group provides information about the heap-allocation failure that caused the Java Virtual Machine hosting the application server to start its garbage-collection routine.

You can use the Allocation Failure attributes in situations to determine the events that caused the JVM to start garbage collection. The attributes within this group are used to build the Allocation Failures workspace.

Allocation Failure Number The identifier assigned to the current allocation-failure block, which is associated with a bar in the Heap Usage - History bar chart. The valid format is a positive integer.

ASID The identifier (decimal) assigned to the address space running this servant region.

Bytes Needed The number of bytes needed on the heap when this allocation failure occurred. The valid format is a positive integer.

GC Cycle Count The number of garbage-collection cycles caused by this allocation failure. The valid format is a positive integer.

Heap Expanded The total number of kilobytes by which the heap expanded or contracted as a result of garbage collection. The valid format is a positive integer.

Heap Free (%) after GC The percentage of heap space that is available after garbage collection. The valid format is a positive integer.

Heap Status Whether the out-of-heap-space alert has been raised. Valid values are Normal and Out_of_heap_space.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Kbytes Free at Start of GC The number of kilobytes available in the heap before garbage collection began in response to this allocation failure. The valid format is a positive integer.

Kbytes Freed by GC The number of kilobytes freed by the garbage collector for this allocation failure. The valid format is a positive integer.

Kbytes Used The number of kilobytes in the heap that were in use when this allocation failure occurred. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Objects Moved The number of objects the garbage collector moved during compaction. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 21. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Time since Last Failure (ms) The time (in milliseconds) since the previous allocation failure. The valid format is a positive integer.

Time to Complete (ms) The time (in milliseconds) required to complete the action that resulted from this allocation failure. The valid format is a positive integer.

Total Kbytes Freed by GC The total number of kilobytes freed by the garbage collector in response to this allocation failure. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Health Status attributes

The **Application Health Status** attributes provide information for real-time and historical application health status.

The attributes within this group are used to build the Application Health Summary workspace.

Application Health The combined application health level. Valid values are Unknown, Good, Fair, and Bad.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer or -1 which means that Application ID is an aggregated statistic for all applications.

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Application Status The status of the monitored application. Valid values are Standby, Discovered, Unknown, Starting, Running, Stopping, Stopped, and Failed.

Application Tier Health The health level of the application tier. Valid values are Unknown, Good, Fair, and Bad. Application tier health indicator is determined from EJB or custom request delays collected on the interval and compared against thresholds configured for application requests.

ASID The identifier (decimal) assigned to the address space running this servant region.

Backend Tier Health The health level of the backend tier. Backend tier health indicator is determined from JDBC, JCA, JNDI, JMS delays collected on the interval and compared against thresholds configured for application requests. Valid values are Unknown, Good, Fair, and Bad.

Client Tier Health The health level of the client tier. Valid values are Unknown, Good, Fair, and Bad. Client tier health indicator is determined from servlet/JSP or portal delays collected on the interval and compared against thresholds configured for application requests.

Completion Level The completion level of the requests during the interval. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from request data as the percentage of number of failed requests to the total number of application requests on the interval.

Custom Requests The availability indicator of the custom requests. Valid values are Unknown, Good, Fair, and Bad.

EJB Container The health level of the EJB container. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from EJB delay types collected during the interval and compared against application thresholds.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA The overall health status of JEE Connector Architecture (JCA) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JCA delay types collected during the interval and compared against application thresholds.

JDBC The overall health status of Java DataBase Connectivity (JDBC) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JDBC delay types collected during the interval and compared against application thresholds.

JNDI The overall health status of Java Naming and Directory Interface (JNDI) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JNDI delay types collected during the interval and compared against application thresholds.

JMS The overall health status of Java Message Service (JMS) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JMS delay types collected during the interval and compared against application thresholds.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Portal Container The health level of the portal container. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from portal delay types collected during the interval and compared against application thresholds.

Response Level The health level of the response time for the requests. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from application requests response times collected during the interval and compared against application thresholds.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 22. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Web Container The health level of the web container. Valid values are Unknown, Good, Fair, and Bad.

SCA The health level of application SCA access. Valid values are Unknown, Good, Fair, and Bad.

WAS Node Name The name of the WebSphere Application Server node group to which the application server belongs. The valid format is an alphanumeric string, with a maximum of 64 characters.

WAS Cell Name The name of the WebSphere Application Server cell to which the application server belongs. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process ID The process ID for the application server. The valid format is a 4-byte integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Monitoring Configuration attributes

The **Application Monitoring Configuration** attributes provide information for the Application Monitoring Configuration.

Use these attributes to monitor different WebSphere applications running within an application server. The attributes within this group are used to build the Selected Application - Configuration workspace.

Application Alias The alias name that you can optionally assign for the application. In practice, this attribute enables you to combine multiple applications under the same common alias and report their request in the Tivoli Enterprise Portal as it would come from same application. This attribute is blank by default. You can assign the value to it from Take Actions at any time in the application monitoring life cycle. The valid format is an alphanumeric string, with a maximum of 256 characters.

App ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Application Name The name of the application to which the request belongs. You can define the pattern of this name in the Application Registry workspace. The valid format is an alphanumeric string, with a maximum of 256 characters.

Bad Completion Rate (%) The bad completion rate threshold for the requests. The valid format is an alphanumeric string, with a maximum of 256 characters.

Baselining Elapsed Time The number of seconds during which the application baselining has been running. The valid format is a positive integer.

Baselining Status The status of the application baselining process. Valid values are Idle, Running, and Standby.

Baselining Scheduled Stop Time The date and time baselining is scheduled to finish. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 23. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Baselining Start Time The date and time when the application baselining was started. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 24. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day

Table 24. Format of the 12-character timestamp (continued)

Character String	Meaning
YY	Year
HH	Hour
MM	Minute
SS	Second

Baselining Update Interval The number of seconds that defines how often active baselining data is incrementally updated to the monitoring agent. The valid format is a positive integer.

Fair Completion Rate (%) The fair completion rate threshold for the requests. The valid format is an alphanumeric string, with a maximum of 256 characters.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Monitoring Status The current application monitoring status. Valid values are Discovered, Enabled, Disabled, and Standby.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Data Monitoring Level The custom request aggregation level for all application requests. Valid values are Default, Disabled, Level1, and Level2. This attribute is set to Default when the application is first discovered.

Request Data Sampling Rate The custom request aggregation rate for all application requests. The valid format is a positive integer.

Reflex Automation Mode When reflex automation mode is enabled, application monitoring level is automatically updated on WASAppHealth* situation event.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 25. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application server status attributes

The **application server Status** attributes provide status information for all WebSphere application servers (and the WebSphere administrative server) being monitored by the agent.

The attributes within this group are used to build the WebSphere Agent workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Cluster Name The name of the server group (cluster) that this application server belongs to. The valid format is an alphanumeric string, with a maximum of 128 characters.

Cluster Type Indicates the type of the server group (cluster) the application server belongs to.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process identifier of the Java virtual machine. The valid format is a positive integer.

Regions Number The number of z/OS regions connected. This applies to z/OS environments only.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 26. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Mode The mode of the WebSphere application server.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Origin Node Name Indicates the origin node name of the server subnode.The valid format is an alphanumeric string, with a maximum of 128 characters.

Server Subnode Name The name of the server node in the navigation tree. The valid format is an alphanumeric string, with a maximum of 32 characters.

Server Type The type of server process. The valid values are:

Table 27. Types of server processes

Server Type	Definition
Unknown	The server type cannot be determined
AppServer	A process that executes applications
AdminServer	The administrative server you use when configuring WebSphere application server environments
NodeAgent	The WebSphere application server node agent
JMSServer	The WebSphere application server JMS server
DeploymentMgr	The WebSphere application server deployment (cell) manager
ManagedProcess	A stand-alone WebSphere application server process
UnManagedProcess	A WebSphere application server process that is managed by a WebSphere application server deployment manager through a node agent

Start Date and Time The date and time when the WebSphere application server started. The valid format is a timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the WebSphere application server. Valid values are Connected and Disconnected.

WAS Cell Name The name of the WebSphere application server cell to which this application server belongs. The valid format is an alphanumeric string, with a maximum of 64 characters.

WAS Configuration Repository Directory Name The name of the WebSphere application server configuration repository directory, which normally resides in the config subdirectory of the product installation root directory. The valid format is an alphanumeric string, with a maximum of 128 characters.

WAS Node Name The name of the WebSphere application server node group to which this application server belongs. The valid format is an alphanumeric string, with a maximum of 64 characters.

VE Host/port The valid format is an alphanumeric string, with a maximum of 256 characters.

Probe ID The valid format is an alphanumeric string, with a maximum of 128 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application server attributes

The **application server** attributes provide the status and summary data for a specific WebSphere application server instance.

The attributes provide performance data for the WebSphere application server runtime (JVM memory), thread pools, HTTP sessions, and configuration parameters. They also provide some information from other attribute groups to give an overall view of the WebSphere application server. Use the application server attributes in situations to monitor the health and performance of a WebSphere application server.

The attributes within this group are used to build the WebSphere application server and the Pool Analysis Workspace workspaces.

Important:

- The attributes in this attribute group contain zeros for performance data if your site sets the configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes when you have installed and configured the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.

- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; this means the attributes in this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

ASID The identifier (decimal) assigned to the address space running this servant region.

CPU Used (ms) Indicates the number of milliseconds the JVM CPU used during the interval. The valid format is a positive integer.

CPU Used (%) Indicates the percentage of the JVM CPU used during the interval. For UNIX users, this attribute has a meaningful value only if the Tivoli Enterprise Monitoring Agent is running with superuser authority. The valid format is a decimal (formatted to one decimal place).

Platform CPU Used (ms) Indicates the number of milliseconds the host platform (OS) CPU used during the interval. This feature does not apply to the z/OS platform.

Garbage Collection Monitoring The monitoring level for garbage-collection data. Valid values are Disabled and Enabled.

Instrumentation Level The JVM instrumentation level. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Hung Threads Total Total number of detected hung threads. The valid format is a positive integer.

Hung Threads Blocked Number of hung threads detected in the BLOCKED state. The valid format is a positive integer.

Hung Threads Waiting Number of hung threads detected in the WAITING state. The valid format is a positive integer.

Hung Threads Timed Waiting Number of hung threads detected in the TIMED_WAITING state. The valid format is a positive integer.

JVM Memory Free (Kbytes) The JVM free memory size in Kbytes.

JVM Memory Total (Kbytes) The JVM total memory size in Kbytes.

JVM Memory Used (Kbytes) The JVM used memory size in Kbytes.

JVM Memory Free (bytes) The JVM free memory size (in bytes). The valid format is a positive integer.

JVM Memory Total (bytes) The JVM total memory size (in bytes). The valid format is a positive integer.

JVM Memory Used (bytes) The JVM used memory size (in bytes). The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process identifier of the Java virtual machine. The valid format is a positive integer.

Probe ID The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Data Monitoring Level The monitoring level for request data stored by the data collector. Valid values are Disabled, Level1 edge request data, such as servlets and JSPs are displayed), and Level2 (nested request data such as JDBC and JMS requests are also displayed).

Request Data Sampling Rate (%) The percentage of Level1 requests (that is, edge requests) being sampled. The valid format is a positive integer.

Resource Data Monitoring The monitoring level for resource (that is, PMI) data stored by the data collector. Valid values are Disabled and Enabled.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12 character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 28. Format of the 12 character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Instance Name The name of the WebSphere application server. This is a logical grouping of one or more server instances (called a "generic server" or "cluster") any one of which can run an application. The valid format is an alphanumeric string, with a maximum of eight characters.

Server Mode Indicates the mode of the WebSphere application server.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Type The type of the WebSphere application server. Valid values are Unknown, AppServer, AdminServer, NodeAgent, JMSServer, DeploymentManager, ManagedProcess, and UnManagedProcess.

Server Subnode Name Indicates the sub node name of the application server.

Start Date and Time The date and time when the WebSphere application server started. The valid format is a timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the WebSphere application server. Valid values are Connected and Disconnected.

Summary Indicates that this row is a summary row of statistical totals for all rows.

System Paging Rate (Kbytes/sec) The system paging rate in kilobytes per second during the interval. The valid format is a positive integer.

WAS Node Name The name of the WebSphere Application Server node group to which the application server belongs. The valid format is an alphanumeric string, with a maximum of 64 characters.

WAS Cell Name The name of the WebSphere Application Server cell to which the application server belongs. The valid format is an alphanumeric string, with a maximum of 64 characters.

Version The version of WebSphere application server. The valid format is an alphanumeric string, with a maximum of eight characters.

VE Host/port The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Baseline attributes

The **Baseline** attributes provide information for baseline extract data for the given application.

The baselining collects statistical information about an application requests completion times and uses this information to assign fair and bad thresholds on the application requests. The product divides the whole request response times into buckets and collects individual hits into each bucket. Use these attributes to get statistics from individual requests collected during baselining interval.

The attributes within this group are used to build the Selected Request - Baseline workspace.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Bad Hits Percentage The percentage of bucket hits in the metric bad value zone. The valid format is a positive integer.

Bucket Number The bucket number of the baselining data. The valid format is a positive integer.

EJB Container Percent The average percent of time that bucket requests were executed inside EJB container. The valid format is a positive integer.

Fair Hits Percentage The percentage of bucket hits in the metric fair value zone. The valid format is a positive integer.

Good Hits Percentage The percentage of bucket hits in the metric good value zone. The valid format is a positive integer.

Hits Percentage The percentage of hits for the bucket during the baselining. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA Percent The average percent of time that bucket requests spent for JCA access. The valid format is a positive integer.

JDBC Percent The average percent of time that bucket requests spent for JDBC access. The valid format is a positive integer.

JMS Percent The average percent of time that bucket requests spent for JMS access. The valid format is a positive integer.

JNDI Percent The average percent of time that bucket requests spent for JNDI access. The valid format is a positive integer.

Lower Boundary (msec) The lower boundary of bucket response times in milliseconds. The valid format is a positive integer.

Metric ID The metric identifier of the baselining data. The valid format is a positive integer.

Metric Type The metric type of the baselining data. Valid formats are Request, Error, and Resource.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Portal Percent The average percent of time that bucket requests were executed inside portal container. The valid format is a positive integer.

Response Time Mean (msec) The mean time of bucket response times. The valid format is a positive integer.

SCA Percent The average percent of time that bucket requests spent for SCA access. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 29. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Hits Percentage The percentage of bucket hits in the metric selection value zone. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet Container Percent The average percent of time that bucket requests were executed inside the servlet container. The valid format is a positive integer.

Total Hits The total hits number for the bucket during the baselining. The valid format is a positive integer.

Upper Boundary (msec) The upper boundary of bucket response times. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Client Communications attributes

The **Client Communications** attributes display overall statistics about server-side monitoring and a client-side API to retrieve performance data.

The attributes within this group are used to build the Client Communications workspace.

Restriction: Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

API Connections The number of API sessions being used by clients that are currently network connected to this application server. Some of these API connections might be being used by internal system processes on behalf of a client. The valid format is a positive integer.

Buffered Read (bytes) The number of bytes of data that have been received from the network and are held pending further processing. Large values might indicate that the application server is unable to process data fast enough to keep up with the clients attached. The valid format is a positive integer.

Buffered Write (bytes) The number of bytes of data being held pending transmission. Large values might indicate network congestion or clients which are unable to process data fast enough to keep up with the application server. The valid format is a positive integer.

Clients Attached The number of distinct client processes currently network connected to this application server. The valid format is a positive integer.

Errors The communication errors that have occurred and resulted in a network connection to a client being disconnected. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the client communications. For WebSphere 5, the valid values are None, Low, Medium, High, and Maximum; for WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Message Read (bytes) The number of bytes of message data received from client processes over network connections. This does not include data used to negotiate the transmission of messages. The valid format is a positive integer.

Messages Received at JMS 0 Priority (bytes) The number of messages received at JMS priority 0. The valid format is a positive integer.

Messages Received at JMS 1 Priority (bytes) The number of messages received at JMS priority 1. The valid format is a positive integer.

Messages Received at JMS 2 Priority (bytes) The number of messages received at JMS priority 2. The valid format is a positive integer.

Messages Received at JMS 3 Priority (bytes) The number of messages received at JMS priority 3. The valid format is a positive integer.

Messages Received at JMS 4 Priority (bytes) The number of messages received at JMS priority 4. The valid format is a positive integer.

Messages Received at JMS 5 Priority (bytes) The number of messages received at JMS priority 5. The valid format is a positive integer.

Messages Received at JMS 6 Priority (bytes) The number of messages received at JMS priority 6. The valid format is a positive integer.

Messages Received at JMS 7 Priority (bytes) The number of messages received at JMS priority 7. The valid format is a positive integer.

Messages Received at JMS 8 Priority (bytes) The number of messages received at JMS priority 8. The valid format is a positive integer.

Messages Received at JMS 9 Priority (bytes) The number of messages received at JMS priority 9. The valid format is a positive integer.

Messages Sent at JMS 0 Priority (bytes) The number of messages transmitted at JMS priority 0. The valid format is a positive integer.

Messages Sent at JMS 1 Priority (bytes) The number of messages transmitted at JMS priority 1. The valid format is a positive integer.

Messages Sent at JMS 2 Priority (bytes) The number of messages transmitted at JMS priority 2. The valid format is a positive integer.

Messages Sent at JMS 3 Priority (bytes) The number of messages transmitted at JMS priority 3. The valid format is a positive integer.

Messages Sent at JMS 4 Priority (bytes) The number of messages transmitted at JMS priority 4. The valid format is a positive integer.

Messages Sent at JMS 5 Priority (bytes) The number of messages transmitted at JMS priority 5. The valid format is a positive integer.

Messages Sent at JMS 6 Priority (bytes) The number of messages transmitted at JMS priority 6. The valid format is a positive integer.

Messages Sent at JMS 7 Priority (bytes) The number of messages transmitted at JMS priority 7. The valid format is a positive integer.

Messages Sent at JMS 8 Priority (bytes) The number of messages transmitted at JMS priority 8. The valid format is a positive integer.

Messages Sent at JMS 9 Priority (bytes) The number of messages transmitted at JMS priority 9. The valid format is a positive integer.

Message Written (bytes) The number of bytes of message data sent to client processes over network connections. This does not include data used to negotiate the transmission of messages. The valid format is a positive integer.

Messaging Engine Name The name of the message engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Multicast Send Messages The number of messages transmitted using multicast protocols. The valid format is a positive integer.

Multicast Write The number of bytes transmitted using multicast protocols. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Reads The number of read operations used to receive data from client processes through network connections. The valid format is a positive integer.

Reads Blocked The number of read operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with client processes. The valid format is a positive integer.

Received at High Priority (bytes) The number of bytes of data received at a high priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Received at Highest Priority (bytes) The number of bytes of data received at the highest possible priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Received at JMS 0 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 0 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 1 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 1 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 2 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 2 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 3 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 3 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 4 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 4 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 5 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 5 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from

time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 6 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 6 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 7 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 7 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 8 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 8 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at JMS 9 Priority (bytes) The number of bytes of data received at the priority used by JMS priority 9 messages. Typically this is an accurate measure of the number of bytes of message data received at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Received at Low Priority (bytes) The number of bytes of data received at a low priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Received at Lowest Priority (bytes) The number of bytes of data received at the lowest possible priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Received at Very High Priority (bytes) The number of bytes of data received at a high priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Received at Very Low Priority (bytes) The number of bytes of data received at a low priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 30. Format of the 12-character timestamp

Character String	Meaning
MM	Month

Table 30. Format of the 12-character timestamp (continued)

Character String	Meaning
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sent at High Priority (bytes) The number of bytes of data transmitted at a high priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Sent at Highest Priority (bytes) The number of bytes of data transmitted at the highest possible priority for transmission. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Sent at JMS 0 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 0 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 1 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 1 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 2 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 2 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 3 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 3 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 4 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 4 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However,

from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 5 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 5 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 6 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 6 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 7 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 7 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 8 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 8 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at JMS 9 Priority (bytes) The number of bytes of data transmitted at the priority used by JMS priority 9 messages. Typically this is an accurate measure of the number of bytes of message data transmitted at this priority level. However, from time to time, control transmissions used to negotiate the flow of messages might be transmitted at this priority level. The valid format is a positive integer.

Sent at Low Priority (bytes) The number of bytes of data transmitted at a low priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Sent at Lowest Priority (bytes) The number of bytes of data transmitted at the lowest priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Sent at Very High Priority (bytes) The number of bytes of data transmitted at a high priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Sent at Very Low Priority (bytes) The number of bytes of data transmitted at a low priority. Message data cannot be transmitted with this priority, so typically these bytes of data comprise control transmissions used to negotiate the flow of messages. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all messaging engines. Valid values are No and Yes.

Total Read (bytes) The number of bytes of data received from client processes. This includes both message data and data used to negotiate the transmission of messages. The valid format is a positive integer.

Total Written (bytes) The number of bytes of data sent to client processes. This includes both message data and data used to negotiate the transmission of messages. The valid format is a positive integer.

Writes The number of write operations used to transmit data to client processes via network connections. The valid format is a positive integer.

Writes Blocked The number of write operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with client processes. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Container Object Pools attributes

Use the **Container Object Pools** attributes in situations to monitor the effectiveness of the object cache and of resource usage.

These attributes provide aggregated information for each defined EJB container that aggregates bean object pool performance for all Enterprise beans deployed to that container and aggregated information for the application server that aggregates bean object pool performance data for all Enterprise beans deployed to the application server. The attributes within this group are used to build the Container Object Pools workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Average Objects Discarded The average number of objects discarded each time the bean object pool was emptied of idle objects during the interval. The valid format is a decimal (formatted to three decimal places).

Objects Discarded Count The number of objects joined in Average Objects Discarded. The valid format is a 4-byte integer. If the number is too big to be stored, the value -2 is used.

Average Objects in Pool The average number of objects in the bean object pool during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Discard Count The number of times the object returned to the bean-object pool was discarded because the bean object pool was already full during the interval. The valid format is a positive integer.

Discard Rate (per sec) The bean object pool discard rate (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Drain Count The number of times the bean object pool was found idle and an attempt was made to remove idle objects during the interval. The valid format is a positive integer.

Drain Rate (per sec) The number of times (per second) that the bean object pool was found idle during the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Find Count The number of times a retrieval call found an available object in the bean object pool during the interval. The valid format is a positive integer.

Find Rate (per sec) The availability of bean object pool retrievals (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Get Count The number of calls retrieving an object from the bean object pool during the interval. The valid format is a positive integer.

Get Rate (per sec) The number of bean objects retrieved (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Instrumentation Level The instrumentation level for this container. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Container Name The name of the container. The valid format is an alphanumeric string, with a maximum of 64 characters.

Return Count The number of calls returning an object to the bean object pool during the interval. The valid format is a positive integer.

Return Rate (per sec) The bean objects returned (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 31. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of All Containers Whether this row is a summary row of statistical totals aggregated over all bean object pools in the application server. Valid values are No and Yes.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Container Transactions attributes

The **Container Transactions** attribute group provides performance information about transactions that run in each defined EJB container and an aggregated value for all transactions that run in the application server.

Use the Container Transactions attributes in situations to monitor transaction activity for each EJB container and for the application server. The attributes within this group are used to build the Container Transactions workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Global Transaction before Completion Duration (ms) The average duration before completion for global transactions during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transaction before Completion Duration Count The count of times that were averaged for Global Transaction before Completion Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Global Transaction Commit Duration (ms) The time (in milliseconds) that the transaction required for its resolution phase during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transaction Commit Duration Count The count of times that were averaged for Global Transaction Commit Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Global Transaction Commit Rate (per sec) The number of times (per second) global transactions were committed during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transaction Duration (ms) The average duration (in milliseconds) for global transactions during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transaction Duration Count The count of times that were averaged for Global Transaction Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Global Transaction Optimize Rate (per sec) The number of times (per second) that global transactions were converted to single phase since the previous sample. The valid format is a decimal (formatted to three decimal places).

Global Transaction Prepare Duration (ms) The average preparation duration (in milliseconds) for global transactions during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transaction Prepare Duration Count The count of times that were averaged for Global Transaction Prepare Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Global Transaction Rollback Rate (per sec) The number of times (per second) that global transactions were undone because they could not complete during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transaction Timeout Rate (per sec) The number of global transaction timeouts (per second) since the previous sample. The valid format is a decimal (formatted to three decimal places).

Global Transaction Timeouts The number of global transactions that timed out during the interval. The valid format is a positive integer.

Global Transactions Active The number of concurrently active global transactions running in the container during the interval. Global transactions involve multiple resource managers. The valid format is a decimal (formatted to three decimal places).

Global Transactions Begin Rate (per sec) The number of times global transactions were started (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Global Transactions Begun The total number of global transactions that the server began during the interval. The valid format is a positive integer.

Global Transactions Committed The number of global transactions that were completed during the interval. The valid format is a positive integer.

Global Transactions Involve Rate (per sec) The number of times (per second) global transactions were involved during the interval. The valid format is a positive integer.

Global Transactions Involved The number of global transactions that were involved at the server during the interval, including transactions that were begun or imported. The valid format is a positive integer.

Global Transactions Rolled Back The total number of global transactions that were undone because they could not complete during the interval. The valid format is a positive integer.

Global Transactions Optimized The number of global transactions converted to single phase for optimization since the previous sample. The valid format is a positive integer.

Instrumentation Level The instrumentation level for this container. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Local Transaction before Completion Duration (ms) The average duration before completion for local transactions during the interval. The valid format is a decimal (formatted to three decimal places).

Local Transaction before Completion Duration Count The count of times that were averaged for Local Transaction before Completion Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Local Transaction Commit Duration (ms) The average duration for commit for local transactions during the interval. The valid format is a decimal (formatted to three decimal places).

Local Transaction Commit Duration Count The count of times that were averaged for Local Transaction Commit Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Local Transaction Commit Rate (per sec) The number of local transactions (per second) committed during the interval. The valid format is a decimal (formatted to three decimal places).

Local Transaction Duration (ms) The average duration (in milliseconds) of local transactions during the interval. The valid format is a decimal (formatted to three decimal places).

Local Transaction Duration Count The count of times that were averaged for Local Transaction Duration. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Local Transaction Rollback Rate (per sec) The number of times (per second) that local transactions were undone because they could not be completed during the interval. The valid format is a decimal (formatted to three decimal places).

Local Transaction Timeout Rate (per sec) The number of local transactions that timed out per second during the interval. The valid format is a decimal (formatted to three decimal places).

Local Transaction Timeouts The number of local transactions that timed out during the interval. The valid format is a positive integer.

Local Transactions Active The number of concurrently active local transactions running in the container during the interval. Local transactions involve a single resource manager. The valid format is a decimal (formatted to three decimal places).

Local Transactions Begin Rate (per sec) The number of times (per second) local transactions were started during the interval. The valid format is a positive integer.

Local Transactions Begun The number of local transactions begun at the server since the previous sample. The valid format is a positive integer.

Local Transactions Committed The number of local transactions committed during the interval. The valid format is a positive integer.

Local Transactions Rolled Back The number of local transactions that were undone during the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and

SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 32. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Data sources attributes

The **Data sources** attributes provide database usage information.

These attributes provide traffic information such as, response times for database requests, the frequencies at which database connections are created and destroyed, and how often databases are accessed. The attributes within this group are used to build the Datasources workspace.

Important: The attributes within this attribute group contain meaningful values only if your site has set the request data monitoring level to Level2 to collect data on data source requests.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer or -1 which means that Application ID is aggregated statistic for all applications.

ASID The identifier (decimal) assigned to the address space running this servant region.

Average Processing Time (ms) The total average processing time (in milliseconds) that the data source is used by an application. The valid format is a decimal (formatted to three decimal places).

Processing Time Count The count of times that were averaged for Average Processing Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Average Query Processing Time (ms) The average time (in milliseconds) per request used by queries to the data source. The valid format is a decimal (formatted to three decimal places).

Average Update Processing Time (ms) The average time (in milliseconds) per request used by updates to the data source. The valid format is a decimal (formatted to three decimal places).

Connection Average Wait Time (ms) The average time (in milliseconds) that applications had to wait for a connection. The valid format is a decimal (formatted to three decimal places).

Connection Count The number of connections to the data source. The valid format is a positive integer.

Connection Max Wait Time (ms) The worst-case time (in milliseconds) that applications had to wait for a connection. The valid format is a positive integer.

Connection Rate (per sec) The number of connection requests (per second) created for the data source. The valid format is a decimal (formatted to three decimal places).

Connection Total Wait Time (ms) The total time (in milliseconds) that applications had to wait for a connection to the data source. The valid format is a positive integer.

Database Product The name of the database product. The valid format is an alphanumeric string, with a maximum of 128 characters.

Database Product Version The version of the database product. The valid format is an alphanumeric string, with a maximum of 128 characters.

Datasource Name The name of the data source. The valid format is an alphanumeric string, with a maximum of 256 characters.

Datasource Label A shortened version of Datasource Name, used to display the data source name in the chart view. The valid format is an alphanumeric string, with a maximum of 12 characters.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the process running the Java Virtual Machine (JVM). The valid format is a positive integer.

Query Count The number of queries performed against the data source. The valid format is a positive integer.

Query Rate (per sec) The number of queries (per second) being made to the data source. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 33. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of edge requests-such as servlets and JSPs-that were sampled for data source requests during the interval. The valid format is a positive integer.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total Query Processing Time (ms) The total time (in milliseconds) used to process queries made to the data source. The valid format is a positive integer.

Total Update Processing Time (ms) The total time (in milliseconds) used to update the data source. The valid format is a positive integer.

Total Wait Time (ms) The time (in milliseconds) that applications had to wait for connections to the data source. The valid format is a positive integer.

Update Count The number of updates performed against the data source. The valid format is a positive integer.

Update Rate (per sec) The number of updates (per second) made to the data source. The valid format is a decimal (formatted to three decimal places).

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DB Connection Pools attributes

The **DB Connection Pools** attributes provide information about the database connection pool for each defined data source, and an aggregated value that aggregates over all data sources.

Examples of DB Connection Pools include; the number of threads waiting for a connection and the number of connections created and released. Use the DB Connection Pools attributes to analyze JDBC performance for WebSphere application server applications. The attributes within this group are used to build the DB Connection Pools and the Pool Analysis workspaces.

Important:

- The attributes within this attribute group contain zeros for performance data if your site set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Application ID Indicates JEE application ID.

Average Free Pool Size Indicates the average size of the pool based upon the number of free connections.

Average Pool Size The average size of the pool (based upon the number of connections) during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Usage Time (ms) The average time (in milliseconds) a connection was in use; blank if no transactions are completed during the interval. The valid format is a decimal (formatted to three decimal places).

Usage Time Count The count of times that were averaged for Average Usage Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Average Wait Time (ms) The average time (in milliseconds) a client waited for a connection; blank if no transactions are completed during the interval. The valid format is a decimal (formatted to three decimal places).

Wait Time Count The count of times that were averaged for Average Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Average Waiting Threads The average number of threads waiting for a connection during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Connection Allocation Rate (per sec) The connections allocated (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connection Creation Rate (per sec) The connections created (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Connection Destruction Rate (per sec) The connections released (per second) during the interval. The valid format is a positive integer.

Connection Handle Indicates the number of Connection objects in use for a particular connection pool.

Connection Used The number of managed connection objects in use for a particular EIS product name. The valid format is a positive integer.

Connections Allocated The number of connections allocated during the interval. The valid format is a positive integer.

Connections Created The number of connections created during the interval. The valid format is a positive integer.

Connections Destroyed The number of connections released during the interval. The valid format is a positive integer.

Connections Granted The sum of connections allocated and connections created during the interval. The valid format is a positive integer.

Datasource Label The abbreviated name of the data source. The valid format is an alphanumeric string, with a maximum of 32 characters.

Datasource Name The name of the data source. The valid format is an alphanumeric string, with a maximum of 256 characters.

Instrumentation Level The instrumentation level for the database connection pool for the data source. For WebSphere 5, the valid values are None, Low, Medium, High, and Maximum; for WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

JDBC Time (ms) Indicates the amount of time spent running in the JDBC driver which includes time spent in the JDBC driver, network, and database.

JDBC Time Count The count of times that were used for JDBC Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Maximum Pool Size The maximum number of connections that can be created in this connection pool. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Pool Size Indicates the size of the connection pool.

Percent of Time Pool at Max The average percentage of time the number of connections in the pool reached the maximum number during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Percent Used The average percentage of the connection pool in use during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Percent Used Bad The bad percent of pool usage by application. The valid format is a positive integer.

Percent Used Fair The fair percent of pool usage by application. The valid format is a positive integer.

Percent Used Good The good percent of pool usage by application. The valid format is a positive integer.

Prep Statement Cache Discard Rate (per sec) The cache discards (per second) of prepared statements during the interval. The valid format is a decimal (formatted to three decimal places).

Prep Statement Cache Discards The number of prepared statements discarded from the cache during the interval. The valid format is a positive integer.

Pool Size Indicates the size of the connection pool.

Return Count The number of connections that applications returned to the pool during the interval. The valid format is a positive integer.

Return Rate (per sec) The number of connections (per second) returned since the previous sample. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and

SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 34. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of All DB Connections Whether this row is a summary row of statistical totals collected for all the DB connection pools. Valid values are No and Yes.

Thread Timeout Rate (per sec) The number of threads (per second) that timed out during the interval. The valid format is a decimal (formatted to three decimal places).

Threads Timed Out The number of threads that timed out while waiting for a connection during the interval. The valid format is a positive integer.

Total Usage (ms) The total time (in milliseconds) the connection object used. The valid format is a decimal (formatted to three decimal places).

Total Wait (ms) The total time (in milliseconds) the connection object waited. The valid format is a decimal (formatted to three decimal places).

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DC Messages attributes

The **DC Messages** attributes provide message information from WebSphere data collector.

The attributes within this group are used to build the Log Analysis workspace.

ASID The identifier (decimal) assigned to the address space running this servant region.

Component The name of the component that caused the error. The value format is an alphanumeric string, with a maximum of 32 characters.

Event Date and Time The date and time the event occurred. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 35. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

File Name The name of the file. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message Description The description of the message. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message ID The unique identifier of the message. The valid format is an alphanumeric string, with a maximum of eight characters.

Method Name The name of the method. The valid format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID Indicates the process ID of the JVM.

Sequence Number The sequence number in the JMX notifications stream. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Severity The severity of the message. Valid values are Info, Warning, Error, and Severe.

Thread ID The identifier of the thread where the event occurred. The valid format is an alphanumeric string, with a maximum of 16 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DCS Stack attributes

The **DCS Stack** attributes reports information about the statistical data within the entire WebSphere application server domain, including multiple nodes and servers.

Examples of DCS Stack attributes include; the incoming and outgoing message size, the number of incoming and outgoing messages, congestion events, and message buffer reallocations. The attributes within this group are used to build the DCS Stacks workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Average Incoming Message Size The average size (in bytes) of the messages that were received by the DCS stack. The valid format is a positive integer.

Average Incoming Message Size Count The count of individual message sizes that were averaged for Average Incoming Message Size. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Average Outgoing Message Size The average size (in bytes) of the messages that were sent through the DCS stack. The valid format is a positive integer.

Average Outgoing Message Size Count The count of individual message sizes that were averaged for Average Outgoing Message Size. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Coalesce Time The amount of time it actually takes to coalesce a view. The valid format is a decimal (formatted to three decimal places).

Coalesce Time Count The count of times that were used for Coalesce Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

DCS Stack Name The name of the Topic Space. The value format is an alphanumeric string, with a maximum of 256 characters.

Group Size The size of the group the local member belongs to. The valid format is a positive integer.

High Severity Congestion Events The number of times that a high severity congestion event for outgoing messages was raised. The valid format is a positive integer.

Incoming Messages The number of messages received by the DCS stack. The valid format is a decimal (formatted to three decimal places).

Instrumentation Level The instrumentation level for the DCS stack. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Join View Change Time The time to do a merge view change. The DCS stack is blocked during this time. The valid format is a decimal (formatted to three decimal places).

Join View Change Time Count The count of times that were used for Join View Change Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Maximum Incoming Message Size The maximal size (in bytes) of the messages that were received by the DCS stack. The valid format is a positive integer.

Maximum Outgoing Message Size The maximal size (in bytes) of the messages that were sent through the DCS stack. The valid format is a positive integer.

Message Buffer Reallocations The number of message buffer reallocations due to inadequate buffer size. If this number is larger than 20 percent of the number of sent messages, contact IBM Support. The valid format is a positive integer.

Minimum Incoming Message Size The minimal size (in bytes) of the messages that were received by the DCS stack. The valid format is a positive integer.

Minimum Outgoing Message Size The minimal size (in bytes) of the messages that were sent through the DCS stack. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Outgoing Messages The number of messages sent through the DCS stack. The valid format is a positive integer.

Remove View Change Time The time to do a split view change. The DCS stack is blocked during this time. The valid format is a decimal (formatted to 3 decimal places).

Remove View Change Time Count The count of times that were used for Remove View Change Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 36. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Sent Messages The number of messages sent through the DCS stack. The valid format is a positive integer.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Suspicions The number of times that the local member suspected other members. The valid format is a positive integer.

Synchronization Completion Time The amount of time needed to guarantee that all view members are synchronized. The valid format is a decimal (formatted to three decimal places).

Synchronization Completion Time Count The count of times that were used for Synchronization Completion Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Synchronization Timeouts The number of times that the synchronization procedure timed out. The valid format is a positive integer.

View Changes The number of times that this member underwent view changes. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Durable Subscriptions attributes

The **Durable Subscriptions** attributes display overall statistics about the durable subscriptions of a selected topic.

A durable subscription can be used to preserve messages published on a topic while the subscriber is not active. The attributes within this group are used to build the Durable Subscriptions workspace.

Important:

- The attributes in this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Aggregate Message Wait Time The time spent by messages in the bus at consumption. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Aggregate Message Wait Time Count The count of times that were used for Aggregate Message Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Assured Persistent Messages Consumed The number of Assured Persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Available Message The number of messages waiting to be consumed. The valid format is a positive integer.

Best Effort Non-persistent Messages Consumed The number of best effort non-persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Express® Non-persistent Messages Consumed The number of express non-persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the Durable Subscriptions. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Local Oldest Message Age The longest time any message has spent on this subscription. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Local Oldest Message Age Count The count of individual message ages that were used for Local Oldest Message Age. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Local Message Wait Time The time spent by messages on this durable subscription at consumption. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Local Message Wait Time Count The count of times that were used for Local Message Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Messaging Engine Name The name of the message engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Reliable Non-persistent Messages Consumed The number of reliable non-persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Reliable Persistent Messages Consumed The number of Reliable Persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 37. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute

Table 37. Format of the 12-character timestamp (continued)

Character String	Meaning
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Subscription Name The name of the subscriptions. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Topic Space Name The name of the topic space. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Messages Consumed The total number of messages consumed from this durable subscription. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Dynamic Cache attributes

The **Dynamic Cache** attribute group provides information about the dynamic cache.

WebSphere application server consolidates several caching activities, including servlets, web services, and WebSphere commands, into one service called the dynamic cache. These caching activities work together to improve application performance and share many configuration parameters, which are set in an application server dynamic cache service. The dynamic cache works within an application server Java Virtual Machine (JVM), intercepting calls to cacheable objects, for example, through a servlet service method or an execute method of a command. It either stores the object output to, or serves the object content from, the dynamic cache.

The attributes within this group are used to build the Cache Analysis workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.

- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Cache Instance Name Indicates the cache instance name.

Cache Instance Type Indicates Cache instance type.

Current In-Memory Cache Size The number of cache entries currently in memory. The valid format is a positive integer.

In-Memory and Disk Timeout Rate (per sec) The rate (per second) of total in-memory and disk timeouts for the sampling interval. The valid format is a decimal (formatted to three decimal places).

In-Memory and Disk Timeouts The total number of in-memory and disk timeouts during the sampling interval. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the dynamic cache. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Maximum In-Memory Cache Size The maximum number of cache entries in memory. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 38. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of Cache Indicates that this row is a summary row of statistical totals collected for all the cache object types.

PMI Enable The valid format is a 4-byte integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Dynamic Cache Templates attributes

The **Dynamic Cache Templates** attribute group provides information about the cache template data.

A cache template is an object type defined by a cache policy specified in the WebSphere application server cachespec.xml file. A cache policy specifies the cache rules indicating what is cached, the invalidation, timeout conditions, and other data. The attributes within this group are used to build the Cache Analysis workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Cache Instance Name The cache instance name.

Cache Instance Type The cache instance type.

Cache Miss Rate (per sec) The rate (per second) of requests for this cacheable object type that were not found in the cache during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Cache Misses The number of requests for this cacheable object type that were not found in the cache (in memory, on disk or on other cooperating caches). This would have caused the underlying servlet or command to be executed in order to obtain the results. The valid format is a positive integer.

Cache Object Type The name of the object type specified in the cache policy of the cache spec XML file. The valid format is an alphanumeric string, with a maximum of 256 characters.

Client Request Rate (per sec) The request rate (per second) for this cacheable object type made by clients directly accessing this application server. The valid format is a decimal (formatted to three decimal places).

Client Requests The number of requests for this cacheable object type made by clients directly accessing this application server. The valid format is a positive integer.

Cluster Request Rate (per sec) The request rate (per second) for this cacheable object type made by cooperating caches in this cluster. The valid format is a decimal (formatted to three decimal places).

Cluster Requests The number of requests for this cacheable object type made by cooperating caches in this cluster. The valid format is a positive integer.

Current Cache Size The current number of entries for this cacheable object type present in the dynamic cache. The valid format is a positive integer.

Disk Hit Rate (per sec) The rate (per second) of the requests for this cacheable object type served from disk during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Disk Hits The number of requests for this cacheable object type that were served from disk. This applies only when the disk offload is turned on for the dynamic cache. The valid format is a positive integer.

Explicit Disk Invalidations Rate (per sec) The rate at which the entries of this cacheable object type were removed from disk due to explicit invalidations issued by the clients. The valid format is a decimal (formatted to three decimal places).

Explicit Disk Invalidations The number of entries of this cacheable object type that were removed from disk due to explicit invalidations issued by the clients (directly accessing the application server and by remote JVMs in the cluster). The valid format is a positive integer.

Explicit Local Invalidation Rate (per sec) The rate at which the explicit invalidations were received for this cacheable object type from clients accessing the application server directly, either programmatically or by a cache policy. The valid format is a decimal (formatted to three decimal places).

Explicit Local Invalidations The number of explicit invalidations received for this cacheable object type from clients accessing the application server directly, either programmatically or by a cache policy. The valid format is a positive integer.

Explicit Memory Invalidation Rate (per sec) The rate at which the entries of this cacheable object type were removed from memory due to explicit invalidations issued by the clients. The valid format is a decimal (formatted to three decimal places).

Explicit Memory Invalidations The number of entries of this cacheable object type that were removed from memory due to explicit invalidations issued by the clients (directly accessing the application server and by remote JVMs in the cluster). The valid format is a positive integer.

Explicit Remote Invalidation Rate (per sec) The rate at which explicit invalidations were received for this cacheable object type from cooperating JVMs in the cluster. The valid format is a decimal (formatted to three decimal places).

Explicit Remote Invalidations The number of explicit invalidations received for this cacheable object type from cooperating JVMs in the cluster. The valid format is a positive integer.

Instrumentation Level The PMI instrumentation level set for collecting dynamic cache data. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Least Recently Used Invalidation Rate (per sec) The rate (per second) at which entries of this cacheable object type were evicted from memory by a least-recently used algorithm. The valid format is a decimal (formatted to three decimal places).

Least Recently Used Invalidations The number of entries of this cacheable object type that were evicted from memory by a least-recently-used algorithm. This happens when the in-memory cache becomes full and subsequent requests for new entries must be accommodated. The entries removed from memory are passivated to disk if disk overflow is enabled. If this number is high, consider increasing the in-memory cache size. The valid format is a positive integer.

Memory Hit Rate (per sec) The rate (per second) of the requests for this cacheable object type served from memory during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Memory Hits The number of requests for this cacheable object type served from memory. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Remote Cache Entries Received The number of entries received from cooperating dynamic caches in this cluster. The valid format is a positive integer.

Remote Cache Entry Receive Rate (per sec) The rate (per second) of entries received from cooperating dynamic caches in this cluster for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Remote Hit Rate (per sec) The rate (per second) of the requests for this cacheable object type served from other JVMs in the cluster during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Remote Hits The requests for this cacheable object type served from other JVMs in the cluster. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 39. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of Cache Templates Whether this row is a summary row of statistical totals collected for all the cache object types. The valid values are No and Yes.

Template Row Number The number of the row in the report that displays in the workspace, which is associated with a bar in the Ten Worst Hits Rate bar chart. The valid format is an integer.

Timeout Invalidation Rate (per sec) The rate (per second) at which entries are removed from the cache (memory or disk) because their timeout has expired. The valid format is a decimal (formatted to three decimal places).

Timeout Invalidations The number of entries of this cacheable object type that were removed from memory or disk because their timeout (as specified in the cache spec XML file) has expired. The valid format is a positive integer.

Total Explicit Invalidation Rate (per sec) The rate at which invalidations were issued for entries for this cacheable object type explicitly by the clients. The valid format is a decimal (formatted to three decimal places).

Total Explicit Invalidations The number of invalidations issued for entries of this cacheable object type explicitly by the clients (directly accessing the application server and by remote JVMs in this cluster). The valid format is a positive integer.

Total Hit Rate (per sec) The total hit rate per second. The valid format is a decimal (formatted to three decimal places).

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

EJB Containers attributes

The **EJB Containers** attributes provide aggregated information for each defined EJB container that aggregates bean performance data for all Enterprise beans deployed to that container.

The attributes also provide aggregated information for the application server that aggregates bean performance data for all Enterprise beans deployed to the application server. These attributes provide load values, response times, and lifecycle activities for Enterprise beans. Use the EJB Containers attributes in situations to monitor application server load and resource usage.

The attributes within this group are used to build the EJB Containers workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Activate Count The number of times a bean instance was activated during the interval. The valid format is a positive integer.

Activation Rate (per sec) The bean activations (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Active Method Count The average number of bean methods concurrently active during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Concurrently Live Beans The average number of bean objects concurrently live during the sampling interval. The valid format is a decimal

(formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Concurrently Ready Beans The average number of beans concurrently active during the last interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Container Name The name of the container. The valid format is an alphanumeric string, with a maximum of 64 characters.

Create Average Time (ms) The average method response time for creates during the interval. The valid format is a decimal (formatted to three decimal places).

Create Time Count The count of times that were averaged for Create Average Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Create Count The number of bean create calls during the interval. The valid format is a positive integer.

Creation Rate (per sec) The bean create calls (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Destroy Count The number of times bean objects were destroyed by garbage collection during the interval. The valid format is a positive integer.

Destruction Rate (per sec) The rate at which the beans are destroyed by garbage collection (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Entity Bean Load Count The number of times an entity bean data was loaded during the interval. The valid format is a positive integer.

Entity Bean Load Rate (per sec) The number of entity beans (per second) that were loaded during the interval. The valid format is a decimal (formatted to three decimal places).

Entity Bean Store Count The number of times entity bean data was written to the database during the interval. The valid format is a positive integer.

Entity Bean Store Rate (per sec) The entity bean stores (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Instantiate Count The number of times bean objects were instantiated during the interval. The valid format is a positive integer.

Instantiation Rate (per sec) The number of times bean objects were instantiated (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Instrumentation Level The instrumentation level for this EJB container. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Method Average Response Time (ms) The average response time (in milliseconds) on remote interface methods for all beans during the interval. The valid format is a decimal (formatted to three decimal places).

Method Response Time Count The count of bean response times averaged for Method Average Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Method Invocation Count The number of method invocations during the interval. The valid format is a positive integer.

Method Invocation Rate (per sec) The rate of invocations (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Passivate Count The number of times a bean instance was passivated during the interval. The valid format is a positive integer.

Passivation Rate (per sec) The bean passivations (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Removal Rate (per sec) The bean remove calls (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Remove Average Time (ms) The average method response time for removes during the interval. The valid format is a decimal (formatted to three decimal places).

Remove Time Count The count of times that were averaged for Remove Average Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Remove Count The number of bean remove calls during the interval. The valid format is a positive integer.

Request Count The number of requests during the interval. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 40. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of All Containers Whether this row is a summary row of statistical totals collected for all EJB containers. The valid values are No and Yes.

Total (ms) The total time used during the interval. The valid format is a decimal (formatted to three decimal places).

Total Create (ms) The total time (in milliseconds) of bean create calls during the interval. The valid format is a decimal (formatted to three decimal places).

Total Method Invocation (ms) The total time (in milliseconds) of method invocations during the interval. The valid format is a decimal (formatted to three decimal places).

Total Remove (ms) The total time (in milliseconds) of bean remove calls during the interval. The valid format is a decimal (formatted to three decimal places).

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Enterprise Java Beans attributes

The **Enterprise Java Beans** attributes provide performance information about each Enterprise Java Bean (EJB) deployed to the application server.

These attributes provide information about bean activity and bean object pool activity. Use the Enterprise Java Beans attributes in situations to monitor performance and problems for an individual bean. The attributes within this group are used to build the Enterprise Java Beans workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and reselect it later. Each time you reselect the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Activate Count The number of times a bean instance was activated during the interval. The valid format is a positive integer.

Activation Rate (per sec) The bean instance activations (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Active Method Count The average number of invocations being processed concurrently for all the methods during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Application EJB Module Name The name of the Web Application or EJB Module. The valid format is an alphanumeric string, with a maximum of 64 characters.

Average Concurrently Live Beans The average number of live bean objects during the interval, which include objects that were instantiated but not yet destroyed. This is a load value providing data on the average level as a function of time. It is the average number of bean objects that exist in the run time, whether active or pooled. This is a measure of how many resources the home interface is consuming. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Concurrently Ready Beans The average number of active beans during the interval. This is a load value providing data on the average level as a function of time. It is the average number of bean instances of the home that are in the ready state. This is a measure of how busy the server is. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Objects Discarded The average number of objects that were discarded each time the bean object pool was emptied of idle objects during the interval. The valid format is a decimal (formatted to three decimal places).

Objects Discarded Count The count of objects used for Average Objects Discarded. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Average Objects in Pool The average number of objects in the bean object pool during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Bean Name The name of the Enterprise JavaBean (EJB). This name prefixes the application name and the EJB compress name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Bean Type The type of bean. Valid values are Stateless, Stateful, Entity, and Message Driven.

Create Average Time (ms) The average method response time to create bean objects during the interval. The valid format is a decimal (formatted to three decimal places).

Create Time Count The count of times that were averaged for Create Average Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Create Count The number of create calls during the interval. The valid format is a positive integer.

Creation Rate (per sec) The create calls (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Destroy Count The average number of times bean objects were destroyed by garbage collection during the interval. The valid format is a positive integer.

Destruction Rate The rate of destructions (per second) for bean objects by the garbage collector during the interval. The valid format is a decimal (formatted to three decimal places).

Discard Count The number of times the returned object to the bean object pool was discarded because the bean object pool was already full during the interval. The valid format is a positive integer.

Discard Rate (per sec) The bean object pool discards (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Drain Count The number of times the bean object pool was found idle and an attempt was made to remove idle objects during the interval. The valid format is a positive integer.

Drain Rate (per sec) The drain rate (per second) for the bean object pool during the interval. The valid format is a decimal (formatted to three decimal places).

Entity Bean Load Count The number of times bean data was loaded during the interval. The valid format is a positive integer.

Entity Bean Load Rate (per sec) The bean data loads (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Entity Bean Store Count The number of times bean data was written to the database during the interval. The valid format is a positive integer.

Entity Bean Store Rate (per sec) The rate at which data was written (per second) to the database for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Find Count The number of times a retrieval call found an object available in the bean object pool during the interval. The valid format is a positive integer.

Find Rate (per sec) The bean object pool retrieve availability (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Get Count The number of calls that retrieved an object from the bean object pool during the interval. The valid format is a positive integer.

Get Rate (per sec) The number of times bean objects were retrieved (per second) during the interval. The valid format is a decimal (formatted to three decimal places).

Instantiate Count The number of times bean objects were created during the interval. The valid format is a positive integer.

Instantiation Rate (per sec) The bean objects created (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Instrumentation Level The instrumentation level of this enterprise bean. The valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Bean Object Pool Instrumentation Level The instrumentation level of the object pool for this enterprise bean. The valid values are None, Low, Medium, High, Basic, Extended, All, Custom, Maximum.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Method Average Response Time (ms) The average response time (in milliseconds) for all methods of the remote interface for this bean during the interval. The valid format is a decimal (formatted to three decimal places).

Method Response Time Count The count of times that were averaged for Method Average Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Method Invocation Rate (per sec) The invocations (per second) for all methods during the sampling interval. The Method Invocation Rate (per sec) is a load value that provides data on the average level as a function of time. This is a measure of how busy the server is. The valid format is a decimal (formatted to three decimal places).

Method Invocations The total number of remote interface method invocations during the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Passivate Count The average number of bean objects passivated during the sampling interval. Minimum instrumentation level required to collect these data: Extended. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Passivation Rate (per sec) The number of passivations (per second) during the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Removal Rate (per sec) The remove calls (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Remove Average Time (ms) The average method response time to remove bean objects during the interval. The valid format is a decimal (formatted to three decimal places).

Remove Time Count The count of times that were averaged for Remove Average Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Remove Count The number of remove calls during the interval. The valid format is a positive integer.

Return Count The number of calls that returned an object to the bean object pool during the interval. The valid format is a positive integer.

Return Rate (per sec) The bean object pool returns (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 41. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute

Table 41. Format of the 12-character timestamp (continued)

Character String	Meaning
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collection Analysis attributes

The **Garbage Collection Analysis** attribute group provides information about the garbage collector in the Java Virtual Machine that is hosting the application server.

The garbage collection attributes report the number of times the collector ran during the interval and the resulting number of objects that the collector freed. Use the Garbage Collection Analysis attributes in situations to monitor garbage-collection performance and possible problems. The attributes within this group are used to build the Garbage Collection Analysis and the WebSphere application server workspaces.

ASID The identifier (decimal) assigned to the address space running this servant region.

GC Rate (per min) The rate (per minute) at which the Java Virtual Machine is invoking its garbage-collection routine. The valid format is a decimal (formatted to three decimal places).

Heap Used (%) The percentage of heap used at the end of the interval. The valid format is a decimal (formatted to one decimal place).

Max Heap KBytes The maximum heap size for the JVM (the -Xmx parameter, if specified) The valid format is a 4-byte integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Kbytes Free The total number of free kilobytes in the heap at the end of the last garbage-collection cycle during the interval. The valid format is a positive integer.

Kbytes Total Freed by GC The total number of kilobytes freed by the garbage collector during the interval. The valid format is a positive integer.

Kbytes Used The number of kilobytes in the heap that were in use at the end of the last garbage collection cycle during the interval. The valid format is a positive integer.

Kbytes Used Delta The difference between the Kbytes Used value for this interval and the Kbytes Used value for the prior interval. The valid format is a positive or negative integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Objects Freed The number of objects the garbage collector freed during the interval (only supported for IBM JDK). The valid format is a positive integer. Not monitored on non-IBM Java Virtual Machines, including objects commonly used on HP-UX and Solaris platforms.

Objects Moved The number of objects the garbage collector moved during the interval (only supported for IBM JDK). The valid format is a positive integer. Not monitored on non-IBM Java Virtual Machines, including objects commonly used on HP-UX and Solaris platforms.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Real Time (ms) The total real time (in milliseconds) the garbage collector required during the most recent cycle. The valid format is a positive integer.

Real Time (%) The percentage of real time that the garbage collector was active during the interval. The valid format is a decimal (formatted to one decimal place).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 42. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Times Run The number of times the garbage collector ran during the interval. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collection Cycle attributes

The **Garbage Collection Cycle** attribute group provides information about a single garbage-collection cycle that the Java Virtual Machine hosting the application server performed.

The Garbage Collection Cycle attributes report the free heap space both before and after garbage collection, the heap space freed, and the number of objects moved during garbage collection. Use the Garbage Collection Cycle attributes in situations to examine the results of a particular garbage collection.

The attributes within this group are used to build the Garbage Collections - Selected Allocation Failure workspace.

Important: This information is only collected for systems using an IBM JVM.

Allocation Failure Number The identifier assigned to the allocation-failure block for which the JVM ran the current garbage-collection cycle, which is associated with a bar in the Heap Usage - History bar chart. If your Java code called System.gc to start garbage collection, this number is 0. The valid format is a positive integer.

ASID The identifier (decimal) assigned to the address space running this servant region.

Compact (ms) The time (in milliseconds) required for the compaction phase of the garbage-collection cycle. The valid format is a positive integer.

Compaction Reason The code describing the reason garbage collection was initiated. The valid format is a positive integer. The compaction codes are:

Table 43. Reasons for initiating garbage collection

Compaction Code	Definition
1	Insufficient free space for the allocation request following the mark and sweep phases.
2	The heap is fragmented and benefits from a compaction.
3	Less than 15% free space available.
4	A call to System.gc requested garbage collection.

Table 43. Reasons for initiating garbage collection (continued)

Compaction Code	Definition
5	Less than 5% free space available.
6	Less than 128 K free space available.
7	Parameter Xcompactgc specified.
8	The transient heap has less than 5% free space available.
9	The heap is fragmented (this code marks additional reasons for compaction apart from compaction code 2).

Final References The number of final reference objects collected during this garbage-collection cycle. The valid format is a positive integer.

Garbage Collection Date and Time The date and time the Java Virtual Machine invoked the garbage collector. The valid format is a 16-character timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Note to Solaris and HP-UX users: Since allocation-failure information is not recorded on these platforms, this column is always empty on these platforms.

Garbage Collection Number The number of this garbage-collection cycle. The valid format is a positive integer.

Heap Capacity (Kbytes) The total number of kilobytes allocated to the main heap after this garbage-collection cycle. The valid format is a positive integer.

Heap Free (%) after GC The percentage of heap space that is available after this garbage-collection cycle. The valid format is a decimal (formatted to one decimal place).

Heap Space Free (Kbytes) The number of kilobytes available within the heap after this garbage-collection cycle. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

KBytes Free at Start of GC The number of kilobytes available in the heap before garbage collection began. The valid format is a positive integer.

Kbytes Freed The number of kilobytes freed by the garbage collector. The valid format is a positive integer.

Kbytes Moved The number of kilobytes moved on the heap during this compaction. The valid format is a positive integer.

Kbytes Used The number of kilobytes in the heap that were in use after this garbage-collection cycle. The valid format is a positive integer.

Mark (ms) The time (in milliseconds) required for the mark phase of the garbage-collection cycle. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Objects Moved The number of objects the garbage collector moved during this compaction. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Phantom References The number of phantom reference objects collected during this garbage-collection cycle. "Phantom" refers to a specific Java class that defines object reachability. The valid format is a positive integer.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 44. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Soft References The number of soft reference objects collected during this garbage-collection cycle. "Soft" refers to a specific Java class that defines object reachability. The valid format is a positive integer.

Sweep (ms) The time (in milliseconds) required for the sweep phase of the garbage-collection cycle. The valid format is a positive integer.

Time to Complete (ms) The time (in milliseconds) required to complete this garbage-collection cycle. The valid format is a positive integer.

Weak References The number of weak reference objects collected during this garbage-collection cycle. "Weak" refers to a specific Java class that defines object reachability. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

High Availability Manager attributes

The **High Availability Manager** attributes provide aggregated information about the high availability managers.

The attributes within this group are used to build the High Availability Manager workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Bulletin-Board Rebuild Time The time taken (in milliseconds) to rebuild the global state of the bulletin-board. During this time no messages are received by the subscribers. If this time is too high, and is unacceptable, you might want to increase the number of coordinators. The valid format is a decimal (formatted to three decimal places).

Bulletin-Board Rebuild Time Count The count of times that were used for Bulletin-Board Rebuild Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Bulletin-Board Subjects The total number of subjects managed. The valid format is a positive integer. This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Bulletin-Board Subscriptions The total number of bulletin-board subscriptions. The valid format is a positive integer. This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Group State Rebuild Time The time taken (in milliseconds) to rebuild the global group state. During the rebuild time, no fail-over can happen. If this time is too high and is unacceptable for the desired availability, you might want to increase the number of coordinators. For proper operation of this counter, you must host the active coordinator in an application server other than the deployment manager. The valid format is a decimal (formatted to three decimal places).

Group State Rebuild Time Count The count of times that were used for Group State Rebuild Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Instrumentation Level The instrumentation level for availability manager counters. For WebSphere 5, the valid values are None, Low, Medium, High, and Maximum; for WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Local Bulletin-Board Subjects The total number of subjects being posted locally. The number includes the proxy postings (if any) done by the core group bridge service on behalf of servers belonging to different WebSphere cells. The valid format is a positive integer. This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Local Bulletin-Board Subscriptions Indicates the total number of bulletin-board subscriptions being posted locally.

Local Groups The total number of local groups. The valid format is a positive integer. This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 45. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

J2C Connection Pools attributes

The **J2C Connection Pools** attribute group provides information about connectors that adhere to J2C, the WebSphere application server implementation of the J2C architecture.

Data counters for this category contain usage information about the J2C architecture that enables enterprise beans to connect and interact with procedural backend systems, such as Customer Information Control System (CICS) and Information Management System (IMS). Examples include the number of managed connections or physical connections and the total number of connections or connection handles.

The attributes within this group are used to build the J2C Connection Pools workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Application ID Indicates JEE application ID.

Average Free Connections The average number of free Managed Connections for the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Pool Size The average number of Managed Connections for the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Usage Time (ms) The average time (in milliseconds) that connections are in use (measured from when the connection is allocated to when it is returned). The valid format is a decimal (formatted to three decimal places).

Usage Time Count The count of times that were averaged for Average Usage Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Average Wait Time (ms) The average waiting time (in milliseconds) until a connection is granted for the interval. The valid format is a decimal (formatted to three decimal places).

Wait Time Count The count of times that were averaged for Average Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Concurrent Waiting Threads The average number of threads concurrently waiting for a connection for the interval. The valid format is a positive integer. This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Connection Allocation Rate (per sec) The rate (per second) of application connections allocated from Managed Connections for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Connection Creation Rate (per sec) The rate (per second) of Managed Connections created for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Connection Destruction Rate (per sec) The rate (per second) of Managed Connections destroyed for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Connection Factory Name The name of the connection factory. The valid format is an alphanumeric string, with a maximum of 256 characters.

Connection Handles The number of open application connections that have been allocated from the managed connections. The valid format is a positive integer.

Connection Pool Timeout Rate (per sec) The rate (per second) of connection pool timeouts for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Connection Pool Timeouts The number of faults, such as timeouts, in connection pools for the sampling interval. The valid format is a positive integer.

Connection Return Rate (per sec) The rate (per second) of allocated application connections that have been returned for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Connection Type The J2C connection type. The valid format is an alphanumeric string, with a maximum of 32 characters.

Connections Allocated The number of application connections allocated from Managed Connections. The valid format is a positive integer.

Connections Created The total number of Managed Connections created during the sampling interval. The valid format is a positive integer.

Connections Destroyed The number of Managed Connections destroyed during the sampling interval. The valid format is a positive integer.

Connections Granted The number of Managed Connections granted during the interval. The valid format is a positive integer.

Connections Returned The number of allocated application connections that have been returned (closed) during the sampling interval. The valid format is a positive integer.

Connections Used The number of Managed Connection objects available in a particular connection pool. This number includes all Managed Connection objects that have been created but not destroyed. The valid format is a positive integer.

Factory Label The abbreviated name of the connection factory. The valid format is an alphanumeric string, with a maximum of 32 characters.

Instrumentation Level The instrumentation level for the J2C connection pools. For WebSphere 5, the valid values are None, Low, Medium, High, and Maximum; for WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval Time The length (in seconds) of the sampling interval. The valid format is a positive integer.

Maximum Pool Size The maximum number of managed connections that can be created in this connection pool (blank for each individual managed connection). The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Percent of Time Pool at Max The average percent of the time that all connections are in use for the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Percent Used Bad The bad percent of pool usage by application. The valid format is a positive integer.

Percent Used Fair The fair percent of pool usage by application. The valid format is a positive integer.

Percent Used Good The good percent of pool usage by application. The valid format is a positive integer.

Pool Used (%) The average percent of the pool that is in use for the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Row Number The number of the row within the report, which corresponds to a bar in the Highest Miss Rates bar chart. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 46. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of J2C Connections Whether this row is a summary row of statistical totals collected for all the J2C connection pools. Valid values are No and Yes.

Total Usage (ms) The total time (in milliseconds) the connection object used. The valid format is a decimal (formatted to three decimal places).

Total Wait (ms) The total time (in milliseconds) the connection object waited. The valid format is a decimal (formatted to three decimal places).

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JMS Summary attributes

The **JMS Summary** attributes provide information about how WebSphere application server applications are interacting with messaging middleware (WebSphere MQ) using the Java Messaging Service (JMS). It provides such information as which queue managers and queues are being used and how many messages are being read and written.

The attributes within this group are used to build the JMS Summary workspace.

Important: The attributes within this attribute group contain meaningful values only if your site has set the request data monitoring level to Level2 to collect data on JMS requests.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer or -1 which means that Application ID is aggregated statistic for all applications.

ASID The identifier (decimal) assigned to the address space running this servant region.

Average Processing Time (ms) The average time (in milliseconds) per request using the JMS. The valid format is a decimal (formatted to three decimal places).

Processing Time Count The count of times that were averaged for Average Processing Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Browse Average Time (ms) The average time (in milliseconds) for each browse request from the queue. The valid format is a decimal (formatted to three decimal places).

Browse Count The number of messages browsed from the queue. The valid format is a positive integer.

Browse Rate (per sec) The number of messages (per second) browsed from a JMS queue. The valid format is a decimal (formatted to three decimal places).

Browse Total Time (ms) The total time (in milliseconds) consumed by browse requests from the queue. The valid format is a positive integer.

Full Name The complete name of the message queue, which consists of the queue manager name concatenated to the queue name and separated by a slash. The valid format is an alphanumeric string, with a maximum of 100 characters.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

JMS Connection Label A shortened version of the full name. The valid format is an alphanumeric string, with a maximum of 12 characters.

Manager Name The name of the WebSphere MQ queue manager (not available currently). This attribute is blank if WebSphere MQ is not being used. The valid format is an alphanumeric string, with a maximum of 48 characters.

Name The name of the WebSphere MQ queue. The valid format is an alphanumeric string, with a maximum of 48 characters.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Publish Average Time (ms) The average time (in milliseconds) for each publish request to be sent to the queue. The valid format is a decimal (formatted to three decimal places).

Publish Count The number of publish requests sent to the queue. The valid format is a positive integer.

Publish Rate (per sec) The number of publish requests (per second) sent to a JMS queue. The valid format is a decimal (formatted to three decimal places).

Publish Total Time (ms) The total time (in milliseconds) consumed by all publish requests for the queue. The valid format is a positive integer.

Receive Average Time (ms) The average time (in milliseconds) for each get from the queue. The valid format is a decimal (formatted to three decimal places).

Receive Count The number of destructive gets from the queue. The valid format is a positive integer.

Receive Rate (per sec) The number of destructive gets (per second) made from the queue. The valid format is a decimal (formatted to three decimal places).

Receive Total Time (ms) The total time (in milliseconds) consumed by gets from the queue. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 47. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Send Average Time (ms) The average time (in milliseconds) for each put to the queue. The valid format is a decimal (formatted to three decimal places).

Send Count The number of messages put to the queue. The valid format is a positive integer.

Send Rate (per sec) The number of messages (per second) put to the queue. The valid format is a decimal (formatted to three decimal places).

Send Total Time (ms) The total time (in milliseconds) consumed by puts to the queue. The valid format is a positive integer.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total Time (ms) The total time (in milliseconds) spent accessing the queue. The valid format is a positive integer.

Type The type of message manager. The valid values are Queue and Topic.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Log Analysis attributes

The **Log Analysis** attributes provide application server error and exception conditions which are recorded in the application server log files.

The log files are SystemOut.log and SystemErr.log. Use the Log Analysis attributes in situations to monitor errors and exception conditions and their severity.

The attributes within this group are used to build the Log Analysis workspace.

ASID The identifier (decimal) assigned to the address space running this servant region.

Component The name of the component that caused the error. The valid format is an alphanumeric string, with a maximum of 32 characters.

Error Date and Time The date and time the event occurred. The valid format is a timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Job ASID The identifier (hexadecimal) assigned to the address space running this servant region. The valid format is an alphanumeric string, with a maximum of four characters.

Job Name The job name assigned to this servant region. The valid format is an alphanumeric string, with a maximum of eight characters.

Message ID The identifier assigned to the message. The valid format is an alphanumeric string, with a maximum of 12 characters.

Message Origin Where the message originates; that is, the log file name and line number. The valid format is an alphanumeric string, with a maximum of 32 characters. This field is not empty only on the z/OS system.

Message Text The text of the message. The valid format is alphanumeric string, with a maximum of 256 characters. All error message text data that goes beyond 256 characters are truncated and are not shown in the portal.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process identifier of the Java virtual machine. The valid format is an alphanumeric string, with a maximum of eight characters. In a z/OS system, this field displays in hexadecimal format.

Sequence Number The sequence number in the JMX notifications stream. The valid format is positive integer.

Server Instance Name The name of the application server instance. This is the name of a single address space that can run application code (called a "specific server" or simply a "server"). The valid format is an alphanumeric string, with a maximum of eight characters.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Severity The severity of the message. The valid values are:

Table 48. Message severities and their meanings

Severity	Definition
Informational	A message intended to convey only user information
Unconditional	A message of type Unconditional
Dump	A message of type Dump
SystemOut	A message written directly to System.out by the user application or internal components
SystemError	A message written directly to System.err by the user application or internal components
User	A message of type User
EntryMethod	A message written upon entry to a method
ExitMethod	A message written upon exit from a method
Event	A message of type Event
Debug	A message of type Debug
Audit	An audit message

Table 48. Message severities and their meanings (continued)

Severity	Definition
Warning	A warning message
Error	An error message
Terminate	A message of type Terminate (exit process)
Fatal	A fatal message
Unknown	A placeholder that indicates the message type was not recognized

Thread ID The unique identifier of the thread where the event occurred. The valid format is an alphanumeric string, with a maximum of 16 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Listener Port attributes

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Sample Date and Time The date and time at which the agent has collected data.

Interval Time Indicates the length of the interval in seconds. The valid format is a 4-byte integer.

Name Name of the Listener Port. The valid format is an alphanumeric string, with a maximum of 256 characters.

Status The status of the Listener Port: Started or Stopped.

JMS Connection Factory The JNDI name for the JMS ConnectionFactory in use by this Listener Port. The valid format is an alphanumeric string, with a maximum of 256 characters.

JMS Destination The JNDI name for the JMS Queue or Topic in use by this Listener Port. The valid format is an alphanumeric string, with a maximum of 256 characters.

Max Sessions The value of max sessions in use by this Listener Port. The valid format is a 4-byte integer.

Max Retries The value of max retries in use by this Listener Port. The valid format is a 4-byte integer.

Max Messages The value of max messages in use by this Listener Port. The valid format is a 4-byte integer.

Messaging Engine Communications attributes

The **Messaging Engine Communications** attributes display statistics for all the messaging engines being hosted by the current application server.

The attributes within this group are used to build the Messaging Engine Communications workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

API Connections The number of sessions being used by messaging engines that are currently network connected to this application server. The valid format is a positive integer.

Buffered Reads (bytes) The number of bytes of data that have been received from the network and are held pending further processing. Large values might indicate that the application server is unable to process data fast enough to keep up with the other application server processes hosting messaging engines. The valid format is a positive integer.

Buffered Writes (bytes) The number of bytes of data being held pending transmission. Large values might indicate network congestion or application server processes hosting messaging engines which are unable to process data fast enough to keep up with the application server. The valid format is a positive integer.

Errors The communication errors that have occurred and resulted in a network connection to a messaging engine being disconnected. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the Messaging Engine Communications. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Message Read (bytes) The number of bytes of message data received from application server processes hosting messaging engines over network connections. This does not include data used to negotiate the transmission of messages. The valid format is a positive integer.

Message Written (bytes) The number of bytes of message data sent to application server processes hosting messaging engines over network connections. This does not include data used to negotiate the transmission of messages. The valid format is a positive integer.

Messaging Engine Attached The number of distinct application server processes hosting messaging engines currently network connected to this application server. The valid format is a positive integer.

Messaging Engine Name The name of the message engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Reads The number of read operations used to receive data from application server processes hosting messaging engines via network connections. The valid format is a positive integer.

Reads Blocked The number of read operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with application server processes hosting messaging engines. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 49. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all messaging engine communications. Valid values are No and Yes.

Total Read (bytes) The Number of bytes of data received from application server processes hosting messaging engines. The valid format is a positive integer.

Total Written (bytes) The Number of bytes of data sent to application server processes hosting messaging engines. The valid format is a positive integer.

Writes The number of write operations used to transmit data to application server processes hosting messaging engines via network connections. The valid format is a positive integer.

Writes Blocked The number of write operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with application server processes hosting messaging engines. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Messaging Engines attributes

The **Messaging Engines** attributes display aggregated information about the performance of the messaging engines supported by WebSphere server.

The attributes within this group are used to build the Messaging Engines workspace.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Important: Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Average Local Wait Time (ms) The time spent by messages on this durable subscription at consumption. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Local Wait Time Count The count of times that were averaged for Average Local Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Expired Messages The number of report enabled messages that expired while on this queue. The valid format is a positive integer.

Incomplete Topic Publications The number of publications not yet received by all current subscribers. If this number is unexpected, view the publication using the admin console to take action. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the messaging engines. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Messaging Engine Name The name of the messaging engine. The valid format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 50. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Status The status of the messaging engine. The valid format is a positive integer.

Summary of All Applications Whether this row is a summary row of statistical totals for all messaging engines. Valid values are No and Yes.

Total Published The total number of publications to the messaging engines. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Portal Page Summary attributes

The **Portal Page Summary** attributes provide information about response times statistics for all portal page requests that completed on monitored WebSphere Portal server during the interval.

The attributes within this group are used to build the Portal Pages Summary workspace.

ASID The identifier (decimal) assigned to the address space running this servant region.

Average Response Time (ms) The average response time (in milliseconds) of requests are processed by the portal pages during the current interval. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Portal Page Name The name of the portal page. The value format is an alphanumeric string, with a maximum of 128 characters.

Request Count The count of requests processed by the portlet page during the current interval. The valid format is a positive integer.

Row Number The number of the row. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 51. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total Response (ms) The total time (in milliseconds) of responses. The valid format is a decimal (formatted to three decimal places).

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Portal Summary attributes

The **Portal Summary** attributes provide aggregated response times statistics about all portal pages and portlet requests that completed on monitored WebSphere Portal server during the interval.

The attributes within this group are used to build the Portal Summary workspace.

ASID The identifier (decimal) assigned to the address space running this servant region.

Authentication Request Count The number of authentication requests during the interval. The valid format is a positive integer.

Authentication Total (ms) The total time (in milliseconds) of authentication requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Authorization Request Count The number of authorization requests during the interval. The valid format is a positive integer.

Authorization Total (ms) The total time (in milliseconds) of authorization requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Average Response Time of Portal Pages (ms) The average response time (in milliseconds) of all portal pages/Gateway Servlet requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Average Response Time of Portlets (ms) The average response time (in milliseconds) of all portlets requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Model Building Request Count The number of model building requests during the interval. The valid format is a positive integer.

Model Building Total (ms) The total time (in milliseconds) of model building requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Page Loading Request Count The number of page loading requests during the interval. The valid format is a positive integer.

Page Loading Total (ms) The total time (in milliseconds) of page loading requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Portal Page Request Count The number of portal page requests during the interval. The valid format is a positive integer.

Portal Pages Total Response (ms) The total response time (in milliseconds) of all portal pages/Gateway Servlet requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Portlets Request Count The number of portlets requests during the interval. The valid format is a positive integer.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Portlets Total Response(ms) The total response time (in milliseconds) of all portlets requests completed during the interval. The valid format is a decimal (formatted to three decimal places).

Response Time of Authentication (ms) The response time (in milliseconds) of authentication. The valid format is a positive integer.

Response Time of Authorization (ms) The response time (in milliseconds) of authorization. The valid format is a positive integer.

Response Time of Model Building (ms) The response time (in milliseconds) of model building. The valid format is a positive integer.

Response Time of Page loading (ms) The response time (in milliseconds) of page loading. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 52. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Portlet Summary attributes

The **Portlet Summary** attributes provide information about response times of all portlet requests that completed on monitored WebSphere Portal.

The attributes within this group are used to build the Portlet Summary workspace.

ASID The identifier (decimal) assigned to the address space running this servant region.

Average Response Time (ms) The average response time for portlet during the current interval. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Portlet Name The name of the portlet. The value format is an alphanumeric string, with a maximum of 256 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Request Count The count of requests for portlet during the current interval. The valid format is a positive integer.

Request Rate The rate at which the requests processed by portlet during the current interval. The valid format is a decimal (formatted to three decimal places).

Row Number The number of the row. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 53. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total Response (ms) The total response time for portlet during the current interval. The valid format is a decimal (formatted to three decimal places).

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Queue attributes

The **Queue** attributes provide aggregated information about the point to point messaging.

The attributes within this group are used to build the Destinations workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Aggregate Message Wait Time The time spent by messages in the bus at consumption. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Aggregate Message Wait Time Count The count of times that were used for Aggregate Message Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Assured Persistent Messages Consumed The number of assured persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Assured Persistent Messages Produced The number of assured persistent messages produced, for the lifetime of this messaging engine. The valid format is a positive integer.

Available Message The number of messages available for a queue for consumption. If this number is close to the destination high messages value, review the high messages value. The valid format is a positive integer.

Best Effort Non-persistent Messages Consumed The number of best effort non-persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Best Effort Non-persistent Messages Produced The number of best effort non-persistent messages produced, for the lifetime of this messaging engine. The valid format is a positive integer.

Express Non-persistent Messages Consumed The number of express non-persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Express Non-persistent Messages Produced The number of express non-persistent messages produced, for the lifetime of this messaging engine. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the database connection pool for the data source. Valid values are None, Low, Medium, High, Basic, Extended, All, Custom, and Maximum. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Local Consumer The number of currently attached local consumers. The valid format is a positive integer.

Local Consumer Attaches The number of times an attachment has been made to this queue by local consumers. The lifetime of this value is the lifetime of the messaging engine. The valid format is a positive integer.

Local Message Wait Time The time spent by messages on this queue at consumption. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Local Message Wait Time Count The count of times that were used for Local Message Wait Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Local Oldest Message Age The longest time any message has spent on this queue. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Local Producer The number of currently attached local producers. The valid format is a positive integer.

Local Producer Attaches The number of times an attachment has been made to this queue by local producers. The lifetime of this value is the lifetime of the messaging engine. The valid format is a positive integer.

Messaging Engine Name The name of the messaging engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Queue Name The name of the queue. The value format is an alphanumeric string, with a maximum of 256 characters.

Reliable Non-persistent Messages Consumed The number of reliable non-persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Reliable Non-persistent Messages Produced The number of reliable non-persistent messages produced, for the lifetime of this messaging engine. The valid format is a positive integer.

Reliable Persistent Messages Consumed The number of reliable persistent messages consumed, for the lifetime of this messaging engine. The valid format is a positive integer.

Reliable Persistent Messages Produced The number of reliable persistent messages produced, for the lifetime of this messaging engine. The valid format is a positive integer.

Report Enabled Messages Expired The number of report enabled messages that expired while on this queue. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 54. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day

Table 54. Format of the 12-character timestamp (continued)

Character String	Meaning
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Total Messages Consumed The total number of messages consumed from this queue, for the lifetime of this messaging engine. The valid format is a positive integer.

Total Messages Produced The total number of messages produced to this queue, for the lifetime of this messaging engine. The valid format is a positive integer.

Unavailable Message The number of messages locked or uncommitted. This means messages that have been added or removed but the transaction has not been committed yet. If this number is high, check which messages are locked and why.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Remote Configuration Requests attributes

The **Remote Configuration Requests** attributes provide information about remote configuration.

The attributes within this group are used to build the WebSphere Agent workspace.

Command Indicates the Command of the request.

Node Name The system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Identifier The Request ID. The valid format is an alphanumeric string, with a maximum of 48 characters.

Package Count Indicates the package count for this request/response.

Package Index Indicates the current index of the packages.

Request Context Indicates the request context.

Response Context Indicates the response context.

Return Value Indicates the return value of the request.

Target Agent Code Two-letter agent product code that specifies information about the product being configured.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Request Analysis attributes

The **Request Analysis** attributes provide response times and functional decomposition information about requests (servlets, JSPs, SCA requests, Portal requests, EJB methods, and web services) that ran on the application server.

The attributes within this group are used to build the Request Analysis workspace.

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Important: The application name is, in the present version, not provided for SCA requests.

Application Time (ms) On monitoring level 1, the average time (in milliseconds) this request spent processing application requests other than JCA, JMS, JNDI, JDBC, web service, and SCA requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. On monitoring level 2, the average time (in milliseconds) this request spent processing Servlet and EJB requests. The valid format is a positive integer.

Application Time Count The count of times that were averaged for Application Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Application Time Percent The percentage of time this request spent processing application requests other than JCA, JMS, JNDI, JDBC requests, web services, and SCA. The valid format is a decimal (formatted to one decimal place).

Application Time Percent Count The count of individual percentages that were averaged for Application Time Percent. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Application Bad Delay (msec) The delay time (in milliseconds) in the application tier collected during the interval. This column is non-zero value when application delay exceeds the bad threshold configured for it. The valid format is a positive integer.

Application Fair Delay (msec) The delay time (in milliseconds) in the application tier collected during the interval. This column is non-zero value when application delay exceeds the fair threshold configured for it. The valid format is a positive integer.

Application Good Delay (msec) The delay time (in milliseconds) in the application tier collected during the interval. This column is non-zero when application delay is less than fair threshold configured for it. The valid format is a positive integer.

Application Tier Delay Type This attribute defines the request type based on its delay time in the application tier. Valid values are Unknown, Good, Fair, and Bad.

Application Tier Response (msec) The actual response time in milliseconds in the application tier collected during the interval. The valid format is a positive integer.

Average Response (ms) The average time (in milliseconds) required each time this request ran during the interval. The valid format is a positive integer.

ASID The identifier (decimal) assigned to the address space running this servant region.

Backend Bad Delay (msec) The delay time (in milliseconds) in the backend tier collected during the interval. This column is non-zero value when backend delay exceeds the bad threshold configured for it. The valid format is a positive integer.

Backend Fair Delay (msec) The delay time (in milliseconds) in the backend tier collected during the interval. This column is non-zero value when backend delay exceeds the fair threshold configured for it. The valid format is a positive integer.

Backend Good Delay (msec) The delay time (in milliseconds) in the backend tier collected during the interval. This column is non-zero when backend delay is less than fair threshold configured for it. The valid format is a positive integer.

Backend Tier Delay Type This attribute defines the request type based on its delay time in the backend tier. Valid values are Unknown, Good, Fair, and Bad.

Backend Tier Response (msec) The actual response time in milliseconds in the backend tier collected during the interval. The valid format is a positive integer.

Client Bad Delay (msec) The delay time (in milliseconds) in the client tier collected during the interval. This column is non-zero value when client delay exceeds the bad threshold configured for it. The valid format is a positive integer.

Client Fair Delay (msec) The delay time (in milliseconds) in the client tier collected during the interval. This column is non-zero value when client delay exceeds the fair threshold configured for it. The valid format is a positive integer.

Client Good Delay (msec) The delay time (in milliseconds) in the client tier collected during the interval. This column is non-zero when client delay is less than fair threshold configured for it. The valid format is a positive integer.

Client Tier Delay Type This attribute defines the request type based on its delay time in the client tier. Valid values are Unknown, Good, Fair, and Bad.

Client Tier Response (msec) The actual response time in milliseconds in the client tier collected during the interval. The valid format is a positive integer.

Completion Count The number of requests that successfully completed during the interval. The valid format is a positive integer.

Custom Request Count The number of custom requests. The valid format is a positive integer.

Custom Request Time (ms) The average time (in milliseconds) the custom requests spent. The valid format is a positive integer.

Custom Request Time (%) The percentage of time the custom requests spent. The valid format is a decimal (formatted to one decimal place).

EJB Count The number of times this request invoked an Enterprise Java Bean (EJB) request. The valid format is a positive integer.

EJB Time (ms) The average time (in milliseconds) this request spent processing Enterprise Java Bean (EJB) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

EJB Time (%) The percentage of time this request spent processing Enterprise Java Bean (EJB) requests. The valid format is a decimal (formatted to one decimal place).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA Count The number of times this request invoked a JEE Connector Architecture (JCA) request. The valid format is a positive integer.

JCA Time (ms) The average time (in milliseconds) this request spent processing JEE Connector Architecture (JCA) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

JCA Time (%) The percentage of time this request spent processing JEE Connector Architecture (JCA) requests. The valid format is a decimal (formatted to one decimal place).

JMS Count The number of times this request invoked a Java Message Service (JMS) request. The valid format is a positive integer.

JMS Time (ms) The average time (in milliseconds) this request spent processing Java Message Service (JMS) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

JMS Time (%) The percentage of time this request spent processing Java Message Service (JMS) requests. The valid format is a decimal (formatted to one decimal place).

JNDI Count The number of times this request invoked a Java Naming and Directory Interface (JNDI) request. The valid format is a positive integer.

JNDI Time (ms) The average time (in milliseconds) this request spent processing Java Naming and Directory Interface (JNDI) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

JNDI Time (%) The percentage of time this request spent processing Java Naming and Directory Interface (JNDI) requests. The valid format is a decimal (formatted to one decimal place).

Level 2 Request Count The number of times this request was run with Mod Level 2 turned on. The valid format is a positive integer.

Level 2 Total Time (ms) The total time (in milliseconds) this request was run with Mod Level 2 turned on. The valid format is a positive integer.

Longest Response (ms) The maximum time (in milliseconds) it took this request to run during the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Portal Processing Count The number of times the request invoked a WebSphere Portal page or portlet request. The valid format is a positive integer.

Portal Processing Time (ms) The average time (in milliseconds) the request spent in processing WebSphere Portal page or portlet requests. This field can have a zero value if the total time is less than the number of requests. The valid format is a positive integer.

Portal Processing Time (%) The percentage of time the request spent in processing WebSphere Portal page or portlet requests. The valid format is a decimal (formatted to one decimal place).

Process ID The process identifier of the Java virtual machine. The valid format is a positive integer.

Request Bad Response Threshold (msec) The threshold that defines the bad requests. A request that spends more time to complete than this threshold to complete is a bad request. The valid format is a positive integer.

Request Completion (%) The percentage of the requests that completed successfully during the interval. The valid format is a positive integer.

Request Completion Level The completion level of the requests during the interval. Valid values are Unknown, Good, Fair, and Bad.

Request Count The number of times this request ran during the interval. The valid format is a positive integer.

Request Detail The URI for servlet requests, or the method name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Fair Response Threshold (msec) The threshold that defines the fair requests. A request that spends more time than this threshold and less time than the *Request Bad Response Threshold (msec)* attribute to complete is a fair request. The valid format is a positive integer.

Request Bad Delay (msec) The delay time (in milliseconds) collected during the interval. This column is non-zero value when the whole request response time exceeds the bad threshold configured for it. The valid format is a positive integer.

Request Fair Delay (msec) The delay time (in milliseconds) collected during the interval. This column is non-zero value when the whole request response time exceeds the fair threshold configured for it. The valid format is a positive integer.

Request Good Delay (msec) The delay time (in milliseconds) collected during the interval. This column is non-zero value when the whole request response time is less than fair threshold configured for it. The valid format is a positive integer.

Request Delay Type The type of the request delay. Valid values are Unknown, Good, Fair, and Bad.

Request Label A shortened version of Request Name, used to display the request name in the chart view. The valid format is an alphanumeric string, with a maximum of 32 characters.

Request Name The URL for servlet requests, or the fully qualified class name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Rate (per sec) The number of requests completed per second during the interval. If the sampling rate is less than 100%, this number is extrapolated to estimate 100% of completed requests. The valid format is a decimal (formatted to three decimal places).

Request Type The type of request being run. Valid values are Servlet, EJB_Method, Custom, All_Workloads, Unknown, Portlet, SCA, web service provider, and web service client.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 55. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of requests being sampled. The valid format is a positive integer.

SCA Count The number of times this request invoked a Service Component Architecture (SCA) request. The valid format is a positive integer.

SCA Time (ms) The average time (in milliseconds) this request spent processing Service Component Architecture (SCA) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SCA Time (%) The percentage of time this request spent processing Service Component Architecture (SCA) requests. The valid format is a decimal (formatted to one decimal place).

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet Count The number of times this request invoked a Servlet request. The valid format is a positive integer.

Servlet Time (ms) The average time (in milliseconds) this request spent processing Servlet requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

Servlet Time (%) The percentage of time this request spent processing Servlet requests. The valid format is a decimal (formatted to one decimal place).

SQL Connect Count The number of times this request connected to a JDBC database. The valid format is a positive integer.

SQL Connect Time (ms) The average time (in milliseconds) this request spent connecting to a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Connect Time (%) The percentage of time this request spent connecting to a JDBC database. The valid format is a decimal (formatted to one decimal place).

SQL Execute Count The number of times this request executed a JDBC database. The valid format is a positive integer.

SQL Execute Time (ms) The average time (in milliseconds) this request spent executing a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Execute Time (%) The percentage of time this request spent executing a JDBC database. The valid format is a decimal (formatted to one decimal place).

SQL Query Count The number of times this request queried a JDBC database. The valid format is a positive integer.

SQL Query Time (ms) The average time (in milliseconds) this request spent querying a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Query Time (%) The percentage of time this request spent querying a JDBC database. The valid format is a decimal (formatted to one decimal place).

SQL Update Count The number of times this request updated a JDBC database. The valid format is a positive integer.

SQL Update Time (ms) The average time (in milliseconds) this request spent updating a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Update Time (%) The percentage of time this request spent updating a JDBC database. The valid format is a decimal (formatted to one decimal place).

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total Time (ms) The total CPU time (in milliseconds) this request consumed during the interval. The valid format is a positive integer.

Web Service Count The number of times this request invoked a Web Service request. The valid format is a positive integer.

Web Service Time (ms) The average time (in milliseconds) this request spent processing Web Service requests. This field can have a zero value if the total time is less than the number of requests. The valid format is a positive integer.

Web Service Time (%) The percentage of time this request spent processing Web Service requests. The valid format is a decimal (formatted to one decimal place).

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Requests Monitoring Configuration attributes

The **Requests Monitoring Configuration** attributes provide information for all requests monitored in application. Use these attributes to monitor application edge requests. The agent supports three types of edge requests, Servlet/JSP, EJB, and Portal.

The attributes within this group are used to build the Request Baseline workspace.

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Application Tier Threshold (msec) The response time threshold in the application tier in milliseconds. The valid format is a positive integer.

Auto Threshold Bad Projection (%) The bad response time projection used for auto threshold. The valid format is a positive integer.

Auto Threshold Fair Ratio The percentage to derive the fair response time threshold from the baseline selection. The valid format is a positive integer.

Auto Threshold Fair Projection (%) The fair response time projection used for auto threshold. The valid format is a positive integer.

Auto Threshold Mode The request auto threshold mode. Valid values are Default, Custom, and Disabled.

Auto Threshold Deviation (%) The maximum allowed deviation of requests baseline data used for auto threshold. The valid format is a positive integer.

Auto Threshold Percent (%) The minimum percent of requests baseline data used for auto threshold. The valid format is a positive integer.

Backend Tier Threshold (msec) The response time threshold in the backend tier in milliseconds. The valid format is a positive integer.

Bad Response Threshold (msec) The time (in milliseconds) that defines the bad requests. A request that spends more time than this threshold to complete is a bad request. Use this attribute with `Fair Response Threshold (msec)` attribute and `Fair Response Zone (msec)` attribute. The valid format is a positive integer.

Bad Errors Rate Threshold The value of bad error rate percentage. The valid format is a positive integer.

Baselined Request Count The total number of requests accumulated in the baseline. This counter shows the data since the baseline starts. The valid format is a positive integer.

Client Tier Threshold (msec) The response time threshold in the client tier in milliseconds. The valid format is a positive integer.

Fair Response Threshold (msec) The time (in milliseconds) that defines the fair requests. A request that spends less time than this threshold to complete is a good request. Use this attribute with `Fair Response Zone (msec)` attribute and `Bad Response Threshold (msec)` attribute. The valid format is a positive integer.

Fair Response Zone (msec) The time span (in milliseconds) that defines the fair requests. This time span is between the fair response time threshold and the bad response time threshold. If the response time of a request falls into this time span, the request is a fair request. Use this attribute with `Fair Response Threshold (msec)` attribute and `Bad Response Threshold (msec)` attribute. The valid format is a positive integer.

Fair Errors Rate Threshold The value of fair error rate percentage. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Detail The request detail name. The valid format is an alphanumeric string, with a maximum of 256 characters.

Request ID The unique identifier of the request that belongs to the application. The valid format is a positive integer.

Request Label A shortened version of Request Name, used to display the request name in the chart view. The valid format is an alphanumeric string, with a maximum of 24 characters.

Request Name The URL for servlet requests, or the fully qualified class name for EJBs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Request Type The type of request being run. Valid values are All, Servlet/JSP, EJB, Portal, SCA Target, Web Service Provider, and Web Service Client.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 56. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Request Times and Rates attributes

The **Request Times and Rates** attribute group provides information about historical request throughput and average response time for a particular application server.

The attributes within this group are used to build the WebSphere App Server workspace.

Application Name The name of the application to which the request belongs. The valid formats are as follows:

- An alphanumeric string, with a maximum of 256 characters.
- An empty string means that this sample is aggregated data for all applications

ASID The identifier (decimal) assigned to the address space running this servant region.

Average Load The average number of concurrent requests during the interval. The valid format is a decimal (formatted to three decimal places).

Average Request Completion Rate The average request completion rate (that is, the request throughput). If the sampling rate is less than 100%, this number is extrapolated to estimate 100% of completed requests. The valid format is a positive integer.

Average Request Response Time The average request response time, in milliseconds. The valid format is a positive integer.

Error Rate (%) The error rate of the request during the interval. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID Indicates the process ID of the JVM.

Request Count The number of requests processed during the current interval. The valid format is a positive integer.

Request Data Monitoring Level Indicates request data monitoring level for application.

Request Type The type of request being run. Valid values are Servlet, EJB_Method, Custom, All_Workloads, Unknown, Portlet, SCA Target, Web Service Provider, and Web Service Client.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 57. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of edge requests-such as servlets and JSPs-that were sampled during the interval. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total (ms) The total time used (in milliseconds) during the interval. The valid format is a decimal (formatted to three decimal places).

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request attributes

The **Selected Request** attribute group provides detailed information about requests for database (that is, JDBC), messaging (that is, JMS), or EIS (that is, J2C) services by transactions.

The attributes within this group are used to build these workspaces:

- Selected Request - Datasources
- Selected Request - JMS Queues
- Selected Request - Resource Adapters
- Selected Request - Portal Processing
- "Selected Request - Service Components workspace" on page 327

Important: The attributes within this attribute group contain meaningful values only if your site has set the request data monitoring level to Level2 to collect data on nested requests.

Activity Category The type of request. Valid values are n/a (not applicable), JDBC, JMS, JCA, and Unknown.

Activity Detail Detailed information about the activity performed by the selected request, for example, the SQL statement being processed. The valid format is an alphanumeric string, with a maximum of 128 characters.

Activity Label An abbreviated version of Activity Name, used to display the activity name in the chart view. The valid format is an alphanumeric string, with a maximum of 32 characters.

Activity Name The resource that the request is accessing, for example, the data source name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Activity Type The type of the resource being requested. Valid values are:

Table 58. Activity types

Type	Definition
n/a	not applicable
Servlet	A call to a doGet or doPost methods of a servlet
EJB_Method_Call	A call to a business method for an EJB class
Obtain_SQL_Connection_from_Datasource	A call to obtain a connection from a JDBC data source
SQL_Query	A Query request to a JDBC data source
SQL_Update	An Update request to a JDBC data source
SQL_Other	Any other request to a JDBC data source
JMS_Message_Browse	A call to browse a message from a JMS queue
JMS_Message_Get	A call to receive a message from a JMS queue (that is, a destructive get)
JMS_Message_Put	A call to put a message from a JMS queue
JMS_Publish_Message	A call to publish a publication to a JMS queue
JCA_CCI_Execute_interaction	A request by a JEE application to execute a JCA interaction (a JDBC, JMS, or other JCA-supported operation) against a backend system
JNDI_Lookup	A call to JNDI to build an InitialContext or to perform a lookup
Unknown	The activity type cannot be determined
Portlet_Processing	A call for portlet processing request
Portlet_Authorization	A call for portlet authorization request
Portal_Authentication	A call for portal authentication request
Portal_Model_Building	A call for portal page model building request
Portal_Page_Loading	A call for portal page loading request
Portal_Page_Rendering	A call for portal page rendering request
Portal_Legacy_Action	A call for portal legacy action request
Portal_Standard_Action	A call for portal standard action (JSR-88) request
SCA	A call for Service Component Architecture (SCA) request
Web Service Provider	A call to a web service provide
Web Service Client	A call to a web service client

ASID The identifier (decimal) assigned to the address space running this servant region.

Average Response (ms) The average time (in milliseconds) executing this request, per occurrence. The valid format is a decimal (formatted to one decimal place).

Delay Percent The percentage of execution time this activity consumed on average when processing this request. The valid format is a decimal (formatted to one decimal place).

Delay Percent Count The count of individual delays used for Delay Percent. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Longest Response (ms) The worst-case response time (in milliseconds) experienced by this request. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Occurrences The number of times this request was executing during the interval. The valid format is a positive integer.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Request Detail The URI for servlet requests, or the method name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Name The URL for servlet requests, or the fully qualified class name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Type The type of transaction being run. Valid values are Servlet, EJB_Method, Custom, All_Workloads, Unknown, and Portlet.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 59. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute

Table 59. Format of the 12-character timestamp (continued)

Character String	Meaning
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of edge requests-such as servlets and JSPs-that were sampled for nested requests during the interval. The valid format is a positive integer.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Indicates that this row is a summary row of statistical totals for all rows.

Total Time (ms) The total CPU time (in milliseconds) consumed by this request. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Servlet Sessions attributes

The **Servlet Sessions** attribute group provides information about servlet sessions.

A session is a series of requests to a servlet, originating from the same user at the same browser. Applications running in a Web container use Sessions to monitor the actions of individual users. The attributes within this group are used to build the Sessions workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Average Concurrently Active Sessions The average number of concurrently active sessions during the sampling interval. A session is active if WebSphere application server is currently processing a request that uses the session. The valid format is a

decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Concurrently Live Sessions The average number of sessions cached in memory during the sampling interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Serializable Session Object Size (bytes) The average size (in bytes) of the serializable attributes of in-memory sessions. This number is at a session level only and includes only session objects that contain at least one serializable attribute object (a session can contain some attributes that are serializable and some that are not). This value is a measurement of the data at the end of the PMI sampling interval, not of the data in the entire sampling interval. The valid format is a positive integer.

Average Session Lifetime (ms) The average session lifetime (in milliseconds), calculated by subtracting the time the session was created from the time it was invalidated. The valid format is a decimal (formatted to three decimal places).

Session Lifetime Count The count of times that were averaged for Average Session Lifetime . The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Broken Session Affinities The number of HTTP session affinities that broke, not counting WebSphere application server intentional breaks of session affinity. This is the number of requests received for sessions that were last accessed from another web application and can indicate failover processing or a corrupted plug-in configuration. The valid format is a positive integer.

Broken Session Affinity Rate (per sec) The rate (per second) of the number of HTTP session affinities that break, not counting the WebSphere application server intentional breaks of session affinity, during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Cache Discard Rate (per sec) The rate (per second) at which session objects have been forced out of the cache during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Cache Discards The number of session objects that have been forced out of the cache. An LRU algorithm removes old entries to make room for new sessions and cache misses (this applies only to persistent sessions). The valid format is a positive integer.

Enterprise Application Name The name of the Enterprise application running the servlet. The valid format is an alphanumeric string, with a maximum of 256 characters.

External Read Size (bytes) The size (in bytes) of the session data read from the persistent store (applicable only to serialized, persistent sessions). The valid format is a decimal (formatted to three decimal places).

External Read Time (ms) The time (in milliseconds) taken to read the session data from the persistent store (applicable only to persistent sessions). For multirow sessions, the metrics are for the attributes; for single-row sessions, the metrics are for the whole session. When using a JMS persistent store, the user has the choice of whether to serialize the data being replicated; if the data are not serialized, this counter is not available. The valid format is a decimal (formatted to three decimal places).

External Write Size (bytes) The size (in bytes) of session data written to the persistent store (applicable only to serialized, persistent sessions). The valid format is a decimal (formatted to three decimal places).

External Write Time (ms) The time (in milliseconds) taken to write the session data from the persistent store (applicable only to serialized, persistent sessions). The valid format is a decimal (formatted to three decimal places).

Failed Session Request Rate (per sec) The rate (per second) that a request for a new session could not be handled because it would exceed the maximum session count for the sampling interval. The valid format is a decimal (formatted to three decimal places).

Failed Session Requests This attribute collects data from the NoRoomForNewSessionCount metric in the Servlet Sessions Counters PMI module. The number of times a request for a new session could not be handled because it would exceed the maximum session count; this applies only to a session in memory with AllowOverflow=false. The valid format is a positive integer.

Instrumentation Level The web instrumentation level for this web application. For WebSphere 5, the valid values are None, Low, Medium, High, and Maximum; for WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Max Serializable Session Object Size (bytes) The maximum size (in bytes) of the serializable attributes of in-memory sessions. This number is at a session level only and includes only session objects that contain at least one serializable attribute object (a session can contain some attributes that are serializable and some that are not). This value is a measurement of the data at the end of the PMI sampling interval, not of the data in the entire sampling interval. The valid format is a positive integer.

Min Serializable Session Object Size (bytes) The minimum size (in bytes) of the serializable attributes of in-memory sessions. This number is at a session level and includes only session objects that contain at least one serializable attribute object (a session can contain some attributes that are serializable and some that are not). This value is a measurement of the data at the end of the PMI sampling interval, not of the data in the entire sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Nonexistent Session Request Rate (per sec) The rate (per second) of requests for a session that no longer exists (presumably because the session timed out) during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Nonexistent Session Requests The number of requests for a session that no longer exists (presumably because the session timed out). Use this counter to determine if the timeout is too short. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 60. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Session Creation Rate (per sec) The rate (per second) of sessions created during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Session Invalidation Rate (per sec) The rate (per second) at which sessions were invalidated during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Sessions Created The number of sessions created during the sampling interval. The valid format is a positive integer.

Sessions Invalidated The number of sessions invalidated during the sampling interval. The valid format is a positive integer.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of Servlet Sessions Whether this row is a summary row of statistical totals collected for the servlet sessions. The valid values are No and Yes.

Time since Last Activated The time difference (in the locale-specific format, for example hh:mm:ss.msecs) between the previous and current access timestamps. Does not include session timeouts. The valid format is a timestamp.

Total Serializable Session Object Size (bytes) The total size (in bytes) of all the in-memory session objects. This includes only the serializable attributes in the session object; at least one such attribute must be present to be included in this total. This value is a measurement of the data at the end of the PMI sampling interval, not of the data in the entire sampling interval. The valid format is a positive integer.

Serializable Session Object Size Count The count of object sizes that were used for Total Serializable Session Object Size. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Web Application Archive The name of the web application WAR file. The valid format is an alphanumeric string, with a maximum of 128 characters.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Servlets JSPs attributes

The **Servlets JSPs** attributes provide performance information for servlets and Java server pages (JSPs).

Examples of Servlets JSPs attributes include the average number of concurrent requests for a servlet and the amount of time it takes for a servlet to complete a request. Use the Servlets JSPs attributes in situations to monitor performance and the usage of servlets and JSPs.

The attributes within this group are used to build the Servlets/JSPs - Selected Web Application workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Average Concurrent Requests The average number of concurrent requests for the servlet or JSP during the interval. The valid format is a decimal (formatted to three

decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Response Time (ms) The time (in milliseconds) it took the servlet to perform a task during the interval. The valid format is a decimal (formatted to three decimal places).

Response Time Count The count of times that were averaged for Average Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Enterprise Application Name The name of the Enterprise application. The valid format is an alphanumeric string, with a maximum of 128 characters.

Error Count The number of errors or exceptions that have occurred in the servlet during the interval. The valid format is a positive integer.

Error Rate (per sec) The servlet exceptions or errors (per second) since the previous sample. The valid format is a decimal (formatted to three decimal places).

Instrumentation Level The instrumentation level for this servlet. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Count The number of total requests for the servlet during the interval. The valid format is a positive integer.

Request Rate (per sec) The servlet requests (per second) since the previous sample. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 61. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Virtual Host Name The name of the virtual host. The valid format is an alphanumeric string, with a maximum of 64 characters.

Servlet/JSP Name The name of the current servlet or JSP. The valid format is an alphanumeric string, with a maximum of 128 characters.

Date and Time the Servlet Was Loaded The date and time when the servlet was loaded.

JSP Path Name The name of the JSP path. The valid format is an alphanumeric string, with a maximum of 128 characters.

Total (ms) The total time (in milliseconds) used during the interval. The valid format is a decimal (formatted to three decimal places).

Type Whether this entry represents a servlet or Java server page (JSP). The valid values are Servlet and JSP.

Web Application Archive The name of the web application WAR file. The valid format is an alphanumeric string, with a maximum of 128 characters.

Set Instrumentation Level Type The WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection. Valid values are: ORBThreadPool, HTTPSession, DBPool, Servlet, Transaction, ObjectPool, EJB, EJBMethod, AllResources, JVMMemory, ThreadPool, WCThreadPool, WebApplicationSubModule, DynamicCache, J2CPool, ORB, WLM, ServletSessions.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Scheduler attributes

The **Scheduler** attributes display data for the Scheduler service.

The attributes within this group are used to build the Scheduler workspace.

Instrumentation Level The instrumentation level for the Scheduler. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Poll Count The number of polls which are collected on the intervals. The valid format is a positive integer.

Poll Duration The average alarms during the latency. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Poll Query Duration The duration of poll query. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Row Number The row number. The valid format is a positive integer.

Run Duration The run duration. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 62. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Scheduler Name The name of the scheduler. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Task Collision Rate The rate of the task collision. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Task Delay Duration The duration of the task delay. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Task Expiration Rate The rate of the task expiration. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Task Failure Count The number of the failed tasks. The valid format is a positive integer.

Task Finish Count The number of the finished tasks which are collected on the intervals. The valid format is a positive integer.

Task Finish Rate The rate of the finished tasks. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Task Run Rate The rate of the run tasks. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Service Component Elements attributes

The **Service Component Elements** attributes provide aggregated information about the performance data for all the service components and their elements.

The attributes within this group are used to build the Service Component Elements workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Average Response Time (ms) The average response time (in milliseconds) in the current interval. The valid format is a decimal (formatted to three decimal places).

Component The type of the component. The value format is an alphanumeric string, with a maximum of 256 characters.

Component Name The component name of the service. The value format is an alphanumeric string, with a maximum of 256 characters.

Element The type of the element. The valid format is a positive integer.

Element Name The name of the element. The value format is an alphanumeric string, with a maximum of 256 characters.

Error Rate (per sec) The computed error rate. The valid format is a decimal (formatted to three decimal places).

Failed Count The failed invocations. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the service component elements. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Count The number of requests during the interval. The valid format is a positive integer.

Request Rate (per sec) The rate of requests during the interval per second. The valid format is a decimal (formatted to three decimal places).

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 63. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Success Count The successful invocations. The valid format is a positive integer.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Total (ms) The total time used (in milliseconds) during the interval. The valid format is a decimal (formatted to three decimal places).

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Service Components attributes

The **Service Components** attributes provide aggregated information about the overview performance of the key service components.

The attributes within this group are used to build the Service Components workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros

until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Count The bad request count. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the service components. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Name The type of the component. Valid values are Service_Component_Architecture, Business Rules, Map, Mediation, Business State Machine, and Selector.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 64. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Thread Pools attributes

The **Thread Pools** attribute group provides information about the data source, as well as connection statistics, for database connection pools in a WebSphere application server. Use it to monitor pools activity and to spot potential throttling.

The attributes within this group are used to build the Pool Analysis and Thread Pools workspaces.

Important:

- The attributes within this attribute group contain zeros for performance data if your site set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Application ID Indicates JEE application ID.

Average Active Threads The average number of concurrently active threads during the sampling interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Free Threads The average number of free threads in the pool. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Pool Size The average number of threads in the pool. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Important: This value might exceed Maximum Pool Size in certain configurations where the pool is enabled to grow beyond the specified maximum size.

Instrumentation Level The instrumentation level for the thread pools. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Maximum Pool Size The configured maximum number of threads allowed in the pool. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Percent of Time Pool at Max The average percentage of time that all threads were in use during the sampling interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Percent Used Bad The bad percent of pool usage by application. The valid format is a positive integer.

Percent Used Fair The fair percent of pool usage by application. The valid format is a positive integer.

Percent Used Good The good percent of pool usage by application. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 65. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Summary of Thread Pools Whether this row is a summary row of statistical totals collected for all thread pools. The valid values are No and Yes.

Thread Creation Rate (per sec) The rate (per second) at which threads were created during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Thread Destruction Rate (per sec) The rate (per second) at which threads were destroyed during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Thread Pool Name The name of the thread pool. The valid format is an alphanumeric string, with a maximum of 256 characters.

Threads Created The number of threads created during the sampling interval. The valid format is a positive integer.

Threads Destroyed The number of threads destroyed during the sampling interval. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Topic Spaces attributes

The **Topic Spaces** attributes provide aggregated information about publish/subscribe messaging.

The attributes within this group are used to build the Destinations workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Assured Persistent Local Subscription Hits The cumulative total of subscriptions which have matched assured persistent publications. The valid format is a positive integer.

Assured Persistent Messages Published The number of Assured Persistent messages published. The valid format is a positive integer.

Best Effort Non-persistent Local Subscription Hits The cumulative total of subscriptions which have matched best effort non-persistent publications. The valid format is a positive integer.

Best Effort Non-persistent Messages Published The number of best effort non-persistent messages published. The valid format is a positive integer.

Durable Local Subscription The number of durable subscriptions. The valid format is a positive integer.

Express Non-persistent Local Subscription Hits The cumulative total of subscriptions which have matched express non-persistent publications. The valid format is a positive integer.

Express Non-persistent Messages Published The number of express non-persistent messages published. The valid format is a positive integer.

Incomplete Publication The number of publications not yet received by all current subscribers. If this number is unexpected, view the publication using the admin console to take actions. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the topic spaces. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Local Oldest Publication The longest time any publication has spent on this topic space. If this time is not what was expected, view the message using the admin console to decide what action needs to be taken. The valid format is a positive integer.

Local Publisher The number of local publishers to topics in this topic space. The valid format is a positive integer.

Local Publisher Attaches The number of times an attachment has been made to this topic space by local producers. The lifetime of this value is the lifetime of the messaging engine. The valid format is a positive integer.

Messaging Engine Name The name of the message engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Non-Durable Local Subscription The number of non-durable subscriptions. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Reliable Non-persistent Local Subscription Hits The cumulative total of subscriptions which have matched reliable non-persistent publications. The valid format is a positive integer.

Reliable Non-persistent Messages Published The number of reliable non-persistent messages published. The valid format is a positive integer.

Reliable Persistent Local Subscription Hits The cumulative total of subscriptions which have matched reliable persistent publications. The valid format is a positive integer.

Reliable Persistent Messages Published The number of reliable persistent messages published. The valid format is a positive integer.

Report Enabled Publication Expired The number of report-enabled incomplete publications that expired while on this topic space. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 66. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Topic Space Name The name of the topic space. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Local Subscription The number of local subscriptions to topics in this topic space. The valid format is a positive integer.

Total Local Subscription Hits The cumulative total of subscriptions which have matched topic space publications. The valid format is a positive integer.

Total Messages Published The total number of publications to this topic space. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Applications attributes

Use the **Web Applications** attributes to create situations that monitor web application performance and application server loads.

The Web Applications attributes provide aggregated information for each web application and for the application server running that application. These performance data describe all servlets and JSPs deployed to that web application as well as performance data for all servlets and JSPs running in the application server. Examples include the number of loaded servlets and JSPs and total requests. The attributes within this group are used to build the Web Applications workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Average Concurrent Requests The average number of concurrent requests for servlets and JSPs during the interval. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Average Response Time (ms) The average time (in milliseconds) required for a servlet to perform a task during the interval. Calculated by dividing the total responses by Request Count; if Request Count is zero, this value is set to blank. The valid format is a decimal (formatted to three decimal places).

Virtual Host Name The name of the virtual host. The valid format is an alphanumeric string, with a maximum of 64 characters.

Enterprise Application Name The name of the Enterprise application. The valid format is an alphanumeric string, with a maximum of 128 characters.

Error Count The number of errors or exceptions that have occurred in the servlet. The valid format is a positive integer.

Error Rate (per sec) The servlet exceptions or errors (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Instrumentation Level The web instrumentation level for this web application. For WebSphere 5, the valid values are None, Low, Medium, High, and Maximum. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Count The number of total requests for the servlet during the interval. The valid format is a positive integer.

Request Rate (per sec) The servlet requests (per second) for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 67. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlets Loaded The number of servlets loaded during the interval. The valid format is a positive integer.

Servlets Reloaded The number of servlets reloaded during the interval. The valid format a positive integer.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Subinstrumentation Level The subinstrumentation level for the web application servlet submodule. For WebSphere application server 5 the valid values are None, Low, Medium, High, and Maximum.

Note: Subinstrumentation Level is not supported in WebSphere application server 6.0 or higher. The Tivoli Enterprise Monitoring agent uses the Instrumentation Level instead.

Summary of All Applications Whether this row is a summary row of statistical totals for all web applications executed during the interval. The valid values are Yes and No.

Total (ms) The total time used during the interval. The valid format is a decimal (formatted to three decimal places).

Response Time Count The number of calls for which the response times are averaged in the Average Response Time field. The valid format is a 4-byte integer. The value -2 is used if the number is very large and can not be provided.

Web Application Archive The name of the web application WAR file. The valid format is an alphanumeric string, with a maximum of 128 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Services attributes

The **Web Services** attributes display aggregated information about the web services, including the number of loaded web services, the number of requests delivered and processed, the request response time, and the average size of requests.

The attributes within this group are used to build the Web Services workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Dispatched Requests The number of requests the service dispatched or delivered. The valid format is a positive integer.

Dispatch Response Time The average response time, in milliseconds, to dispatch a request. The valid format is a decimal (formatted to three decimal places).

Dispatch Response Time Count The count of times that were averaged for Dispatch Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Instrumentation Level The instrumentation level for the web services counters. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Payload Size The average payload size in bytes of a received request or reply. The valid format is a positive integer.

Payload Size Count The count of sizes that were averaged for Payload Size. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Processed Requests The number of requests the service successfully processed. Valid format is a positive integer.

Received Requests The number of requests the service received. The valid format is a positive integer.

Reply Payload Size The average payload size (in bytes) of a reply. The valid format is a positive integer.

Reply Payload Size Count The count of sizes that were averaged for Reply Payload Size. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Reply Response Time The average response time, in milliseconds, to prepare a reply after dispatch. The valid format is a decimal (formatted to three decimal places).

Reply Response Time Count The count of times that were averaged for Reply Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Request Payload Size The average payload size, in bytes, of a request. The valid format is a positive integer.

Request Payload Size Count The count of sizes that were averaged for Request Payload Size. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Request Response Time The average response time, in milliseconds, to prepare a request for dispatch. The valid format is a decimal (formatted to three decimal places).

Request Response Time Count The count of times that were averaged for Request Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Response Time The average response time (in milliseconds) for a successful request. The valid format is a decimal (formatted to three decimal places).

Response Time Count The count of times that were averaged for Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 68. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Total Dispatch (ms) The total time (in milliseconds) the service dispatches requests. The valid format is a decimal (formatted to three decimal places).

Total Prepare (ms) The total time (in milliseconds) the service prepares requests. The valid format is a decimal (formatted to three decimal places).

Total Processing (ms) The total time (in milliseconds) the service processes requests. The valid format is a decimal (formatted to three decimal places).

Web Service The name of the web service. The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Services Gate Way attributes

The **Web Services Gate Way** attributes display aggregated information about the Web Services Gateway, including synchronous requests, asynchronous requests, synchronous responses, and asynchronous responses.

The attributes within this group are used to build the Web Services workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling; thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Asynchronous Requests The number of asynchronous requests made. The valid format is a positive integer.

Asynchronous Responses The number of asynchronous responses made. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the Web Services Gateway counters. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and

SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 69. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary Whether this row is a summary row of statistical totals for all rows. Valid values are No and Yes.

Synchronous Requests The number of synchronous requests made. The valid format is a positive integer.

Synchronous Responses The number of synchronous responses made. The valid format is a positive integer.

Web Service The name of the web service. The valid format is an alphanumeric string, with a maximum of 256 characters.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WebSphere Agent Events attributes

The **WebSphere Agent Events** attributes provide information about agent-level events that affect the ability of the Tivoli Enterprise Monitoring Agent to collect data about WebSphere application server. These attributes provide exception and error messages, their IDs, and their severities.

The attributes within this group are used to build the WebSphere Agent workspace.

Event Date and Time The date and time the event occurred. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 70. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Message Description The message description. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message ID The message ID. The valid format is an alphanumeric string, with a maximum of eight characters.

Node Name The system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sequence Number The sequence number of the message. The valid format is a positive integer.

Severity The severity of the event. Valid values are Info, Warning, Error, and Severe.

Function The valid format is an alphanumeric string, with a maximum of 32 characters.

File Name The name of the file. The valid format is an alphanumeric string, with a maximum of 8 characters.

Line Number The number of the line. The valid format is a 4-byte integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WMQ Client Link Communications attributes

The **WMQ Client Link Communications** attributes display aggregated information for all the clients of WMQ Queue Managers that are or have been connected to this application server.

The attributes within this group are used to build the WMQ Client Link Communications workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and reselect it later. Each time you reselect the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

API Calls Serviced The number of MQ API call requests serviced on behalf of WMQ JMS clients. The valid format is a positive integer.

Batches Sent The number of batches of messages sent to network attached WMQ JMS clients. The valid format is a positive integer.

Clients Attached The current number of WMQ JMS clients attached to this application server. The valid format is a positive integer.

Comms Errors The number of errors that have caused connections to WMQ JMS clients to be dropped. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the WMQ client link communications. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Messages Received The number of messages received from network attached WMQ JMS clients. The valid format is a positive integer.

Messages Sent The number of messages sent to network attached WMQ JMS clients. The valid format is a positive integer.

Messaging Engine Name The name of the message engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Reads Blocked The number of read operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with WMQ JMS clients. The valid format is a positive integer.

Received (bytes) The number of bytes of data received from network attached WMQ JMS clients. This includes bytes of message data as well as bytes of data used to control the flow of messages. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 71. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sent (bytes) The number of bytes of data sent to network attached WMQ JMS clients. This includes bytes of message data as well as bytes of data used to control the flow of messages. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Writes Blocked The number of write operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with WMQ JMS clients. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WMQ Link Communications attributes

The **WMQ Link Communications** attributes display aggregated information for all the WMQ Queue Managers that are or have been connected to this application server.

The attributes within this group are used to build the WMQ Link Communications workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.
- Attributes in this group are only provided for IBM WebSphere application server version 6.0 or later.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Batches Received The number of batches of messages received from network attached WMQ Queue Managers. The valid format is a positive integer.

Batches Sent The number of batches of messages sent to network attached WMQ Queue Managers. The valid format is a positive integer.

Comms Errors The number of communication errors that resulted in a network connection to a WMQ Queue Manager being disconnected. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the WMQ link communications. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Long Retries The number of long retries. This indicates the number of times channels were disconnected and could not be re-established for longer periods of time. The valid format is a positive integer.

Messages Received The number of messages received from network attached WMQ Queue Managers. The valid format is a positive integer.

Messages Sent The number of messages sent to network attached WMQ Queue Managers. The valid format is a positive integer.

Messaging Engine Name The name of the message engine. The value format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

QM Attached The total number of WMQ Queue Managers currently network attached to this application server. The valid format is a positive integer.

Reads Blocked The number of read operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with WMQ Queue Managers. The valid format is a positive integer.

Receiver Received (bytes) The number of bytes of data received by receiver channels from network attached WMQ Queue Managers. The valid format is a positive integer.

Receiver Sent (bytes) The number of bytes data sent by receiver channels to network attached WMQ Queue Managers. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 72. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sender Received (bytes) The number of bytes of data received by sender channels from network attached WMQ Queue Managers. The valid format is a positive integer.

Sender Sent (bytes) The number of bytes of data sent by sender channels to network attached WMQ Queue Managers. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Short Retries The number of short retries. This indicates the number of times channels were disconnected and could not be re-established for short periods of time. The valid format is a positive integer.

Writes Blocked The number of write operations that could not be completed immediately. This number can be used as an indicator of network congestion when communicating with WMQ Queue Managers. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workload Management Client attributes

The **Workload Management Client** attribute group provides information about the client that initiates workload requests.

Workload management (WLM) optimizes the distribution of client processing tasks. Incoming work requests are distributed to the application servers, enterprise beans, servlets, and other objects that can most effectively process their requests. Workload management also provides failover when servers are not available, improving application availability. In a WebSphere application server environment, you implement workload management by using clusters, transports, and replication domains.

The attributes within this group are used to build the Workload Management workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Client Cluster Update Rate (per sec) The rate (per second) at which times this client has received new server cluster information during the sampling interval. Use this metric to determine how often cluster information is being propagated. The valid format is a decimal (formatted to three decimal places).

Client Cluster Updates The number of times initial or updated server cluster data is sent to a WLM-enabled client from a server cluster member. Use this metric to determine how often cluster information is being propagated. The valid format is a positive integer.

Client Response Time (ms) The response time (in milliseconds) for IIOP requests sent by a client. This response time is calculated based on the time the client sends the request to the time the server sends the reply. The valid format is a decimal (formatted to three decimal places).

Client Response Time Count The count of times that were used for Client Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Instrumentation Level The instrumentation level for the WLM client. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Outgoing Request Rate (per sec) The rate (per second) at which outgoing IIOP requests were being sent from this client to an application server during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Outgoing Requests The number of outgoing IIOP requests being sent from this client to an application server. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 73. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

Total Client Response (ms) The total response time (in milliseconds) for IIOP requests sent by a client. The valid format is a decimal (formatted to three decimal places).

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workload Management Server attributes

The **Workload Management Server** attribute group provides information about the workload management server.

Workload management (WLM) optimizes the distribution of client processing tasks. Incoming work requests are distributed to the application servers, enterprise beans, servlets, and other objects that can most effectively process those requests. Workload management also provides failover when servers are not available, improving application availability. In a WebSphere application server environment, you implement workload management by using clusters, transports, and replication domains.

The attributes within this group are used to build the Workload Management workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Concurrent Requests The number of remote IIOP requests being processed by this server. The valid format is a decimal (formatted to three decimal places). This value is an average of several values collected over an interval. The interval can be either between this time and the previous time you activated this workspace, or fixed (normally 60 seconds), depending on the configuration of the Monitoring Agent.

Incoming Non-WLM Object Request Rate (per sec) The rate (per second) of incoming IIOP requests from an application running on a non-WLM client during the sampling interval. This type of client either does not have the WLM runtime present, or the client object reference was flagged not to participate in workload management. The valid format is a decimal (formatted to three decimal places).

Incoming Non-WLM Object Requests The number of incoming IIOP requests to an application from a client that does not have the WLM runtime present or whose object reference was flagged not to participate in workload management. The valid format is a positive integer.

Incoming Nonaffinity Request Rate (per sec) The rate (per second) of incoming IIOP requests to an application server based on no affinity during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Incoming Nonaffinity Requests The number of incoming IIOP requests to an application server based on no affinity. This request was sent to this server based

on workload management selection policies that were decided in the client WLM run time. The valid format is a positive integer.

Incoming Request Rate (per sec) The rate (per second) of incoming IIOP requests to an application server during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Incoming Requests The number of incoming IIOP requests to an application server during the sampling interval. The valid format is a positive integer.

Incoming Strong Affinity Request Rate (per sec) The rate (per second) of incoming IIOP requests to an application server that are based on a strong affinity during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Incoming Strong Affinity Requests The number of incoming IIOP requests to an application server that are based on a strong affinity. A strong affinity request is one that must be serviced by this application server because of a dependency that resides on the server. This request could not successfully be serviced by another member of the server cluster. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the WLM server. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 74. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Cluster Update Rate (per sec) The rate (per second) at which this server received new server cluster information during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Server Cluster Updates The number of times initial or updated server cluster data are sent to a server member from the deployment manager. This metric determines how often cluster information is being propagated. The valid format is a positive integer.

Server Name The name of the WebSphere application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Response Time (ms) The response time (in milliseconds) at which IIOP requests were serviced by an application server, calculated based on the time the request is received versus the time the reply is sent. The valid format is a decimal (formatted to three decimal places).

Server Response Time Count The count of times that were used for Server Response Time. The valid format is a 4-byte integer. If the number is too big to be stored, a value of -2 is used.

Total Server Response (ms) The total response time (in milliseconds) at which IIOP requests were serviced by an application server. The valid format is a decimal (formatted to three decimal places).

Set Instrumentation Level Type Indicates the WebSphere resource category, which is used by the agent to modify the Instrumentation Level for transaction data collection.

WLM Clients Serviced The number of WLM-enabled clients this application server has serviced during the interval. The valid format is a positive integer.

WLM Clients Serviced Rate (per sec) The rate (per second) at which this server has serviced WLM-enabled clients during the sampling interval. The valid format is a decimal (formatted to three decimal places).

WLM Server Weight A control for work directed to the application server. If the server weight value is greater than the weight values assigned to other servers in the cluster, then the server receives a larger share of the cluster workload. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workplace Mail IMAP/POP attributes

The **Workplace Mail IMAP/POP** attributes displays the usage information for the IMAP service and POP3 service connectivity.

The attributes within this group are used to build the IMAP/POP workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and then select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Active Sessions The number of active sessions during the sampling interval. The valid format is a positive integer.

Active SSL Sessions The number of active, secure sessions during the sampling interval. The valid format is a positive integer.

Authentication Failures The number of authentications failures during the sampling interval. The valid format is a positive integer.

Connection (ms) The time (in milliseconds) spent connected to clients during the sampling interval. The valid format is a positive integer.

IMAP Instrumentation Level The instrumentation level for IMAP PMI module. Valid values are None, Low, Medium, High, Basic, Extended, All, Custom, and Maximum. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Peak Session load The maximum number of concurrent sessions during the sampling interval. The valid format is a positive integer.

POP Instrumentation Level The instrumentation level for POP PMI module. Valid values are None, Low, Medium, High, Basic, Extended, All, Custom, and Maximum. Blank if no instrumentation level is set.

Protocol The protocol type of the workplace mail. Valid values are IMAP and POP.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and

SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 75. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Successful Authentications The number of successful authentications during the sampling interval. The valid format is a positive integer.

Instrumentation Level The Instrumentation Level for this container. Valid fixed values None, Low, Medium, High, Basic, Extended, All, Custom, Maximum.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workplace Mail Queues attributes

The **Workplace Mail Queues** attributes display information about the message delivery state, including ready, retry, unprocessed, and dead.

The attributes within this group are used to build the Messages Queues workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and select again it later. Each time you select the workspace, these attributes are updated with the latest data.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Dead The number of messages in the dead state in the queue during the sampling interval. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the service components. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Queue The Lotus Workplace Mail queue name. Valid values are A, B, C, D, E, F, G, H, and Summary.

Ready The number of messages in the ready state in the queue during the sampling interval. The valid format is a positive integer.

Retry The number of messages in the retry state in the queue during the sampling interval. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 76. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Unprocessed The number of messages in the unprocessed state in the queue during the sampling interval. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Workplace Mail Service attributes

The **Workplace Mail Service** attributes display aggregated usage information about the incoming mail message traffic.

The attributes within this group are used to build the Workplace Mail workspace.

Important:

- The attributes within this attribute group contain zeros for performance data if your site has set configuration value Resource Data Collection Method to On Demand (for on-demand sampling) and if you have not yet run applications that generate performance data. To report performance data in these attributes after installing and configuring the data collector, use the WebSphere administrative console to set the appropriate PMI instrumentation level.
- The Tivoli Enterprise Monitoring Agent is set by default to provide on-demand sampling. Thus the attributes within this attribute group initially contain zeros until you select the workspace and select it again later. Each time you select the workspace, these attributes are updated with the latest data.

Important: This workspace does not support linking to ITCAM for Application Diagnostics Managing Server visualization engine.

Tip: For information about WebSphere PMI metrics, see "WebSphere PMI Attribute Mapping" on page 583.

Active LDAP Connections The number of active LDAP connections during the sampling interval. The valid format is a positive integer.

Delivered Messages The total number of delivered messages during the sampling interval. The valid format is a positive integer.

Deliverer Dropped Messages The total number of messages rejected by the SMTP outbound server during the sampling interval. The valid format is a positive integer.

Deliverer Message (ms) The total time in milliseconds taken by SMTP outbound server to process messages during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Dropped SMTP Connections The total number of the dropped SMTP connections during the sampling interval. The valid format is a positive integer.

DSN Failure Messages The total number of failure DSNs sent during the sampling interval. The valid format is a positive integer.

Handled Messages The total number of messages processed by the mail handler server during the sampling interval. The valid format is a positive integer.

Handler Dropped Messages The total number of messages rejected by the mail handler server during the sampling interval. The valid format is a positive integer.

Instrumentation Level The instrumentation level for the service components. For WebSphere 6 or higher, the valid values are None, Basic, Extended, All, Custom. Blank if no instrumentation level is set.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Live SMTP Server Connections The number of live SMTP server connections during the sampling interval. The valid format is a positive integer.

Live SMTP Client Connections The number of live SMTP client connections during the sampling interval. The valid format is a positive integer.

Lost SMTP Client Connections The total number of lost SMTP client connections during the sampling interval. The valid format is a positive integer.

Lost SMTP Connections The total number of the lost SMTP connections during the sampling interval. The valid format is a positive integer.

Message Delivery (ms) The total time in milliseconds taken to deliver messages during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Messages Handling (ms) The total time in milliseconds taken to handle messages during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Peak LDAP Connections The maximum number of concurrent LDAP connections during the sampling interval. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 77. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute

Table 77. Format of the 12-character timestamp (continued)

Character String	Meaning
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Smarthost Messages The total number of messages sent to the Smarthost during the sampling interval. The valid format is a positive integer.

SMTP Client Connections The total number of SMTP client connections during the sampling interval. The valid format is a positive integer.

SMTP Client (ms) The total time in milliseconds taken to deliver messages during the sampling interval. The valid format is a decimal (formatted to three decimal places).

SMTP Connections The total number of connections to the SMTP server during the sampling interval. The valid format is a positive integer.

SMTP (ms) The total time in milliseconds that SMTP has conversed during the sampling interval. The valid format is a decimal (formatted to three decimal places).

SMTP Server Threads The number of active SMTP outbound server threads during the sampling interval. The valid format is a positive integer.

For more information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Agent Configuration attributes

The attributes in the Agent Configuration table are for internal use only.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Id The row identifier. The valid format is a 2-byte integer.

View Id The view identifier. The valid format is a 2-byte integer.

Act Id The act identifier. The valid format is a 2-byte integer.

Data The data string. The valid format is an alphanumeric string, with a maximum of 4096 characters.

KYN Command attributes

The attributes in the KYN command table are for internal use only.

Origin Node The origin node. The valid format is an alphanumeric string, with a maximum of 128 characters.

Command The command string. The valid format is an alphanumeric string, with a maximum of 256 characters.

User ID The user ID. The valid format is an alphanumeric string, with a maximum of 16 characters.

Result Status The result status code. The valid format is a 4-byte integer.

Server Configuration attributes

The attributes in the Server Configuration table are for internal use only.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Id The row identifier. The valid format is a 2-byte integer.

View Id The view identifier. The valid format is a 2-byte integer.

Act Id The act identifier. The valid format is a 2-byte integer.

Data The data string. The valid format is an alphanumeric string, with a maximum of 4096 characters.

Situations for ITCAM Agent for WebSphere Applications

IBM Tivoli Composite Application Manager Agent for WebSphere Applications has a number of predefined situations that you can use to complete the following tasks:

- Monitor your WebSphere application servers
- Monitor and manage widely dispersed WebSphere application server resources through localized automation
- Create your own situations using the predefined situations as examples

These predefined situations display an alert status. When these situations trigger an alert, you can investigate the event by opening its workspace. For example, you can use these situations to monitor a WebSphere application server for errors occurring within it or web applications based at your site.

How the situations work

Situations are tests expressed in IF-TRUE format of system conditions that you want to monitor; the tested value is an ITCAM Agent for WebSphere Applications attribute expressed in the form *attribute-group.attribute-name*. If the specified condition occurs or exists, the situation is true, and an alert is issued.

Avoid using negative values

If you define situations that use a counter or a range of numbers, always provide a threshold or use values in a positive range of numbers. For example, use a greater-than-or-equal-to-zero expression as shown in some of the following predefined situations. This practice prevents a situation from falsely tripping. If the ITCAM Agent for WebSphere Applications Tivoli Enterprise Monitoring Agent encounters an undefined attribute value, it interprets this value as a negative number and erroneously fires a situation that specifies a negative number.

Important: For the following situations:**WASDBConPAvgWaitTimeHigh DB_Connection_Pools**,**J2C_Connection_Pools** and **Thread_PoolsApplication** the **Application ID** column is not supported and always displays -1 by default.

Accessing the situations

Some of the predefined situations run by default from the WebSphere Agent. For the situations that do not run automatically, you must start these situations manually. To start these situations, access the situations in the Tivoli Enterprise Portal using one of the following methods:

- In the WebSphere agent, right-click the specific server. Right-click **Enterprise** then, click **Manage Situations** to display all the managed situations available on the server. If you want to start, stop, edit, or model a situation, right-click the situation and select the option you want.
- From the toolbar on the main menu, click the Situation Editor icon  and scroll to the situation that you want to view.

For information about predefined situations and formulas see "Predefined situations-descriptions and formulas (that run automatically)" and "Predefined situations descriptions and formulas (that run manually)" on page 493. See also "Attributes for ITCAM Agent for WebSphere Applications" on page 349.

Predefined situations-descriptions and formulas (that run automatically)

The following predefined situations run automatically from the WebSphere Agent and support the following applications servers unless stated otherwise.

- WebSphere application server
- WebSphere application server portal
- WebSphere application server Process
- WebSphere application server ESB
- Lotus Workplace server

Tip:

- To start, stop, or edit any of these situations, see "Accessing the situations."
- For information about situations that run manually, see "Predefined situations descriptions and formulas (that run manually)" on page 493.

WASAppDiscovered monitors WebSphere applications deployed in the application server and issues an Informational alert when a new application is discovered. The monitoring agent checks for new applications each time it connects to the data collector or when an application is deployed when the data collector is already active. The formula is:

If

Application_Monitoring_Configuration.Monitoring_Status equals Discovered

then

the situation WASAppDiscovered is true.

Important: This situation does not support Lotus Workplace server.

The predefined **Take Action** command **Start_Baselining** associated with the WASAppDiscovered situation enables you to automate the baselining of newly discovered applications.

WASAppHealthBad monitors the overall application health and issues a Critical alert when the application health is bad. The formula is:

If

Application_Health_Status.Web_Tier_Health equals Bad

then

the situation WASAppHealthBad is true.

The predefined **Take Action** command **Set_Appliction_Monitoring** associated with WASAppHealthBad situation increases the request monitoring rate for applications generated alert. This command enables you to collect more detailed performance data and helps to collect the most precise data about each application tier health level.

WASAppHealthFair monitors the overall application health and issues a warning alert when application health is fair. The formula is:

If

Application_Health_Status.Application_Health equals Fair

then

the situation WASAppHealthFair is true.

The predefined **Take Action** command **Set_Appliction_Monitoring** associated with WASAppHealthFair situation raises the request monitoring level for applications generated alert. This command enables you to collect detailed performance data that helps to pinpoint a bottleneck down to particular application tiers.

WASAppHealthGood monitors the overall application health and issues an Informational alert when application health is good. The formula is:

If

Application_Health_Status.Application_Health equals Good

then

the situation WASAppHealthGood is true.

The predefined **Take Action** command **Set_Appliction_Monitoring** associated with the WASAppHealthGood situation lowers the request monitoring level for applications generated alert, and reduces the monitoring workload.

WASError monitors the error severity for a single WebSphere application server and issues a Critical condition whenever that severity is greater than 21. Its formula is:

If

Log_Analysis.Severity is greater than Warning

then

the situation WASError is true.

WASHighCPUPercentUsed monitors the percentage of the CPU being consumed and issues a Critical condition whenever that time exceeds 80%. The formula is:

If

Application_Server.CPU_Used_Percent is greater than 80

then

the situation WASHighCPUPercentUsed is true.

WASHighGCTimePercent monitors the percentage of time being spent by the garage collector and issues a Critical condition whenever that time exceeds 80%. The formula is:

If

Garbage_Collection_Analysis.Real_Time_Percent is greater than 80

then

the situation WASHighGCTimePercent is true.

WASHighResponseTime monitors the average request response time and issues a Critical condition whenever that time exceeds 2 seconds. The formula is:

If

Request_Times_and_Rates.Average_Request_Response_Time is greater than 2000

then

the situation WASHighResponseTime is true.

WASNotConnected monitors the connection between the ITCAM for Data Collector WebSphere running in an application server and the ITCAM Agent for WebSphere Applications monitoring agent to ensure that the monitoring agent is connected and issues a Critical condition whenever it is not. Its formula is:

If

Application_Server_Status.Status equals Disconnected

then

the situation WASNotConnected is true.

WASOutofHeapSpace monitors the heap allocation status and issues a Critical condition whenever heap space is exhausted. The formula is:

If

Allocation_Failure.Heap_Status equals Out of heap space or Insufficient space

then

the situation WASOutofHeapSpace is true.

Important: This situation is not available when monitoring non-IBM Java Virtual Machines, including machines commonly used on HP-UX and Solaris platforms.

WASAvgHeapSizeAfterGCHigh monitors the average heap size free percentage after garbage collection. This situation issues a Critical alert if the average heap size free percentage after garbage collection is greater than 80%. Its formula is:

If

Garbage_Collection_Cycle.Heap_Free_Percent_after_GC is greater than 80

then

the situation WASAvgHeapSizeAfterGCHigh is true.

Important: This situation does not support Lotus Workplace server.

WASJ2CConnectionPoolUsageMaxed monitors the J2C pool percentage usage and issues a Warning alert if the pool usage is greater than or equal to 100%. Its formula is:

If

J2C_Connection_Pools.Pool_Used_Percent is greater than or equal to 100%

then

the situation WASJ2CConnectionPoolUsageMaxed is true.

Important: This situation does not support Lotus Workplace server.

WASDBConnectionPoolUsageMaxed monitors the JDBC pool usage and issues a Critical alert if the pool usage is greater than or equal to 100%. Its formula is:

If

DB_Connection_Pools.Percent_Used is greater than or equal to 100%

then

the situation WASDBConnectionPoolUsageMaxed is true.

Important: This situation does not support Lotus Workplace server.

WASHungThreadsDetected monitors hung threads and issues a Critical alert if any thread is hung. Its formula is:

If

Application_Server.Hung_Threads_Total is greater than 0

then

the situation WASHungThreadsDetected is true.

WASListenerPortStarted monitors starting of the WAS listener port. Its formula is:

If

Listener_Port.Status equals Started

then

the situation WASListenerPortStarted is true.

WASListenerPortStopped monitors stopping of the WAS listener port and issues a Critical alert if the port is stopped. Its formula is:

If

Listener_Port.Status equals Stopped

then

the situation WASListenerPortStopped is true.

See also "Attributes for ITCAM Agent for WebSphere Applications" on page 349.

Predefined situations descriptions and formulas (that run manually)

Situations that are run manually

The following situations do not run automatically. To run them you must access them from the **Manage Situations** view. These situations support the following application servers unless stated otherwise:

- WebSphere application server
- WebSphere application server portal
- WebSphere application server Process
- WebSphere application server ESB
- Lotus Workplace server

Tip:

- To start, stop, or edit any of these situations, see "Accessing the situations" on page 489.
- For information about situations that run automatically, see "Predefined situations-descriptions and formulas (that run automatically)" on page 489.

WASDBConnectionPoolThrdTimeout monitors the thread timeout count. This situation issues a Critical condition whenever the timeout count is greater than zero. Its formula is:

If

DB_Connection_Pools.Threads_Timed_Out is greater than 0

then

the situation WASDBConnectionPoolThreadTimeout is true.

WASContainerTransactionRollback monitors the rollback count of the WebSphere application server. This situation issues a Critical alert whenever the count becomes nonzero. Its formula is:

If

Container_Transactions.Global_Transactions_Rolled_Back is greater than 0

or

Container_Transactions.Local_Transactions_Rolled_Back is greater than 0

then

the situation WASContainerTransactionRollBack is true.

WASEJBCreateTimeHigh monitors the average time of a bean create call and issues a Critical alert when the time is longer than 2 seconds. Its formula is:

If

Enterprise_Java_Beans.Create_Average_Time is greater than 2000

then

the situation WASEJBCreateTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASEJBRemoveTimeHigh monitors the average time of a bean remove call and issues a Critical alert when the time is longer than 2 seconds. Its formula is:

If

Enterprise_Java_Beans.Remove_Average_Time is greater than 2000

then

the situation WASEJBRemoveTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASEJBMethodResponseTimeHigh monitors the average response time on remote interface methods for all beans. This situation issues a Critical alert if the response time is longer than 2 seconds. Its formula is:

If

Enterprise_Java_Beans.Method_Average_Response_Time is greater than 2000

then

the situation WASEJBMethodResponseTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASThreadFreeLow monitors the average free threads in the pool and issues a Critical alert if the number of threads is less than 200. Its formula is:

If

Thread_Pools.Average_Free_Threads is greater than or equal to 0

and

Thread_Pools.Average_Free_Threads is less than 200

and

Thread_Pools.Average_Active_Threads is greater than or equal to 0

then

the situation WASThreadFreeLow is true.

Important: This situation does not support Lotus Workplace server.

WASDataSrcConWaitTimeHigh monitors the average time an application has to wait for a connection. This situation issues a Critical alert if the wait time is longer than 2 seconds. Its formula is:

If

Datasources_Connection_Average_Wait_Time is greater than 2000

then

the situation WASDataSrcConWaitTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASReqSQLExecuteTimePercentHigh monitors the percentage of time the request spends executing a JDBC database. This situation issues a Critical alert if the percentage of time is higher than 80%. Its formula is:

If

Request_Analysis.SQL_Execute_Time_Percent is greater than 80

then

the situation WASReqSQLExecuteTimePercentHigh is true.

Important: This situation does not support Lotus Workplace server.

WASReqSQLQueryTimePercentHigh monitors the percentage of time the request spends querying a JDBC database. This situation issues a Critical Alert if the percentage of time is higher than 80%. Its formula is:

If

Request_Analysis.SQL_Query_Time_Percent is greater than 80

then

the situation WASReqSQLQueryTimePercentHigh is true.

Important: This situation does not support Lotus Workplace server.

WASReqSQLUpdateTimePercentHigh monitors the percentage of time the request spends updating a JDBC database. This situation issues a Critical Alert if the percentage of time is higher than 80%. Its formula is:

If

Request_Analysis.SQL_Update_Time_Percent is greater than 80

then

the situation WASReqSQLUpdateTimePercentHigh is true.

Important: This situation does not support Lotus Workplace server.

WASDBConPAverageUsageTimeHigh monitors the average time that a connection in use is high. This situation issues a Critical alert if the average time the connection in use is longer than 2 seconds. Its formula is:

If

DB_Connection_Pools.Average_Usage_Time is greater than 2000

then

the situation WASDBConPAverageTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

If

DB_Connection_Pools.Percent_Used is greater than 80

then

the situation WASDBConPPercentUsedTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASDBConPAvgWaitTimeHigh monitors the average time that a client has to wait for a connection. This situation issues a Critical alert when the time period is longer than 2 seconds. Its formula is:

If

DB_Connection_Pools.Average_Wait_Time is greater than 2000

then

the situation WASDBConPAvgWaitTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASJ2CCPAverageUsageTimeHigh monitors the average time that connections are in use (it measures from when the connection is allocated to when it is returned). This situation issues a Critical alert when the combined connection allocation and return time are longer than 2 seconds. Its formula is:

If

J2C_Connection_Pools.Average_Usage_Time is greater than 2000

then

the situation WASJ2CCPAverageUsageTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASJ2CCPAvgWaitTimeHigh monitors the average wait time until a connection is granted. This situation issues a Critical alert if the time period is longer than 2 seconds. Its formula is:

If

J2C_Connection_Pools.Average_Wait_Time is greater than 2000

then

the situation WASJ2CCPAvgWaitTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASCTGlbTransDurationHigh monitors the average duration of global transactions. This situation issues a Critical alert if the time period is longer than 2 seconds. Its formula is:

If

Container_Transactions.Global_Transaction_Duration is greater than 2000

then

the situation WASCTGlbTransDurationHigh is true.

Important: This situation does not support Lotus Workplace server.

WASCTLclTransDurationHigh monitors the average duration of local transactions. This situation issues a Critical alert if the time period is longer than 2 seconds. Its formula is:

If

Container_Transactions.Local_Transaction_Duration is greater than 2000

then

the situation WASCTLclTransDurationHigh is true.

Important: This situation does not support Lotus Workplace server.

WASPortalPageResponseTime monitors the portal page response time and issues a Critical alert when the average request response time is higher than 2 seconds. The formula is:

If

Portal_Page_Summary.Average_Response_Time is greater than 2000

then

the situation WASPortalPageResponseTime is true.

WASPortletResponseTime monitors the portlet response time and issues a Critical alert when the average request response time is higher than 2 seconds. The formula is:

If

Portlet_Summary.Average_Response_Time is greater than 2000

then

the situation WASPortletResponseTime is true.

WASServletsJSPsError monitors the error count for servlets and JSPs invoked by a WebSphere application server application. This situation issues a Critical condition whenever the count becomes nonzero. Its formula is:

If

Servlets_JSPs.Error_Count is greater than 0

then

the situation WASServletsJSPsError is true.

Important: This situation does not support Lotus Workplace Server.

WASSrvlSessAvgActiveSessionHigh monitors the average number of concurrently active sessions. This situation issues a Critical alert if the average number of concurrently active sessions is greater than 100. Its formula is:

If

Servlet_Sessions.Average_Concurrently_Active_Sessions is greater than 100

then

the situation WASSrvlSessAvgActiveSessionHigh is true.

The following situations are configured to run automatically.

Important: This situation does not support Lotus Workplace server.

WASSrvlSessExtReadTimeHigh monitors the time it takes to read the session data from the persistent store. This situation issues a Critical alert if the time period is longer than 2 seconds. Its formula is:

If

Servlet_Sessions.External_Read_Time is greater than 2000

then

the situation WASSrvlSessExtReadTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASSrvlSessExtWriteTimeHigh monitors the time it takes to write session data to persistent store. This situation issues a Critical alert if the time period is longer than 2 seconds. Its formula is:

If

Servlet_Sessions.External_Write_Time is greater than 2000

then

the situation WASSrvlSessExtReadTimeHigh is true.

Important: This situation does not support Lotus Workplace server.

WASThreadPoolPercentMaxed monitors the average usage time of all threads, and issues a Critical condition whenever that time exceeds 100%. The formula is:

If

Thread_Pools.Percent_of_Time_Pool_at_Max is greater than 100

then

the situation WASThreadPoolPercentMaxed is true.

WASWebApplicationError monitors the error log status of the WebSphere server and issues a Critical condition when an error occurs. Its formula is:

If

Web_Applications.Error_Count is greater than 0

then

the situation WASWebApplicationError is true.

Important: This situation does not support Lotus Workplace Server.

See also "Attributes for ITCAM Agent for WebSphere Applications" on page 349.

Take Action commands for ITCAM Agent for WebSphere Applications

The Take Action feature lets your interactive Tivoli Enterprise Portal users enter a command or stop or start a process at any system in your network where one or more Tivoli Enterprise Monitoring Agents are installed. The ITCAM Agent for WebSphere Applications Take Action commands enable you use the Tivoli Enterprise Portal interface to start, stop, or recycle a WebSphere application server or to control the level of monitoring for the current server.

Users can start a Take Action command from a workspace, from the Navigator, from a situation that you create, on demand, or by recalling a saved Take Action command.

Note:

The following take action commands are for internal use only and are not for use in the Tivoli Enterprise Portal. The configuration workspaces use these take action commands to communicate internally with the monitoring agent.

- **`Configure`**
- **`ConfigureCancel`**
- **`ConfigurePing`**

Add_XD_Cell: Add an XD Cell to a WebSphere agent

Use the Add_XD_Cell command to add an XD cell to the WebSphere Agent. This take action task is used to configure XD Cell monitoring.

Command syntax

YN:AddXDCell *cellName*

where *cellName* is the name of the XD cell.

Enable_Auto_Threshold: set threshold parameters

Use the Enable_Auto_Threshold Take Action to set automatic threshold parameters and remove any overrides of the thresholds.

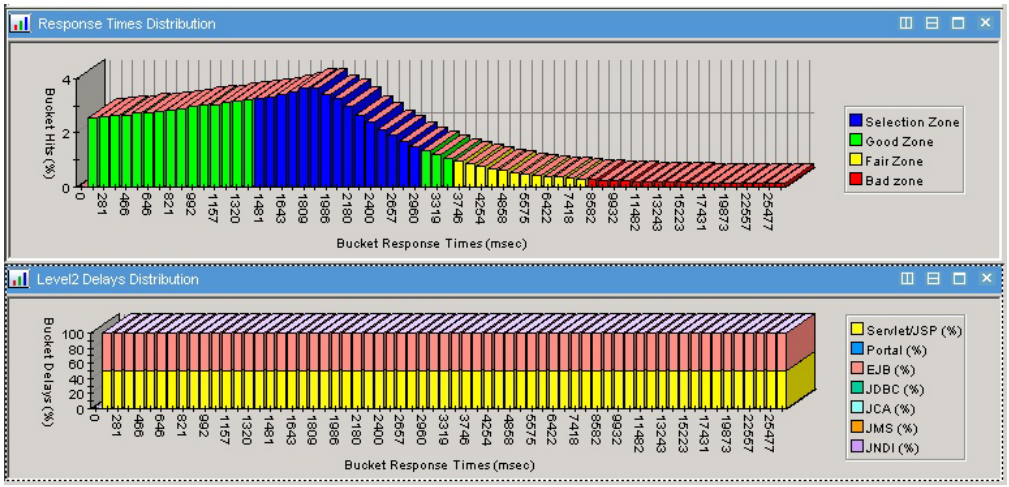
The baselining process supplies statistical information about request response times. ITCAM interprets this information to set automatic thresholds. Several parameters control this interpretation.

The default values for these parameters are sufficient for most cases. However, if the thresholds are not adequate and the baselining process was run recently, you might have to change these parameters. If there are many false alarms or warnings, you must raise the bad or fair threshold; if alarms or warnings are not triggered when needed, you must lower the bad or fair threshold.

Although you can change the parameters for the entire application or for all requests of a given type, most likely you do this only for an individual request.

- To set threshold parameters for all requests in the application, select this application in the "Application Health workspace" on page 283 or "Application Registry workspace" on page 285, and select the Enable_Auto_Threshold take action command.
- To set threshold parameters for all requests of a given type in the application, select this request type in the Application Request Configuration table of the "Application Configuration workspace" on page 321, and select the Enable_Auto_Threshold take action command.
- To set threshold parameters for an individual request, select this request in the "Request Baseline workspace" on page 319, and select the Enable_Auto_Threshold take action command.

In the "Request Baseline workspace" on page 319, when you select a line representing a request, you can see the bar charts representing statistical data for this request. This data was gathered during the baselining process. Colors on the bar charts show the way in which the parameters are applied. You can change the parameters using the Enable_Auto_Threshold take action command, and immediately see the effects on the bar charts.



The **Response Times Distribution** chart shows the statistical distribution of response times for this request. The height of every bar shows the percentage of requests that had the indicated response time during the baselining period.

Some bars represent bigger time intervals than others; more bars are devoted to most common response times. For example, if the maximum encountered time is 1000 ms but most response times are 300 - 500 ms, then the first bar can be 0 - 50 ms, but there can also be bars of 305 - 310 ms and 400 - 402 ms.

The bars colored blue show the zone into which the "typical" response times for this application fall. The green bars show response times that are not "typical", but

are below the fair threshold. Response times above the fair threshold but below the bad threshold are shown as yellow bars; for those above the bad threshold, the bars are red.

Use the Enable_Auto_Threshold take action command to set the parameters that affect both the position of the "typical" zone and the way the thresholds are derived from this zone.

For more information about how the bar chart and parameters work, see “Threshold calculation detail” on page 510.

The **Level2 Delays Distribution** chart shows the distribution of time spent in "nested requests" within the requests that had this response time range. Each bar represents a response time of the top-level request . Within this bar, colored sections show how much time is spent within nested requests of different types; the color legend is shown on the bar. ITCAM uses this distribution within the selection zone (that is for typical overall request types) to work out the average share of time that each nested request type takes. When an error or warning arises, ITCAM checks which of the request types takes more than its usual share of time; based on this, it displays whether the likely cause is the application, backend, or server.

Command syntax

YN:Enable_Auto_Threshold *App_Id Request_Id Auto_Threshold_Percent Auto_Threshold_Deviation Auto_Threshold_Fair_Projection Auto_Threshold_Bad_Projection Use_Default*

Parameters:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Request_Id
The request ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Auto_Threshold_Percent
Auto_Threshold_Deviation
ITCAM uses these two parameters to calculate the borders of the "typical zone". See “Threshold calculation detail” on page 510.

Auto_Threshold_Fair_Projection
This determines the position of the fair threshold. Increase this parameter to increase the fair threshold; decrease the parameter to decrease the fair threshold. For details, see “Threshold calculation detail” on page 510. The bad threshold is not affected.

Auto_Threshold_Bad_Projection
This determines the position of the bad threshold. Increase this parameter to increase the bad threshold; decrease the parameter to decrease the bad threshold.For details, see “Threshold calculation detail” on page 510. The fair threshold is not affected.

Use_Default
If set to 0, the auto threshold settings are modified according to the other parameters in this Take Action. If set to 1, the value of the auto threshold

settings for this request are taken from the "parent": the values that have been set for the request type, for the entire application, or the ITCAM default values.

Example: YN:Enable_Auto_Threshold 1 12 50 200 150 300 0

Override_Auto_Threshold: override threshold values

Use the Override_Auto_Threshold Take Action to override fair and bad response time threshold values for any request in the application. In this case, while the baselining statistical data is still preserved, ITCAM does not use automatically calculated thresholds.

Do not override threshold values unless you have analyzed the application performance in detail (or were instructed to override threshold values by IBM Level 3 Support). To adjust threshold values without manually overriding them, see “Enable_Auto_Threshold: set threshold parameters” on page 500.

To remove an override, select a request in the “Request Baseline workspace” on page 319, and select the Enable_Auto_Threshold take action command. Leave all parameters as they are, in order to use the same auto threshold parameters as were used before the override. If you need to change these parameters, see “Enable_Auto_Threshold: set threshold parameters” on page 500.

Command syntax

YN:Override_Auto_Threshold *App_Id Request_Id Fair_Response_Threshold Bad_Response_Threshold*

Parameters:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Request_Id
The request ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Fair_Response_Threshold
The fair response time threshold, in milliseconds.

Bad_Response_Threshold
The bad response time threshold, in milliseconds.

Remove_WebSphere_SubNode: Remove an inactive WebSphere application server

Use the Remove_WebSphere_SubNode command to remove a no-longer-active WebSphere application server from the Navigator tree.

Command syntax

If invoked from the Navigator WebSphere Agent entry, the syntax is:

YN:RemSubNode *server_name*

where *server_name* is the WebSphere server-that is, subnode-name.

If, however, this command is invoked from a subnode of the Navigator WebSphere Agent entry, the syntax is:

YN:RemSubNode

In this case, *server_name* is not required because the subnode name-that is, the server name-is already known.

Set_Application_Monitoring: Set monitoring

Use the Set_Application_Monitoring command to set monitoring of the WebSphere application.

Command syntax

YN:Set_Application_Monitoring *App_Id Monitoring_Enabled Request_Data_Monitoring_Level Request_Data_Sampling_Rate*

where *App_Id* is the application ID which is automatically assigned in the portal from the selection context when Take Action was invoked.

Monitoring_Enabled is a boolean value and the valid values are 0 and 1. It defines whether the monitoring agent application dashboard monitoring feature is enabled for the given application.

Request_Data_Monitoring_Level is an integer value that defines custom request monitoring level for the given application. Valid values are 0 (DISABLE), 1 (LEVEL1), and 2 (LEVEL2). If the Request Monitoring Control Level (see "Application Dashboard (Basic) tab" on page 269) is set to Application, this level overrides any level that the user might set for the entire monitoring agent.

Important: If the Request Monitoring Control Level is set to Server, this Take Action command has no effect on the actual monitoring level. The new level is recorded, but the server-wide setting is applied. If you change the Request Monitoring Control Level to Application, the recorded application monitoring level takes effect.

Request_Data_Sampling_Rate is an integer value that defines custom request monitoring rate (in percentage) for the given application. Valid values range from 0 to 100.

Important: When this Take Action is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Reflex _Automation_Mode is a boolean value and the valid values are 0 and 1. When you select 1, WR application health monitoring accepts reflex automation commands from WASAppHealthGood/Fair/Bad situations and adjusts the monitoring level automatically based on the current application health status. For more information about Tivoli Monitoring (ITM) reflex automation, search for 'reflex automation' in the IBM Tivoli Monitoring Version 6.2.3 info center

Set_Completion_Thresholds: Set completion thresholds

Use the Set_Completion_Thresholds command to define the thresholds of the error rate for the WebSphere application.

Command syntax

YN:Set_Completion_Thresholds *App_Id Fair_Completion_Rate Bad_Completion_Rate*

where *App_Id* is the application ID which is automatically assigned in the portal from the selection context when Take Action was invoked.

Fair_Completion_Rate and *Bad_Completion_Rate* are the values in percentage that define thresholds for fair and bad requests completion rates.

Set_Request_Sampling_Rate: Set the sampling rate for request data

Use the Set_Request_Sampling_Rate command to define the percentage of requests to monitor.

Command syntax

YN:SetRequestSamplingRate *percent*

where *percent* is the percentage of requests you want sampled, an integer from 1 to 100.

Important: This command sets the default sampling rate for a server. This rate is applied to new applications that are detected on this server after you run the command. To change the sampling rate for an existing application, use the command "Set_Application_Monitoring: Set monitoring" on page 504.

Start_Baselining: Start the baselining process

ITCAM can run a *baselining process* for every application. During this process, which runs for a preset period, the data collector collects statistical data on metric values for a given period. Based on this statistical data, the monitoring agent can automatically set the fair and bad thresholds, as well as the typical breakdown of response times for nested requests. Use the Start_Baselining Take Action to start the baselining process.

When ITCAM begins monitoring an application for the first time, it automatically starts this process for the application. However, with time, average response times can change because of configuration, load pattern, database size, and other issues. You can manually start the baselining process again to take these changes into account. You can also use IBM Tivoli Monitoring (ITM) policies and workflow management to run the baselining process every few months.

As soon as you take the Start Baselining action, the baselining process begins. The thresholds are updated when either the Period or the Update Interval passes.

While the baselining process is running, you can trigger a baseline update to immediately set the thresholds based on the information collected so far.

Command syntax

YN:Start_Baselining *App_Id Period Update_Interval Run_Clean*

Parameters:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Period
The time for which the baselining process runs. The data collector collects the necessary statistical information for this entire period. When ITCAM starts the baselining process automatically, it sets the period to seven days. The format is `ddd/hh:mm:ss`.

Update_Interval
If you set this parameter to a time interval, ITCAM updates the thresholds according to the information already collected every time this interval passes. For example, when ITCAM starts the baselining process automatically, it sets the update interval to 1 hour. During the seven days that the initial baselining runs, every hour the thresholds are updated according to the statistical data collected so far (for all request types where at least one request was received during the baselining process). The format is `ddd/hh:mm:ss`.

Run_Clean
Set to either 0 or 1. If set to 0, statistical data collected in any previous baselining for the same requests are kept and "amalgamated" with the new data. If set to 1, only the new data are used for setting the thresholds. Normally, you set this to 1.

Note: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Start_GC_Monitoring: Begin reporting garbage-collection data

Use the Start_GC_Monitoring command to activate the display of garbage-collection statistics. This setting is in addition to the WebSphere application server Verbose Garbage Collection value, which must also be active for garbage-collection data to be reported.

Command syntax

YN:StartGCMonitor

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Start_Request_Monitoring : Begin reporting request data

Use the Start_Request_Monitoring command to activate the display of request data.

Command syntax

YN:StartRequestMonitor *level*

where *level* is the resource-data collection level, either Level1 or Level2. When the collection level is set to Level1, only edge request data-such as servlets and JSPs-are collected. When set to Level2, nested request data, such as JDBC and JMS requests, are also collected.

If the Request Monitoring Control Level (see "Application Dashboard (Basic) tab" on page 269) is set to Server, this level applies to all applications monitored on the server.

Important: If the Request Monitoring Control Level is set to Application, you can set the monitoring level for any operation separately, using the "Set_Application_Monitoring: Set monitoring" on page 504 Take Action command. The monitoring level from the Start_Request_Monitoring Take Action command is applied by default to newly discovered applications.

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Start_Resource_Monitoring: Begin reporting PMI data

Use the Start_Resource_Monitoring command to activate the display of resource (that is, PMI) data. This setting is in addition to the WebSphere application server PMI instrumentation levels, which must also be set for resource data to be reported.

Command syntax

YN:StartResourceMonitor

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Start_WebSphere_Server: Start a WebSphere application server

Use the Start_WebSphere_Server command to start a WebSphere application server.

Command syntax

If invoked from the Navigator WebSphere Agent entry, the syntax is:

YN:StartAppSvr *server_name* *user* *password*

where *server_name* is the WebSphere server name, and *user* and *password* are your own WebSphere application server identifiers set via the WebSphere administrative console (required only if WebSphere global security is enabled).

If, however, this command is invoked from a subnode of the Navigator WebSphere Agent entry, the syntax is:

YN:StartAppSvr *user* *password*

where *user* and *password* are your own WebSphere application server identifiers set via the WebSphere administrative console; these are required only if WebSphere global security is enabled. (In this case, *server_name* is not required because the subnode name-that is, the server name-is already known.)

Stop_Baselining: stop the baselining process

Use the Stop_Baselining Take Action command to immediately stop the baselining process for an application, and recalculate the thresholds based on the request data available up to this point.

Normally you do not have to complete this action. To recalculate the thresholds based on the request data available up to this point, without stopping the baselining process, see "Update_Baseline: trigger a baseline update" on page 509.

Command syntax

YN:Stop_Baselining *App_Id*

Parameters:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Stop_GC_Monitoring: Stop reporting garbage-collection data

Use the Stop_GC_Monitoring command to end the display of garbage-collection statistics.

Command syntax

YN:StopGCMonitor

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Stop_Request_Monitoring: Stop reporting request data

Use the Stop_Request_Monitoring command to end the display of request data.

Command syntax

YN:StopRequestMonitor

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Stop_Resource_Monitoring: Stop reporting PMI data

Use the Stop_Resource_Monitoring command to end the display of resource (that is, PMI) data.

Command syntax

YN:StopResourceMonitor

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

Stop_WebSphere_Server: Stop a WebSphere application server

Use the Stop_WebSphere_Server command to stop an application server.

Command syntax

If invoked from the Navigator WebSphere Agent entry, the syntax is:

YN:StopAppSvr *server_name* *user* *password*

where *server_name* is the WebSphere server name, and *user* and *password* are your own WebSphere application server identifiers set through the WebSphere administrative console (required only if WebSphere global security is enabled).

If, however, this command is invoked from a subnode of the Navigator WebSphere Agent entry, the syntax is:

YN:StopAppSvr *user* *password*

where *user* and *password* are your own WebSphere application server identifiers set through the WebSphere administrative console; these are required only if WebSphere global security is enabled. (In this case, *server_name* is not required because the subnode name-that is, the server name-is already known.)

Update_Baseline: trigger a baseline update

If the baselining process is running, the thresholds are set automatically when either the Period or the Update Interval passes. For the initial baselining process, the first automatic update happens after one hour. With the Update_Baseline Take Action, you can force ITCAM to update the thresholds immediately, based on the information collected so far. This can be useful if you do not want to wait for the periodic automatic update. When the automatic update time comes, the threshold is updated again.

If a baselining process is not running for the application, an error is raised. If no requests of a given request type have been received since the baselining process has started, the update has no effect for this request type.

Command syntax

YN:Update_Baseline *App_Id*

Parameters:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Important: When this Take Action command is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere application server instance).

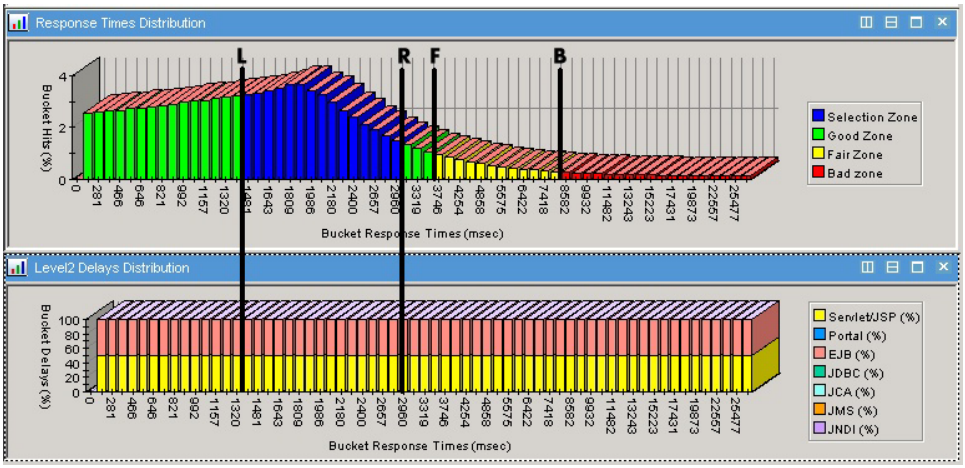
Threshold calculation detail

If you want to set parameters that affect the calculation of the automatic response time thresholds, you might need to know the details of this calculation.

ITCAM processes the baselining statistical data and applies the auto threshold parameters in the following way. The parameters are set in the Enable_Auto_Threshold take action command, see “Enable_Auto_Threshold: set threshold parameters” on page 500.

1. The response time results are sorted into up to 64 "buckets", from zero to the maximum response time encountered for this request. The buckets do not represent equal time intervals; for response time regions where most of the "hits" fall, the buckets represent smaller intervals. For example, if the maximum encountered time is 1000 ms but most response times are 300 - 500 ms, then the first bucket might be 0 - 50 ms, but there can also be buckets representing response times of 305 - 310 ms and 400 - 402 ms. ITCAM distributes the bucket borders so that the largest number of hits in any one bucket is not more than three times the number of hits in the smallest bucket.

   ITCAM calculates the percentage of the total amount of requests that fall into each response time bucket. ITCAM then divides it by the time interval width represented by the bucket. This is shown on the **Response Times Distribution** bar chart in the “Request Baseline workspace” on page 319.



   Each bar represents a bucket, and the bar height shows the percentage of the requests in this bucket. All the subsequent calculations are rounded up to buckets.

2. ITCAM determines the selection zone, which contains the "typical" response time values. This zone is represented by the bars colored blue on the chart.

   ITCAM finds the response time interval (left border L to right border R) where the following statements are true:

   - The percentage of hits that fall into this interval is no less than the Auto_Threshold_Percent parameter.
   - The spread of the time interval, calculated as (R/L)*100 - 100, is not greater than the Auto_Threshold_Deviation parameter.

     Important: The Auto_Threshold_Deviation parameter does not denote the statistical definition of deviation.

If several zones match these criteria, ITCAM chooses the one where the following value is the greatest: S/(R-L), where S is the total number of hits that fell into this zone.

If a zone where both requirements are true cannot be found at all, ITCAM first determines the interval where the percentage of hits is not less than the Auto_Threshold_Percent parameter while the spread is as little as possible. Then, within this interval, it finds a zone where the spread is not greater than the Auto_Threshold_Deviation parameter and the percentage of hits is as large as possible.

ITCAM determines the typical nested request times from the nested request times in this zone, shown on the **Level2 Delays Distribution** chart.

3. Finally, ITCAM calculates the thresholds.

   The fair threshold is determined using the left and right borders of the selection zone and the Auto_Threshold_Fair_Projection parameter:

   F = L + ((R-L) * Auto_Threshold_Fair_Projection / 100)

   The bad threshold is calculated in the same way, using the Auto_Threshold_Bad_Projection parameter:

   B = L + ((R-L) * Auto_Threshold_Bad_Projection / 100)

   Example: the left border of the selection zone might be L=1450 ms, and the right border R=3000 ms. By default, Auto_Threshold_Fair_Projection=150, and Auto_Threshold_Bad_Projection=300. In this case:

   - The fair response threshold is F = 1450 + ((3000-1450) * 150 / 100) = 3775 ms
   - The bad response threshold is B = 1450 + ((3000-1450) * 300 / 100) = 6100 ms

WebSphere XD Overview for ITCAM Agent for WebSphere Applications

ITCAM for Application Diagnostics provides enhanced support for monitoring Virtual Enterprise and Compute Grid products, as well as the Intelligent Management and Batch features of WebSphere Application Server 8.5. The cells managed by these products or features are called WebSphere XD (Extended Deployment) cells. For each XD cell, configured for monitoring by the WebSphere agent, the Tivoli Enterprise Portal shows the subnode under the agent node in the navigation tree. The workspaces under the XD cell subnode show the XD monitoring information.

The XD monitoring data is collected through a JMX connection to the deployment manager server and does not require a data collector. However, if a data collector is installed on any WebSphere XD server, it is possible to drill down to more detailed information.

ITCAM provides the following Virtual Enterprise monitoring features:

Monitors the status and metrics of the ODR (On Demand Router) server:

- ODR server status – running, not running, number of running ODR servers in the cell.
- ODR server process JVM and OS metrics.
- Collects requests metrics from ODR servers in the cell and provides summarized statistics over cell, cluster, server, and application.

Monitors the status and metrics of dynamic clusters

- Dynamic clusters topology
- Cluster configuration and state

- Application servers in cluster
- Current number of servers running in cluster
- Max number of servers in cluster
- Dynamic WLM Weight
- ODR server process JVM and OS metrics

Monitors XD application servers JVM information
- Server process JVM and OS metrics

ITCAM provides the following Compute Grid monitoring features:

Monitors Job Scheduler servers
- Job Scheduler server status – running, not running, number of running Job Scheduler servers.
- Job performance metrics reported by job scheduler servers, summarized over cell and per job scheduler.
- Details on queued and executing jobs, including notifications and job steps.
- Job Scheduler server process JVM and OS metrics.

Monitors Grid Endpoint servers
- Grid Endpoint server status – running, not running.
- Job performance metrics reported by grid endpoint servers, summarized over cell, service policy, and application.
- Grid endpoint server process JVM and OS metrics.

There are a number of situations provided to detect problems in the XD environment and to open Tivoli Enterprise Portal events.

ITCAM for Application Diagnostics - WebSphere XD Cell workspaces

The Tivoli Enterprise Portal XD component has the following workspaces:

Table 78. WebSphere XD Cell Workspaces

Navigation tree and Workspaces	Secondary Workspaces (not in tree)	Description
WebSphere Agent		Displays the overall summary information about the XD Cell.
XD Cell		XD Cell Subnode that contains all the workspaces associated with WebSphere XD.
	Configuration	Used to configure XD connection settings
Virtual Enterprise		Displays ODR Summary Statistics.
Service policies		Displays ODR Statistics for each service policy.
	Service policy violations	Displays a list of service policy violation tasks.
	Service policy	Displays ODR Statistics for the selected policy.
Applications		Displays ODR statistics for each transaction class, application, and module.

Table 78. WebSphere XD Cell Workspaces (continued)

Navigation tree and Workspaces	Secondary Workspaces (not in tree)	Description
	Application	Displays ODR statistics for each transaction class, application, and module for the selected application.
Deployment Targets		Displays ODR statistics for deployment targets.
	Static Cluster	Displays ODR statistics for and individual static cluster.
Servers		Displays ODR and JVM operating system statistics.
	Server	Displays ODR and JVM operating system statistics for the selected server.
ODRs		Displays Statistics for ODRs.
	ODR	Displays Statistics for ODRs for the selected ODR server.
Dynamic Clusters		Displays dynamic clusters statistics.
	Dynamic Cluster	Displays dynamic clusters statistics for the selected cluster.
Compute Grid		Displays overall performance, loading, and health statistics on the compute grid.
Job Service Policies		Displays Job Statistics per service policy and summarizes all transactions classes that belong to the service policy.
	Job Service Policy	Displays Job Statistics per service policy and summarizes all transactions classes that belong to the service policy.
Job Applications		Displays Job statistics by transaction class for applications and modules.
	Job Application	Displays Job statistics by transaction class for applications and modules for the selected job application.
Grid Endpoints		Displays total statistics on grid endpoints.
	Grid Endpoint	Displays total statistics for the selected grid endpoint.
Job Scheduler Servers		Displays job scheduler statistics.
	Job Scheduler Server	Displays job scheduler statistics for the selected server.
Jobs		Displays running jobs.
	Job	Displays running jobs for the selected job.

WebSphere XD Cell subnode workspace

The XD Cell subnode displays overall request rates, job execution rates, and service policy violations.

This workspace displays data provided by the "XD Cell attributes" on page 538.

The predefined workspace contains the following items:

- **Situation Event Console** view displays all situation events for the XD Cell.
- The **Cell Summary** table displays connections settings and versions for the cell. You can click the link icon in any table row to click the Configuration link to access the XD Configuration workspace.

Accessing the XD Cell Subnode workspace

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the list of WebSphere agents.
4. Expand the **WebSphere XD Cell**.
5. Click **XD Cell**.

Applications workspace

The applications workspace displays (On Demand Router) ODR Statistics for each unique application, module, transaction, class, and protocol combination summarized over all ODRs in the cell.

All statistics that display are the same as the Virtual Enterprise workspace. You can view more detailed information about the selected combination by clicking the link in the Applications table

This workspace displays data provided by the "ODRs attributes" on page 530.

The predefined workspace contains the following items:

- **Arrived Request Rate - History** chart displays the same information as the ODR request rate on the cell subnode. Rates display per second for each request.
  - Request Arrived Rate
- **Average Response Time - History** displays information about response timings in milliseconds:
  - Average Response Time gives ODR end to end request time
- The **Applications** table displays the latest status of each application

Accessing the Applications workspace

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent node.
4. Expand the **XD Cell**.
5. Click the **Applications** workspace.

Accessing the Application workspace for a specific application

- Click the link icon to view details for a specific application.
- In the Applications report, right-click the link icon and from the menu, click **Application** to display details about a specific application.
  - **Service Policy** Click this link to view specific server policies related to the server in the Service Policy workspace.

- **Per ODRS** Click this link to view the ODRS specific to the server in the ODRS workspace.
- **Per Deployment Target** Click this link to access Dynamic clusters specific to the server in the Deployment Target workspace.
- **Applications**
- **Link Wizard** Click to create links to workspaces.
- **Link Anchor** Click to display visual indicators on tables where customized links have been defined.

Note: Only applications that have some loading through the On Demand Router (ODR) display in the Applications workspace.

The application workspace displays Request Rate History and Average Request Time History as well as the following charts Request percent History and Request Time Deviation history.

Compute Grid workspace

The Compute Grid workspace displays Job Statistics for the cell. Information includes overall performance, loading, and health statistics of the compute grid for the cell.

This workspace displays data provided by the "Compute Grid Attributes" on page 526.

The predefined workspace contains the following items:

- Job Rate History chart: displays the overall job rates for the cell in minutes. The rates are Dispatched, Dispatch Error, Started and Completed.
- Average Job Time - History chart: displays the average job timings for the cell in seconds. The values are Queue, Dispatch, Dispatch Error, and Execute.
- Job Count - History displays the job count for the cell. The chart displays the status as either Queued or Running.
- Compute Grid table displays the current job statistics for the cell.

Accessing the Compute Grid workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Compute Grid** workspace.

The Compute Grid table displays details on job performance. In the table, right-click the link icon and from the menu, click **Service Policies** to view the Job Service Policies workspace.

Deployment Targets workspace

The deployment targets workspace displays ODR statistics per deployment target and per protocol.

This workspace displays data provided by the "ODRs attributes" on page 530.

The predefined workspace contains the following items:

- **Arrived Request Rate History** displays the arrived request rate history for each deployment target.
- **Average Response Time History** displays the average response time history for each deployment target.
- **Deployment Targets** displays the latest status of each deployment target.

Accessing the Deployment Targets workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Deployment Targets** workspace.

Accessing the specific information from the Deployment targets table

You can view more detailed information from the Deployment targets table.

- From the Deployments Target report, right-click the link icon and from the pop-up menu, click one of the following options to move to the relevant workspace.

  Important: Dynamic cluster and Static cluster and Server workspace links display only for the corresponding deployment target types.
  - **Server** Click this link to view the Server workspace.
  - **Dynamic Cluster** Click this link to view the Dynamic Cluster workspace. Dynamic Cluster can start, stop, and, create servers on demand.
  - **Static Cluster** Click this link to view the Static Cluster workspace. Static cluster servers are created and started by users.
  - **Per Service Policies:** Click this link to view the service policies specific to the selected deployment target.
  - **Per Applications** Click this link to view the applications specific to the selected deployment target.
  - **Per ODRs** Click this link to view the deployment target statistics for each ODR server that routes requests to the selected deployment target.

Dynamic Clusters workspace

The Dynamic Clusters workspace displays the topology of the dynamic clusters, servers, nodes, and node groups in your environment.

This workspace displays data provided by the “Dynamic Clusters attributes” on page 528 and the “Dynamic Cluster Topology attributes” on page 528.

The predefined workspace contains the following items:

- Dynamic Cluster Topology displays visual representation of the dynamic clusters and servers that belong to them with the corresponding nodes. You can point to each item for details.
- Dynamic Clusters table displays the latest information about the dynamic clusters in the cell.

Accessing the Dynamic Clusters workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Dynamic Clusters** workspace.

Accessing a specific Dynamic Cluster workspace

To access this workspace from the dynamic clusters workspace, use one of the following procedures:

- From the Dynamic Clusters table, click the link icon and select **Dynamic Cluster**.

When you click **Dynamic Clusters** the following information displays:

- **Topology** displays the selected dynamic cluster with its servers and corresponding nodes, and node groups. Point to the items to view additional information.
- **Request Rate History** chart displays the arrived request rate history in requests per second for the selected dynamic cluster.
- **Average Request Time History** chart displays the average request time history in milliseconds for the selected dynamic cluster.
- **Dynamic Cluster** table displays configuration and status data specific to the selected dynamic cluster.
- **ODR** statistics table displays ODR data specific to the selected dynamic cluster.

Grid Endpoints workspace

The Grid Endpoints workspace displays job statistics for each grid endpoint, that is for each WebSphere server that runs jobs.

This workspace displays data provided by the “Grid Endpoint attributes” on page 529.

The predefined workspace contains the following items:

- Job Started Rate - History chart displays the Job Started rate per minute.
- Average Completion Time- History chart displays the Average Job Completion Time in minutes.
- The Grid Endpoints table that displays statistics for the server.

Accessing the Grid Endpoints workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.

5. Click the **Grid Endpoints** workspace.

Accessing a specific Grid Endpoint workspace

- From the Grid Endpoints table, right-click the link icon and from the menu, click **Grid Endpoint**.

When you click **Grid Endpoint** the following information displays:

- **Average Completion Time - History** chart displays the average job completion time in minutes.
- **Job Rate - History** chart displays the job rate per minute. The status is Started or Completed.
- **JVM Heap Size** chart displays the amount of heap size in MB, the status is Used or Free.
- **CPU Used - History** displays the percentage of the host CPU used by the JVM.
- The Grid Endpoint table displays the job statistics for the selected policy.
- It is also possible that other workspace links that are available depending on the configuration. Click the relevant link to view the workspace of your choice:
- **Server Diagnostic** Click to view the WebSphere application server.
- **Per Job Service Policies** Click to view the Job Service Policies workspace
- **Per Applications** Click to view the Job Applications workspace.
- **Grid Endpoint** Click to view details of a specific grid endpoint.
- **Link Wizard** Click to create links to workspaces.
- **Link Anchor** Click to display visual indicators on tables where customized links have been defined.

Job Applications workspace

The Job Applications workspace displays job statistics for each application, module, and transaction class combination.

The predefined workspace contains the following items:

- Job Started Rate - History chart displays the Job Started rate per minute.
- Average Completion Time- History chart displays the Average Job Completion Time in minutes.
- The Job Applications table displays job statistics for each job application.

Accessing the Job Applications workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Job Applications** workspace.

Accessing a specific Job Application workspace

- From the Job Applications table, right-click the link icon and from the menu, click **Job Application**.

When you click **Job Application** the following information displays:

- Jobs Running - History chart displays the jobs running per minute.
- Average Completion Time - History chart displays the average job completion time in minutes.
- Job Started Rate - History chart displays the job rate per minute. The status is Started or Completed.
- The Job Application table displays the job statistics for the selected policy.

Job Scheduler Servers workspace

The Job Scheduler Servers workspace displays job statistics for each WebSphere application server server where Job Scheduler is running.

This workspace displays data provided by the "Compute Grid Attributes" on page 526 attributes.

The predefined workspace contains the following items:

- Job Dispatched Rate - History chart displays the Job Dispatched rate per minute.
- Average Queue Time- History chart displays the Average Job Queue Time in minutes.
- The Job Scheduler Servers table displays job statistics for each Job Scheduler.

Accessing the Job Scheduler Servers workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Job Scheduler Servers** workspace.

Accessing a specific Job Scheduler Servers workspace

- From the Job Scheduler Servers table, right-click the link icon and from the menu, click **Job Scheduler Server**.

When you click **Job Scheduler Server** the following information displays:

- Job Rate - History chart displays the job rate per minute. The statuses are Dispatched, Dispatched Error, or Completed.
- Job Average Time - History chart displays the average job completion time in seconds. The status is Queue, Dispatched, or Dispatched Error.
- JVM Heap Size chart displays the amount of heap size in MB, the status is Used or Free.
- CPU Used - History displays the percentage of the host CPU used by the JVM.
- The Job Scheduler Server table displays the job statistics for the selected policy.
- **Link Wizard** Click to manually add links to other workspaces.
- **Link Anchor** Click to display visual indicators on tables where customized links have been defined

Job Service Policies workspace

The Job Service Policies workspace displays jobs statistics for each service policy.

This workspace displays data provided by the "Service Policy Violations attributes" on page 536 attributes.

The predefined workspace contains the following items:

- Job Started Rate - History chart displays the Job Started rate per minute.
- Average Completion Time- History chart displays the Average Job Completion Time in minutes.
- The Job Service Policies table displays job statistics for each job service policy.

Accessing the Job Service Policies workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Job Service Policies** workspace.

Accessing a specific Job Service policy workspace

- From the Job Service Policies table, right-click the link icon and from the menu, click **Job Service Policy**.

When you click **Job Service Policy** the following information displays:

- Jobs Running - History chart displays the jobs running per minute.
- Average Completion Time - History chart displays the average job completion time in minutes.
- Job Rate - History chart displays the job rate per minute. The status is Started or Completed.
- The Service Policy table displays the job statistics for the selected policy.
- It is also possible that other workspace links that are available depending on the configuration. Click the relevant link to view the workspace of your choice:
- **Job Applications** Click this link to view the Job Applications workspace
- **Job Service Policy** Click to view details of a specific job service policy.
- **Link Wizard** Click to manually add links to other workspaces.
- **Link Anchor** Click to display visual indicators on tables where customized links have been defined

The Service Policy table displays information about jobs and the service policy goal.

Jobs workspace

The Jobs workspace displays scheduled and running jobs.

This workspace displays data provided by the "Jobs attributes" on page 533 and "WebSphere XD Job Filter attributes" on page 533.

The predefined workspace contains the following items:

- Job Filter - right-click the link to access the Set Job Filter where you can modify the filter display fields.

- Job Topology - displays the submitted jobs that have passed through the filter. You can point to each item to view the details.
- The Jobs Instances table displays the status of each job.

Accessing the Jobs workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Jobs** workspace.

Setting the Job Filter

- From the Job Filter table, right-click the link icon and from the menu, click **Set Job Filter**.

You can add information to the following fields to change how the job filter table displays:

- **Grid Endpoint**
- **Job Scheduler**
- **Total Time Greater**
- **Queue Time Greater**
- **Job Name**
- **Maximum Jobs**
- **Sort Order** The options are Ascending or Descending
- **Sort By** The options are Total Time and Queue Time
- **Job ID**

When you click OK to implement the changes, the changes display in the Job Filter view. Once you move to another workspace the Job Filter settings revert to the default settings. If you want the Job Filter to retain the changes, edit the Agent Configuration workspace, Job Filter Settings tab.

Accessing the Job Instances workspace

- From the Job Instances table, right-click the link icon and from the menu, click **Job Instance**.

When you click **Job Instance** the following information displays and relates directly to the job you selected.

- Topology displays details relating to the selected job. You can point to each item to view the details.
- Job Notifications chart displays the notifications received for the selected job.
- Job Steps chart displays executed and executing job steps.
- The Job table displays the job statistics for the selected job.

ODRs workspace

The ODRs workspace displays request statistics for each ODR server (per protocol).

This workspace displays data provided by the "ODRs attributes" on page 530.

The predefined workspace contains the following items:

- **Arrived Request Rate - History** Chart displays the arrived request rate history in requests per second for each ODR server (per protocol).
- **Average Response Time - History** Chart displays the average response time history in milliseconds for each ODR server (per protocol).
- The **ODRs** table shows the status and statistics for each ODR server (per protocol for the last interval).

Accessing the ODRs workspace

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **ODRs** workspace

Viewing an individual ODR workspace

Use the following instructions to view specific details on an individual ODR server.

- From the ODRs table report, click the link icon and select **ODR**.

The following links are available for detailed request statistics as routed by the selected ODR:

- **Per Service Policies** Click to view the service policy request statistics.
- When you click **ODR** (the default link) the following charts display:
  - Request Rate History
  - Average Request Time History
  - JVM Heap Size History Displays the heap size used and free.
  - CPU Used History displays the percentage of the host CPU used by JVM.

Servers workspace

The servers workspace displays ODR statistics for each server and protocol.

This workspace displays data provided by the "Servers attributes" on page 535.

The predefined workspace contains the following items:

- **Arrived Request Rate - History** Chart displays the arrived request rate history in requests per second for each server (per protocol).
- **Average Response Time - History** Chart displays the average response time history in milliseconds for each ODR server (per protocol).
- The **Servers** table shows current server status and statistics for each server.

Accessing the Servers workspace

To access this workspace, complete the following steps:

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Servers** workspace

Viewing server details

- From the Servers report, click the link icon.
- If the server you select is monitored by a data collector, you can also see a **Server Diagnostics** link. Click this link to access the application server data collector workspaces.
- Click **Server** to view details relating to a specific server.

When you click **Server** the following charts display:

- Request Rate History
- Average Request Time History
- JVM Heap Size History: displays the heap size used and free.
- CPU Used History: displays the percentage of CPU used by JVM.

A Server table is also displayed. The table contains a single row that shows performance information for the server.

Service Policies workspace

The Service policies workspace displays (On Demand Router) ODR statistics for service policies and protocol, summarized over all the ODRs in the XD Cell.

The statistics that display are the same as the statistics in the virtual enterprise workspace. You can view more detailed information about the selected service policy by clicking the link in the policies table.

Accessing the Service policies workspace

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Service Policies** workspace

This workspace displays data provided by the “Service Policy Violations attributes” on page 536.

The predefined workspace contains the following items:

- **Arrived Request Rate - History** Chart displays the arrived request rate history in requests per second for each server (per protocol).
- **Average Response Time - History** Chart displays the average response time history in milliseconds for each ODR server (per protocol).

- The **Service Policies** table shows a current sample of the workplace statistics. Click the link in a table row to view details of individual service policies.

Accessing the Service policy violations workspace

The Service Policy violations workspace displays a list of service policy violation tasks.

1. To access this workspace, from the **XD Cell**, select the **Service Policies** node.
2. Right-click**Service Policies > Workspaces > Service Policy Violations**.
3. The **Open Service Policy Violation Tasks** report displays a list of all open tasks.
4. You can sort the report according to the columns displayed. Click the column that you want to sort by to display the filter arrow in the column, then click the arrow up or down to sort.
5. Click the link arrow for a row to view a specific service policy violation.

Accessing the Service policies workspace

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Service Policies** workspace.

This workspace displays data provided by the "Service Policy Violations attributes" on page 536.

The predefined workspace contains the following items:

- **Arrived Request Rate - History** Chart displays the arrived request rate history in requests per second for each server (per protocol).
- **Average Response Time - History** Chart displays the average response time history in milliseconds for each ODR server (per protocol).
- The **Service Policies** table shows a current sample of the workplace statistics. Click the link in a table row to view details of individual service policies.
- From the Service policies report, click the link icon for a row.
- Right-click the link icon to view the following menu options:
  - **Service Policy** (default) displays details of the selected policy
  - **Per ODRs** Click this link to view the service policy request statistics per ODR server.
  - **Per Deployment Target** Click this link to view the service policy request statistics per deployment target.
  - **Applications** Click this link to access the applications specific to the service policy in the Applications workspace.

This Service policy workspace displays the charts for the following information:

- **Request Rate History** Chart displays request rates in requests per second.
  - **Request Arrived Rate** displays the rate per second of requests arrived at ODR during the sampling interval.

  - **Request Dispatched Rate** displays the rate per second of the requests dispatched from ODR to server during the sampling interval.
  - **Request Serviced Rate** displays the rate per second of requests returned from server to ODR during the sampling interval.
- **Average Request Time - History** displays information about request timings in milliseconds.
  - **Average Wait Time** the average wait time of the request in the queue.
  - **Average Response Time** the average response time of the request.
  - **Average Service Time** the average service time of the request.
  - **Average Service Time out** The average service time of the requests completed due to timeout during the sampling interval.
  - **Average Response Time out** The average response time of the requests completed due to timeout during the sampling interval.
  - **Average Dequeue Time** The average dequeue time of the requests dequeued during the sampling interval.
- **Request Percent History** Chart displays miscellaneous request percents.
  - **Failed**
  - **Dropped**
  - **Above Goal**
  - **Timed Out**
- **Average ODR Queue Length** Chart displays average number of requests in ODR queue for the selected service policy.

Virtual Enterprise workspace

The virtual enterprise workspace displays ODR statistics by protocol summarized over all the ODRs in the cell.

If you are using simple setup only the HTTP/S protocol is present.

This workspace displays data provided by the "ODRs attributes" on page 530.

The predefined workspace contains the following items:

- **Request Rate History** displays the same information as the ODR request rate on the cell subnode. Rates are displayed per second for each request.
  - **Request Failed Rate** displays the rate per second of requests returned with an error indicator during the sampling interval.
  - **Request Arrived Rate** displays the rate per second of requests arrived at ODR during the sampling interval.
  - **Request Dropped Rate** Displays the rate per second of requests dropped by ODR after arrival or later from the queue during the sampling interval.
  - **Request Dispatched Rate** displays the rate per second of the requests dispatched from ODR to server during the sampling interval.
  - **Request Serviced Rate** displays the rate per second of requests returned from server to ODR during the sampling interval.
  - **Request Above Goal Rate** displays the rate per second of requests above response time threshold during the sampling interval.
- **Average Request Time - History** displays information about request timings in milliseconds.
  - **Average Wait Time** displays the average wait time of the request in the queue.

- **Average Response Time** displays the average response time of the request.
- **Average Service Time** displays the average service time of the request.
- **Average Service Time out** Displays the average service time of the requests completed due to timeout during the sampling interval.
- **Average Response Time out** Displays the average response time of the requests completed due to timeout during the sampling interval.
- **Average Dequeue Time** Displays the average dequeue time of the requests dequeued during the sampling interval.
- **Average Queue Length** Displays this as zero in a simple setup.
- **Average Request Time Deviation History** Displays is the average response time in milliseconds.
- **Response Time Deviation** displays the response time deviation.
- **Service Time Deviation** displays the average service time deviation during the sampling interval.

- The **Virtual Enterprise** table shows a current sample of the workplace statistics.
- You can also access the Service Policies workspace from this table.

Accessing the Virtual Enterprise workspace

1. In the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. In the node list, expand the entry that corresponds to the node that you want to select.
3. In that node list of monitored applications, expand the WebSphere agent.
4. Expand the **XD Cell**.
5. Click the **Virtual Enterprises** workspace.

Accessing the service policy workspace

From the Virtual Enterprise report, click the link icon and then click **Service Policy** to access the Service Policy workspace.

WebSphere XD Attributes

IBM Tivoli Composite Application Manager Agent for WebSphere Applications uses several attribute groups to store data for WebSphere XD monitoring.

Compute Grid Attributes

The **Compute Grid** attributes provide information and statistics about the WebSphere XD Compute Grid.

The attributes within this group are used to build the “Compute Grid workspace” on page 515 and the “Job Scheduler Servers workspace” on page 519.

Average Job Dispatch Time (min) Indicates the average time spent to dispatch jobs during the sampling interval.

Average Dispatch Error Time (min) Indicates the average time spent for the failed job dispatches during the sampling interval.

Average Job Execution Time (min) Indicates the average job execution time during the sampling interval.

Average Job Queue Time (min) Indicates the average time jobs spent in queue during the sampling interval.

Cell Name Indicates the WebSphere XD Cell name.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Job Completed Count Indicates the number of jobs completed during the sampling interval.

Job Completed Rate (per min) Indicates the rate per minute of the jobs completed during the sampling interval.

Job Dispatch Error Count Indicates the number of job dispatch errors during the sampling interval.

Job Dispatch Error Rate (per min) Indicates the rate per minute of the job dispatch errors during the sampling interval.

Job Dispatched Count Indicates the number of jobs dispatched during the sampling interval.

Jobs Dispatched Rate (per min) Indicates the rate per minute of the jobs dispatched during the sampling interval.

Jobs In Queue Indicates the current number of jobs in Job Scheduler queue.

Jobs Running Indicates the current number of running jobs.

Job Scheduler Server Name Indicates the WebSphere server name of the Job Scheduler.

Job Scheduler Server Node Indicates the WebSphere node name of the Job Scheduler.

Job Scheduler Servers Running Indicates the current number of running Job Scheduler servers.

Job Scheduler Deployment Target Name The deployment target name of the Job Scheduler.

Job Scheduler Deployment Target Type The deployment target type of the Job Scheduler.

Job Started Count Indicates the number of jobs started during the sample interval.

Job Started Rate (per min) Indicates the rate per minute of the jobs started during the sampling interval.

Label Indicates the row label.

Origin Node Indicates the XD Cell subnode.

Row Type Indicates the row type.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Server State Indicates the WebSphere server state of the Job Scheduler. The state can be Running, Stopped, Starting, Stopping, or Maintenance.

Dynamic Clusters attributes

The **Dynamic Clusters** attributes provide information about WebSphere XD dynamic clusters.

The attributes within this group are used to build the "Dynamic Clusters workspace" on page 516

Cell Name Indicates the WebSphere XD Cell name.

Cluster Member Type Indicates the cluster member type.

Cluster Name Indicates the cluster name.

Cluster State Indicates the cluster state.

Configured Servers Indicates the configured number of servers in the dynamic cluster.

Interval (sec) Indicates the length of the sample interval in seconds.

Maximum Servers Indicates the maximum number of servers in the dynamic cluster.

Minimum Servers Indicates the minimum number of servers in the dynamic cluster.

Origin Node Indicates the XD Cell subnode.

Running Servers Indicates the current number of running servers in the dynamic cluster.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Dynamic Cluster Topology attributes

The WebSphere XD **Dynamic Cluster Topology** attributes provide topology information about the dynamic clusters.

The attributes within this group are used to build the "Dynamic Clusters workspace" on page 516.

Cell Name Indicates the WebSphere XD Cell name.

Filter Indicates the filter which is applied to the topology view.

From Node ID Indicates the incoming link topology node ID.

ID Indicates the topology node ID.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Name Indicates the topology node name.

Origin Node Indicates the XD Cell subnode.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

State Indicates the topology node state.

To Node ID Indicates the outgoing link topology node ID.

Type Indicates the topology node type.

Grid Endpoint attributes

The WebSphere XD **Grid Endpoint** attributes provide information about compute grid endpoint servers.

The attributes within this group are used to build the "Grid Endpoints workspace" on page 517, "Job Applications workspace" on page 518 and,"Job Service Policies workspace" on page 519

Application Indicates the job application name.

Application Label Indicates the application label.

Average Job Completion Time (min) Indicates the average job execution time during the sampling interval.

Cell Name Indicates the WebSphere XD Cell name.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Job Completed Count Indicates the number of jobs completed during the sampling interval.

Job Completed Rate (per min) Indicates the rate per minute of the jobs completed during the sampling interval.

Jobs Running Indicates the average job execution time during the sampling interval.

Job Started Count Indicates the number of jobs started during the sampling interval.

Job Started Rate (per min) Indicates the rate per minute of the jobs started during the sampling interval.

Label Indicates the row label.

Module Indicates the job module name.

Node Name The name of the Grid Endpoint server node.

Origin Node Indicates the XD Cell subnode.

Row Type Indicates the row type.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Server Name The name of the Grid Endpoint server.

Service Policy Indicates the job service policy name.

Service Policy Goal Value (min) Indicates the service policy goal value.

Service Policy Goal Type Indicates the service policy goal type.

Service Policy Importance Indicates the service policy importance.

Transaction Class Indicates the job transaction class name.

ODRs attributes

The **ODRs** (On Demand Routers) attributes provide information about the WebSphere XD On Demand Routers.

The attributes within this group are used to build the "ODRs workspace" on page 522.

Average Dequeue Time (msec) The average dequeue time of the requests dequeued during the sampling interval.

Average Queue Length The average ODR queue length.

Average Response Time (msec) The average response time of the request.

Average Response Timeout (msec) The average response time of the requests completed due to timeout during the sampling interval.

Average Service Time (msec) The average service time of the request.

Average Service Timeout (msec) The average service time of the requests completed due to timeout during the sampling interval.

Average Wait Time (msec) The average wait time of the request in the queue.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

ODR Server Name The ODR server name.

ODR Server Node The ODR node name.

ODR Servers Running How many ODR servers are running The valid format is a 4-byte integer.

Protocol The protocol name.

ODR Server Status The ODR server status. Status can be Stopped, Running or Maintenance.

Request Above Goal Count The number of requests above response time threshold during the sampling interval.

Request Above Goal Percent Indicates the percent of serviced requests which were above response time threshold during the sampling interval.

Request Above Goal Rate (per sec) The rate per second of requests above response time threshold during the sampling interval.

Request Arrived Count The number of requests arrived at ODR during the sampling interval.

Request Arrived Rate (per sec) The rate per second of requests that arrived at ODR during the sampling interval.

Request Failed Count The number of requests returned with an error indicator during the sampling interval.

Request Failed Percent Indicates the percent of requests that returned, during the reported interval, with an error indicator.

Request Failed Rate (per sec) The rate per second of requests returned with an error indicator during the sampling interval.

Request Dispatched Count The number of requests dispatched from ODR to server during the sampling interval.

Request Dispatched Rate (per sec) The rate per second of the requests dispatched from ODR to server during the sampling interval.

Request Dropped Count The number of requests dropped by ODR after arrival or later from the queue during the sampling interval.

Request Dropped Rate (per sec) The rate per second of requests dropped by ODR after arrival or later from the queue during the sampling interval.

Request Dropped Percent Indicates the percent of requests dropped at OnDemand Router.

Response Time Deviation (msec) The response time deviation.

Request Timed Out Percent Indicates the percent of requests that returned, during the reported interval, due to service timeout.

Request Serviced Count The number of requests returned from server to ODR during the sampling interval.

Request Serviced Rate (per sec) The rate per second of requests returned from server to ODR during the sampling interval.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Service Time Deviation (msec) The average service time deviation during the sampling interval.

Origin Node The XD Cell subnode.

Cell Name The WebSphere XD Cell name.

Deployment Target Name The Deployment Target name.

Deployment Target Type Indicates the deployment target type. Deployment target types can be Dynamic Cluster, Static Cluster, or Server.

Row Type Indicates the row type.

Server Name Indicates the server name.

Server Node Indicates the server node name.

Server Status Indicates the server status. The Server status can be Stopped, Running or Maintenance.

Server Weight Indicates the server weight in cluster.

Application Indicates the application name.

Module Indicates the name of the module.

Transaction Class Indicates the transaction class.

Service Policy Indicates the service policy name.

Service Policy Importance Indicates the service policy importance.

Service Policy Goal Type Indicates the service policy goal type.

Service Policy Goal Value (msec) Indicates the service policy goal value.

Request Timed Out Count Indicates the number of requests that completed due to service timeout during the sampling interval.

Requests Timed Out Rate (per sec) Indicates the rate per second of requests that completed due to service timeout during the sampling interval.

Label Indicates the row label.

Application Label Indicates the application label.

Jobs attributes

The WebSphere XD **Jobs** might be in queue, in a submitted state or in a suspended state. The Jobs workspace provided information about jobs that are active and not finished.

The attributes within this group are used to build the "Jobs workspace" on page 520.

Cell Name Indicates the WebSphere XD Cell name.

Execute Time (min) The job execution time.

Grid Endpoint Server Name The Grid Endpoint server name where job is running.

Grid Endpoint Server Node The Grid Endpoint node name.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Job ID The job ID.

Job Type The job type.

Job Scheduler Server Name The Job Scheduler server name on which job is scheduled.

Job Scheduler Server Node The Job Scheduler node name.

Origin Node Indicates the XD Cell subnode.

Queue Time (min) The current number of minutes job spent in queue.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Start Date and Time The job start date and time.

State Indicates the job state.

Submitter The job submitter name.

Suspended Until The job is suspended until the specified date and time.

Total Time (min) The total time passed since the job was submitted.

Filter The name of the job filter. The valid format is an alphanumeric string, with a maximum of 256 characters.

WebSphere XD Job Filter attributes

The WebSphere XD **Job Filter** attributes provide information about the current job filter.

The attributes within this group are used to build the jobs filter table in the "Jobs workspace" on page 520workspace.

Cell Name Indicates the WebSphere XD Cell name.

Total Time Greater (min) Indicates the filter for job total time to be greater than the specified time.

Grid Endpoint Indicates the filter for the Grid Endpoint full server name (node:server).

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Job ID Indicates the filter for the job ID.

Job Name Indicates the filter for the job name.

Job Scheduler Server Indicates the filter for the Job Scheduler full server name (node:server).

Maximum Jobs Indicates the maximum number of jobs to pass the filter.

Queue Time Greater (min) Indicates the filter for job time in queue to be greater than the specified time.

Sort Order Indicates the job sort order.

Sort By Indicates the jobs attribute to sort by.

Origin Node Indicates the XD Cell subnode.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Filter ID The ID of the filter. The valid format is a 4-byte integer.

Filter The name of the filter. The valid format is an alphanumeric string, with a maximum of 256 characters.

Job Notification attributes

The WebSphere XD **Job Notification** attributes provide information about the current job notifications.

The attributes within this group are used to build the “Jobs workspace” on page 520.

Cell Name Indicates the WebSphere XD Cell name.

Date and Time Indicates the notification date and time.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Job ID Indicates the job ID.

Origin Node Indicates the XD Cell subnode.

Notification Type Indicates the notification type.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Sequence Number Indicates the notification sequence number.

Step Name Indicates the step name.

Job Topology attributes

The WebSphere XD **Job Topology** attributes provide topology information about running jobs.

The attributes within this group are used to build the "Jobs workspace" on page 520.

Cell Name Indicates the WebSphere XD Cell name.

From Node ID Indicates the incoming link node ID.

ID Indicates the node ID.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Job ID Indicates the job ID.

Name Indicates the node name.

Origin Node Indicates the XD Cell subnode.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

State Indicates the node state.

To Node ID Indicates the outgoing link node ID.

Type Indicates the node type.

Filter The name of the job filter. The valid format is an alphanumeric string, with a maximum of 128 characters.

Servers attributes

The **Servers** attributes provide JVM and process information about the WebSphere XD servers.

The attributes within this group are used to build the "Servers workspace" on page 522.

Origin Node Indicates the XD Cell subnode.

Cell Name Indicates the WebSphere XD Cell name.

CPU Used Percent Indicates the server CPU usage percent.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

JVM Heap Used (MB) Indicates the size of the used JVM heap space.

JVM Heap Free (MB) Indicates the size of the free JVM heap space.

JVM Heap Total (MB) Indicates the total JVM heap size.

Process Resident Memory (MB) Indicates the server process resident memory.

Process Total Memory (MB) Indicates the server process total memory.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Server Name The name of the server.

Sever Node The name of the server node.

Service Policy Violations attributes

The WebSphere XD **Service Policy Violations** attributes provide information about open service policy violation tasks.

The attributes within this group are used to build the"Service Policies workspace" on page 523.

Cell Name Indicates the WebSphere XD Cell name.

Date and Time Indicates the date and time when the service policy violation task was open.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Origin Node Indicates the XD Cell subnode.

Row Number Indicates sequential row number.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Service Policy Indicates the service policy name.

Service Policy Importance Indicates the service policy importance.

Service Policy Goal Type Indicates the service policy goal type.

Service Policy Goal Unit Indicates the service policy goal unit.

Service Policy Goal Value Indicates the service policy goal value.

Violation Indicates the service policy violation task description.

Steps attributes

The WebSphere XD **Steps** attributes provide information about current steps.

The attributes within this group are used to build the "Jobs workspace" on page 520.

Cell Name Indicates the WebSphere XD Cell name.

End Date and Time Indicates the step end date.

Execution Time (min) Indicates the step execution time in minutes.

Job ID Indicates the job ID.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Origin Node Indicates the XD Cell subnode.

Result Code Indicates the finished step result code.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Start Date and Time Indicates the step start date.

Status Indicates the finished step status.

Step Name Indicates the step name.

XD Cell subnode attributes

The **XD Cell** Configuration attributes provide configuration information.

The attributes within this group are used to build the "WebSphere XD Cell subnode workspace" on page 513.

Origin Node Indicates the XD Cell subnode.

Cell Name Indicates the WebSphere XD Cell name.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

Row Id The row identifier. The valid format is a 2-byte integer.

View Id The view identifier. The valid format is a 2-byte integer.

Act Id The act identifier. The valid format is a 2-byte integer.

Data The data string. The valid format is an alphanumeric string, with a maximum of 4096 characters.

XD Cell attributes

The WebSphere **XD Cell** attributes provide information about a WebSphere XD cell.

The attributes within this group are used to build the "WebSphere XD Cell subnode workspace" on page 513.

Connection Status Indicates the connection status to the cell deployment manager.

Connection Host Indicates the connection address of the deployment manager.

Connection Port Indicates the connection port of the deployment manager.

Connection Type Indicates the connection type of the deployment manager.

WebSphere Version Indicates the WebSphere version.

WebSphere Location Indicates the WebSphere XD root directory.

Cell Name Indicates the WebSphere XD Cell name.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Origin Node Indicates the XD Cell subnode.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; *Example:* 09/25/09 16:20:46 indicates the data was collected on September 25 2009, at 14:20:46.

ITCAM for Application Diagnostics - WebSphere XD Take Actions

Use the XD Take Action commands in the Tivoli Enterprise Portal interface to add and remove XD cells to and from a WebSphere application server.

WebSphere XD uses two take action commands.

- Add_XD_Cell
- Remove_XD_Cell

You must use the Add_XD_Cell command as part of the XD Cell configuration process.

To remove an XD Cell, you must access this option from the node of the XD cell that you want to remove. For more information, see"Remove_XD_Cell: Remove an XD cell from the WebSphere XD Cell" on page 539

Remove_XD_Cell: Remove an XD cell from the WebSphere XD Cell

Use the Remove_XD_Cell command to remove an XD cell from the WebSphere XD Cell. This take action command is available on the specific cell subnode only.

Command syntax

Before you remove an XD cell from the agent, disable monitoring using the configuration workspace. If you do not disable monitoring, then the Remove_XD_Cell finishes displaying an error message notifying you to disable monitoring beforehand. See "Application Diagnostics Configuration - Basic Tab" on page 268

1. To access the command, right-click the XD Cell node that you want to remove. Click **Take Actions > Select**.
2. From the **Name** drop-down menu, select **Remove_XD_Cell**.
3. Click **OK** to remove the cell.

YN:RemXDCell

See also "ITCAM for Application Diagnostics - WebSphere XD Take Actions" on page 538

ITCAM for Application Diagnostics - XD Agent situations

WebSphere XD has a number of predefined situations that you can use to complete the following tasks:

- Detect importance conditions in the WebSphere XD cell you monitor.
- Create your own situations using the predefined situations as examples.

These predefined situations display an alert status. When these situations trigger an alert, you can investigate the event by opening its workspace. For example, you can use these situations to monitor a WebSphere XD cell for requests not meeting the expected goal.

How the situations work

Situations are tests expressed in IF-TRUE format of system conditions that you want to monitor; the tested value is an ITCAM Agent for WebSphere Applications attribute expressed in the form *attribute-group.attribute-name*. If the specified condition occurs or exists, the situation is true, and an alert is issued.

Avoid using negative values

If you define situations that use a counter or a range of numbers, always provide a threshold or use values in a positive range of numbers. For example, use a greater-than-or-equal-to-zero expression as shown in some of the following predefined situations. This practice prevents a situation from falsely tripping. If the ITCAM Agent for WebSphere Applications Tivoli Enterprise Monitoring Agent encounters an undefined attribute value, it reports this value as a negative number and a situation that specifies a negative number can fire erroneously.

Accessing the situations

A number of predefined situations are shipped to monitor the WebSphere XD Cell. All of the situations except **XDVEAppReqArrivedRateHigh** are active by default.

You can customize thresholds used in a situation to suit your environment. Situation **XDVEAppReqArrivedRateHigh**, needs to be run manually. To start a situation manually, access the situations in the Tivoli Enterprise Portal using one of the following methods:

- From the toolbar on the main menu, click the Situation Editor icon  and scroll to the situation that you want to view.
- In the WebSphere agent, right-click the specific workspace. Click **Manage Situations** to display all the managed situations available. If you want to start, stop, edit or model a situation, right-click the situation and select the option you want.

XDConnectionError monitors connectivity between the monitoring agent and the deployment manager and issues a Critical alert when there is no connection between the monitoring agent and the deployment manager. The formula is:

If

WebSphere_XD_Cell.Connection_Status equals error

then

the situation XDConnectionError is true.

XDCPUUsedHigh monitors the percentage of the CPU being consumed and issues a Critical alert when the CPU usage is higher than 80%. The formula is:

If

WebSphere_XD_Servers.CPU_Used_Percent is greater than 80

then

the situation XDCPUUsedHigh is true.

XDVEODRNotRunning monitors the ODR servers running in the XD cell and issues a Critical alert when the number of ODR servers running equals 0. The formula is:

If

WebSphere_XD_ODR.Servers_Running equals 0 and
WebSphere_XD_ODR.Row_Type equals ODR

then

the situation XDVEODRNotRunning is true.

XDVEODRQueueLengthHigh monitors the average ODR server queue length and issues a Critical alert when the queue length is longer than 10. The formula is:

If

WebSphere_XD_ODR.Average_Queue_Length is greater than 10 and
WebSphere_XD_ODR.Row_Type equals ODR

then

the situation XDVEODRQueueLengthHigh is true.

XDVEAppReqFailedPercentHigh monitors the percentage of failed application requests and issues a Critical alert when more than 0.5 percent of application request fail. The formula is:

If

WebSphere_XD_ODR.Request_Failed_Percent is greater than 0.5 and WebSphere_XD_ODR.Row_Type equals Application

then

the situation XDVEAppReqFailedPercentHigh is true.

XDVEAppReqDroppedPercentHigh monitors the percentage of application requests that drop and issues a Critical alert when percent of dropped request is higher than 0.5 percent. The formula is:

If

WebSphere_XD_ODR.Request_Dropped_Percent is greater than 0.5 and WebSphere_XD_ODR.Row_Type equals Application

then

the situation XDVEAppReqDroppedPercentHigh is true.

XDVEAppReqArrivedRateHigh monitors the application request arrival rate and issues a Critical alert when the request arrival rate is higher than 1000 per second. The formula is:

If

WebSphere_XD_ODR.Request_Arrived_Rate is greater than 1000 and WebSphere_XD_ODR.Row_Type equals Application

then

the situation XDVEAppReqArrivedRateHigh is true.

Note: You must run this situation XDVEAppReqArrivedRateHigh manually.

XDVEAppReqAboveGoalPercentHigh monitors the number of application requests above the response time threshold during the sampling interval and issues a Critical alert when the average request response time is 0.5 percent. The formula is:

If

WebSphere_XD_ODR.Requests_Above_Goal_Percent is greater than 0.5 and WebSphere_XD_ODR.Row_Type equals Application

then

the situation XDVEAppReqAboveGoalPercentHigh is true.

XDVEAppReqTimedOutPercentHigh monitors the percentage of application requests that time out and issues a Critical alert when the percent of timed-out requests is higher than 0.5 percent. The formula is:

If

WebSphere_XD_ODR.Request_Timed_Out_Percent is greater than 0.5 and WebSphere_XD_ODR.Row_Type equals Application

then

the situation XDVEAppReqTimedOutPercentHigh is true.

XDVEServerMaintenance monitors the need for server maintenance and issues a Critical alert when a server requires maintenance. The formula is:

If

WebSphere_XD_ODR.Server_State equals Maintenance and WebSphere_XD_ODR.Row_Type equals Server

then

the situation XDVEServerMaintenance is true.

XDVEDynClusterPartlyRunning monitors for any dynamic clusters that are partially running and issues a Critical alert when this is the case. A cluster is considered partially running when some servers are active, but the number of active servers is lower than the configured minimum amount of instances in the cluster. For example, if a cluster is configured to run with a minimum of four instances, but only three are active, the cluster is partially running. The formula is:

If

WebSphere_XD_Dynamic_Clusters.Cluster_State equals Partially_Running

then

the situation WASPortletResponseTime is true.

XDVESPolicyTaskOpen searches for open service policy violation tasks and issues a Critical alert when one occurs. The formula is:

If

WebSphere_XD_Policy_Violations.Service_Policy is not equal to "empty string"

then

the situation XDVESPolicyTaskOpen is true.

XDCGJobSchNotRunning monitors the connectivity of the Job Scheduler servers and issues a Critical alert when a Job Scheduler server is not running. The formula is:

If

WebSphere_XD_Compute_Grid.Job_Scheduler_Servers_Running equals Cell and WebSphere_XD_Compute_Grid.Row_Type equals Cell

then

the situation XDCGJobSchNotRunning is true.

XDCGJobSchQueueLengthHigh monitors the Job Scheduler queue length and issues a Critical alert when the queue length is higher than 5000 per second. The formula is:

If

WebSphere_XD_Compute_Grid.Jobs_in_Queue is greater than 5000 and WebSphere_XD_Compute_Grid.Row_Type equals Job_Scheduler_Server

then

the situation XDCGJobSchQueueLengthHigh is true.

XDCGJobSchDispErrorPercentHigh monitors the percentage of job scheduler jobs that fail and issues a Critical alert when a job fails. The formula is:

If

WebSphere_XD_Compute_Grid.Average_Job_Dispatch_Error_Time is greater than 0 and WebSphere_XD_Compute_Grid.Row_Type equals Job_Scheduler_Server

then

the situation XDCGJobSchDispErrorPercentHigh is true.

XDCGJobExcessiveTotalTime monitors the duration time of a job and issues a Critical alert when the job execution time is longer than one day (1440 mins). The formula is:

If

WebSphere_XD_Jobs.Total_Time is greater than 1440

then

the situation XDCGJobExcessiveExecTotalTime is true.

XDCGJobExcessiveQueueTime monitors job queue time and issues a Critical alert when the job is in a queue for longer than 10 minutes. The formula is:

If

WebSphere_XD_Jobs.Queue_Time is greater than 10

then

the situation XDCGJobExcessiveQueueTime is true.

WebSphere Extreme Scale overview for ITCAM Agent for WebSphere Applications

ITCAM Agent for WebSphere Applications provides monitoring for WebSphere Extreme Scale deployment. You can configure monitoring for an Extreme Scale zone, or several zones, under the node for any server belonging to the zone or zones. You can drill down to view information for different servers, map sets, and partitions within the zone or zones.

WebSphere Extreme Scale prerequisites

For WebSphere Extreme Scale monitoring, your environment must meet the following conditions.

- In the WebSphere Extreme Scale server properties file, statistics must be enabled. For details, see the WebSphere Extreme Scale Administration Guide, in the section on monitoring with managed beans (MBeans).
- ITCAM Agent for WebSphere Applications must be installed on at least one server in the monitored Extreme Scale zone, with Catalog Service enabled on the server. Alternatively, the Agent can be installed on a different host with good network connectivity to such a server.
- For hung thread detection, ITCAM Agent for WebSphere Applications must be installed, and the data collector must be configured, for every monitored Extreme Scale server. If the data collector is not configured for any server, hung thread detection does not work for it, but all other metrics remain available.

WebSphere Extreme Scale Workspaces

Use workspaces to configure monitoring of an Extreme Scale zone and to view monitoring information.

WebSphere Extreme Scale configuration workspace

Use the WebSphere Extreme Scale configuration workspace to configure monitoring for a WebSphere Extreme Scale zone.

After creating an Extreme Scale zone node using the WXS_Add_Zone Take Action command (see "WXS_Add_Zone: add a zone node" on page 581), you must configure it using this workspace. The monitoring workspaces for the zone do not display any data before you configure it. You can also change this configuration at a later time.

Attention: Before configuring monitoring for a security-enabled WebSphere Extreme Scale environment, you must complete additional configuration steps, as described in the *IBM Tivoli Composite Application Manager: Agent for WebSphere Applications Installation and Configuration Guide*. Otherwise, the monitoring might fail.

This workspace includes the following tabs:

- Connection Settings
- Security Settings
- Collection Settings

Connection Settings tab

Specify WebSphere Extreme Scale zone information, including catalog servers, in this tab. The agent uses the catalog servers to retrieve monitoring information about the zone.

The Connection Settings tab contains the following fields for the zone:

- **Zone Name** The name of the zone to monitor. You can also specify * to monitor aggregated information for all zones that include the listed catalog servers.
- **Connection type** The type of the zone: **Embedded WAS** if catalog servers running within WebSphere application servers, or **Standalone** if it is running as stand-alone containers (with their own JVMs).

It also contains a table of catalog servers, listing all the servers that are configured or discovered. The table shows the following information:

- **Info** Blank if the agent has connected to the server successfully. If the connection has failed, the column contains a red icon. Place the mouse cursor over the icon to display details about the failure.
- **Server Name** The catalog name for the catalog server. If the catalog name detected at the configured host and port is different, the agent marks the catalog server as `OUT_OF_SYNC` in the Catalogs workspace. You can set * as the catalog host. In this case, the agent detects the name when connecting to the server for the first time.
- **Host name** The host name of the catalog server. For best performance, configure WebSphere Extreme Scale monitoring under a node corresponding to a server within the monitored zone, with Catalog Service enabled. In this case, use `localhost` for this field. Alternatively, to monitor a different server, enter its hostname in this field.
- **Port number** The RMI port for the catalog server. The default value is 1099 for WebSphere Extreme Scale stand-alone, 9809 for embedded WebSphere Application Server.

Before you configure a catalog server for the zone, the agent does not monitor the zone. You can only view default settings in the other tabs of this workspace. To start monitoring a zone, you must provide information about at least one catalog serve for the zone. The server must be running and available.

After you configure a server and monitoring starts, the agent discovers other catalog servers. When you open the configuration window again, you can see the discovered servers added to the table.

To configure the settings for a catalog server for the zone, enter the settings in the fields and click **Add**. You can enter * for the server name; in this case, the agent determines the name automatically. You must enter the host name and port number.

To change the settings for a catalog server, select it in the table, modify the settings in the fields and click **Modify**.

To remove a catalog server from the table, select it in the table and click **Remove**. Use this option for catalog servers that are no longer active. If you remove a catalog server that is still active, the agent discovers it later and adds it to the table again.

Important: If the server is monitored by a version of the agent older than 7.2, this workspace contains information for one catalog server only. You can set the zone name, catalog server name and hostname, port number, and connection type.

Attention: If the server is monitored by a version of the agent older than 7.2 and the catalog server name changes after configuration, the agent reports a connection error. To fix this error, use the configuration workspace again.

Security Settings tab

If JMX connection security (SOAP SSL) is enabled on WebSphere Extreme Scale Catalog Service, specify the security and authentication information in this tab.

The Security Settings tab contains the following fields:

- **Connector Security Is Enabled** Select `TRUE` if JMX connection security is enabled on WebSphere Extreme Scale Catalog Service. Otherwise, select `FALSE`; the default setting is `FALSE`.
- **User** The user name for authentication with the WebSphere Extreme Scale Catalog Service JMX connection.
- **Password**, **Confirm Password** The password for authentication with the WebSphere Extreme Scale Catalog Service JMX connection.
- **SSL Trust Store File** The path to the SSL Trust Store File.
- **SSL Trust Store Password**, **Confirm SSL Trust Store Password** The password for the SSL Trust Store.
- **SSL Trust Store Type** The type of the SSL trust store.

Important: All paths in this tab must be set for the server on which the Agent is installed, that is, the server represented by the node under which the WebSphere Extreme Scale zone node is located.

Collection Settings tab

You can modify settings for collection of monitoring information from WebSphere Extreme Scale in this tab.

The Collection Settings tab contains the following fields:

- **Discovery Interval (seconds)** Interval for discovering grid topology changes, such as new container or catalog services added to the zone.
- **Collection Interval (seconds)** Interval for collecting all Extreme Scale metrics from the Catalog Service (using MBeans).
- **Hung ThreadsMonitoring** Enables or disables monitoring of hung threads inside Extreme Scale JVM processes on all servers within the zone.

  Important: For hung thread detection, ITCAM Agent for WebSphere Applications must be installed, and the data collector must be configured, for every monitored Extreme Scale server.
- **Hung Thread Detection Timeout (seconds)** Timeout for detecting hung threads.

Accessing the WebSphere Extreme Scale configuration workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.

- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, select the Extreme Scale zone node of your choice.
- Right-click, and choose **WebSphere XS Zone – Configuration**.

To create an Extreme Scale zone node, use the WXS_Add_Zone Take Action, see "WXS_Add_Zone: add a zone node" on page 581.

WebSphere Extreme Scale Zone Summary workspace

This workspace shows the summary statistics for the monitored Extreme Scale zones. It is available if more than one zone node is configured for an agent node.

This workspace displays data provided by the "Extreme Scale Grids attributes" on page 570,"Extreme Scale Map attributes" on page 572 and "Extreme Scale Servers attributes" on page 568.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all Extreme Scale events on this agent node.
- Transaction Times History graph, which shows the history and real-time value for the average transaction time for every configured zone node.
- Transaction Rates History graph, which shows the history and real-time value for the average transaction rate for every configured zone node.
- Hit Rates History graph, which shows the history and real-time value for the average map hit rate for every configured zone node.
- Available Processors History graph, which shows the history and real-time value for the number of available CPUs in every configured zone node.
- Memory History graph, which shows the history and real-time values for the total used and free JVM heap memory, in megabytes, in every configured zone node.

From all the graphs, you can select the **WebSphere XS Zone** link to open the "WebSphere Extreme Scale Zone workspace" for the selected zone.

Accessing the WebSphere Extreme Scale Zone Summary workspace

Under an Agent for WebSphere Applications node for which two or more Extreme Scale zone nodes are configured, select the **WebSphere XS Zone** node.

WebSphere Extreme Scale Zone workspace

This workspace shows the common statistics for a monitored Extreme Scale zone node.

This workspace displays data provided by the "Extreme Scale Grids attributes" on page 570,"Extreme Scale Map attributes" on page 572 and "Extreme Scale Servers attributes" on page 568.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all Extreme Scale events for this zone node.

- Transaction Times history graph, which shows the history and real-time value for the average transaction time for this zone node.
- Transaction Rate history graph, which shows the history and real-time value for the average transaction rate for this zone node.
- Map Hit Rates graph, which shows the history and real-time value for the average map hit rate for this zone node.
- Available Processors graph, which shows the history and real-time value for the number of available CPUs in this zone node.
- Heap Memory History graph, which shows the history and real-time values for the total used and free JVM heap memory, in megabytes, for this zone node.

Accessing the WebSphere Extreme Scale Zone workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, select the Extreme Scale zone node of your choice.

Catalogs workspace

This workspace shows the statistics for the catalog services running within a monitored Extreme Scale zone node.

This workspace displays data provided by the "Extreme Scale Catalog attributes" on page 565.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for catalog-related Extreme Scale events for this zone node.
- Active Catalogs History graph, displaying the history and real-time value for the numbers of active catalog services and of catalog services that have a quorum.
- Catalog Services table, displaying the attributes and statistics for every catalog service in the zone node.

Accessing the Catalogs workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Catalogs** node.

Containers workspace

This workspace shows the statistics, including hung threads, for the Grid containers running within a monitored Extreme Scale zone node.

This workspace displays data provided by the "Extreme Scale Servers attributes" on page 568 and the "Extreme Scale Thread Pools attributes" on page 578.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all Extreme Scale events for this zone node.
- Container Servers Summary table, displaying the attributes and statistics, including hung threads, for every server JVM in the zone node.
- Highest Memory Used: by Server bar chart, displaying the memory usage for the server JVMs that consume most memory.
- Highest Active Threads: by Server bar chart, displaying the active thread count for the server JVMs that are, or were, running the highest number of threads. Only available for WebSphere Extreme Scale 7.1.

To view detailed statistics for the server, use the following links from the Container Servers Summary table and from the bar charts:

Selected Server To view the "Selected Containers workspace" for the server.

Selected Server – Transactions To view the "Transactions – Selected Level workspace" on page 552 for the server.

Selected Server – Maps To view the "Maps – Selected Level workspace" on page 557 for the server.

Selected Server – Shards To view the "Shards – Selected Level workspace" on page 561 for the server.

Accessing the Containers workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Containers** node.

Selected Containers workspace

This workspace shows the statistics for a Grid container (server) JVM running within a monitored Extreme Scale zone node.

This workspace displays data provided by the "Extreme Scale Servers attributes" on page 568,"Extreme Scale Grids attributes" on page 570, and "Extreme Scale Thread Pools attributes" on page 578.

The predefined workspace contains the following items:

- JVM Memory History graph, displaying the history and real-time value for the free and used JVM memory on the server. Only available for WebSphere Extreme Scale 7.1.
- Maps Memory History graph, displaying the history and real-time value for the estimated memory used by every map running on the server. This value

includes only the memory that every map uses on this server, and not on other servers. Only available for WebSphere Extreme Scale 7.1.

- Transaction Times history graph, which shows the average transaction time history and real-time value for the server.
- Threads Activity history graph, which shows the active thread count and maximum thread pool size history and real-time values for the server. Only available for WebSphere Extreme Scale 7.1.
- Server Summary table, displaying the attributes and statistics for the server.

To view detailed statistics for the server, select any item in table or the graphs, right click, and select one of the following links:

- **Selected Server – Transactions** To see the "Transactions – Selected Level workspace" on page 552 workspace for the server.
- **Selected Server – Maps** To see the "Maps – Selected Level workspace" on page 557 workspace for the server.
- **Selected Server – Shards** To see the "Shards – Selected Level workspace" on page 561 workspace for the server.

Accessing the Selected Containers workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Containers** node.
- In the "Containers workspace" on page 548, select a server in a table or bar chart, right click, and select **Selected Server**.

Grids workspace

This workspace shows the statistics for the Grids running within a monitored Extreme Scale zone node.

This workspace displays data provided by the "Extreme Scale Placements attributes" on page 577.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all grid-related Extreme Scale events for this zone node.
- Outstanding Work Items History graph, displaying the history and current values for the number of outstanding work items (that is, requests that have not yet been completed) for every grid within the zone node.
- Active Containers History graph, displaying the history and current values for the number of active containers (JVM server processes) for every grid within the zone node.
- Grids Summary table, displaying the attributes and statistics for every combination of grid and map set in the zone node.

To view detailed statistics for any grid, use the following links from the Grids Summary table and from the graphs:

Selected Grid – Transactions To view the "Transactions – Selected Level workspace" on page 552 for the grid.

Selected Grid – Maps To view the "Maps – Selected Level workspace" on page 557 for the grid.

Selected Grid – Shards To view the "Shards – Selected Level workspace" on page 561 for the grid.

Accessing the Grids workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Grids** node.

Transactions workspace

This workspace shows information on grid and server transaction rates and transaction response times in a monitored Extreme Scale zone node. The information is aggregated on the grid level (aggregated across servers) and on the server level.

This workspace displays data provided by the "Extreme Scale Grids attributes" on page 570.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all transaction-related Extreme Scale events for the zone node.
- Transaction Times History: Grids graph, displaying the history and real-time values for average transaction times for every grid in the zone node.
- Transaction Rates History: Grids graph, displaying the history and real-time values for average transaction rates for every grid in the zone node.
- Longest Transaction Times: by Server bar chart, displaying the servers with the highest average transaction time values for this zone node. In this table, you can see the specific server JVMs that have, or recently had, the longest transaction times.
- Highest Transaction Rates: by Server bar chart, displaying the servers with the highest transaction rate values for this zone node. In this table, you can see the specific server JVMs that have, or recently had, the highest transaction rates.

To view detailed transaction statistics for a grid, select the grid in the **Transaction Times History: Grids** or the **Transaction Rates History: Grids** graph. Then right click and select **Selected Grid – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the grid.

To view detailed transaction statistics for a server, select the server in the **Longest Transaction Times: by Server** or the **Highest Transaction Rates: by Server** bar

chart. Then right click and select **Selected Server – Transactions**. The system displays "Transactions – Selected Level workspace" for the server.

To view container details for a server, select the server in the **Longest Transaction Times: by Server** or the **Highest Transaction Rates: by Server** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

Accessing the Transactions workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Transactions** node.

Transactions – Selected Level workspace

This workspace shows information about transaction rates and transaction response times in a selected server, grid, or map set within the monitored Extreme Scale environment.

This workspace displays data provided by the "Extreme Scale Grids attributes" on page 570.

The exact content of the workspace depends on whether it is invoked for a server, grid, or map set.

Transactions – Selected Grid

This workspace contains transaction data for a grid, aggregated from all servers.

The predefined workspace contains the following items:

- Transactions Summary table, which shows the attributes and statistics for map sets available in the grid.
- Transaction Rates History graph, displaying the history and real-time values for average transaction rates for every map set in the grid.
- Transaction Times History graph, displaying the history and real-time values for average transaction times for every map set in the grid.
- Highest Transaction Rates bar chart, displaying the transaction rate on servers that have, or recently had, the highest transaction rates.
- Longest Transaction Times bar chart, displaying the transaction time on servers that have, or recently had, the longest transaction times.

To view detailed transaction statistics for a map set, select the map set in the **Transactions Summary** table or in the **Transaction Rates History** or the **Transaction Times History** graph. Then right click and select **Selected Map Set – Transactions**. The system displays the "Transactions – Selected Level workspace" for the map set.

To view detailed transaction statistics for a server, select the server in the **Highest Transaction Rates** or the **Longest Transaction Times** bar chart. Then right click and select **Selected Server – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the server.

To view container details for a server, select the server in the **Highest Transaction Rates** or the **Longest Transaction Times** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view map information for the same grid, select any row in the **Transactions Summary** table. Then right click and select **Selected Level – Maps**. The system displays the "Maps – Selected Level workspace" on page 557 for the grid.

To view shard information for the same grid, select any row in the **Transactions Summary** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Level workspace" on page 561 for the grid.

Transactions – Selected Server

This workspace contains transaction data for a single server.

The predefined workspace contains the following items:

- Transactions Summary table, which shows the attributes and summary transaction statistics for map sets available in the server.
- Transaction Rates History graph, displaying the history and real-time values for average transaction rates for every map set on the server.
- Transaction Times History graph, displaying the history and real-time values for average transaction times for every map set on the server.
- Highest Transaction Rates bar chart, displaying the transaction rate on shards that have, or recently had, the highest transaction rates.
- Longest Transaction Times bar chart, displaying the transaction time on shards that have, or recently had, the longest transaction times.

To view detailed transaction statistics for a map set, select the map set in the **Transactions Summary** table or in the **Transaction Rates History** or the **Transaction Times History** graph. Then right click and select **Selected Map Set – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the map set.

To view detailed transaction statistics for a shard, select the shard in the **Highest Transaction Rates** or the **Longest Transaction Times** bar chart. Then right click and select **Selected Partition – Transactions**. The system displays the "Transactions – Selected Partition workspace" on page 555 for the partition corresponding to the shard.

To view container details for the server, select any shard in the **Highest Transaction Rates** or **Longest Transaction Times** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view map information for the same server, select any row in the **Transactions Summary** table. Then right click and select **Selected Level – Maps**. The system displays the "Maps – Selected Level workspace" on page 557 for the server.

To view shard information for the same server, select any row in the **Transactions Summary** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Level workspace" on page 561 for the server.

Transactions – Selected Map Set

This workspace displays transaction data for a map set aggregated among all servers (if called from a Selected Grid workspace) or on one server (if called from a Selected Server workspace).

The predefined workspace contains the following items:

- Transactions Summary table, which shows the attributes and summary transaction statistics for the map set. This table has one row.
- Transaction Rates History graph, displaying the history and real-time value for the average transaction rate for the map set.
- Transaction Times History graph, displaying the history and real-time value for average transaction time for the map set.
- Highest Transaction Rates bar chart, displaying the transaction rate for the shards that have, or recently had, the highest transaction rates.
- Longest Transaction Times bar chart, displaying the transaction time for the shards that have, or recently had, the longest transaction times.

To view detailed transaction statistics for a shard, select the shard in the **Highest Transaction Rates** or the **Longest Transaction Times** bar chart. Then right click and select **Selected Partition – Transactions**. The system displays the "Transactions – Selected Partition workspace" on page 555 for the partition corresponding to the shard.

To view container details for the server running a shard, select the shard in the **Highest Transaction Rates** or the **Longest Transaction Times** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view shard information for the same map set, select the row in the **Transactions Summary** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Level workspace" on page 561 for the map set.

Accessing the Transactions – Selected Level workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Transactions** node.
- Complete one of the following steps:
  - To view the **Transactions – Selected Level** workspace for a grid, select the grid in the **Transaction Times History: Grids** or the **Transaction Rates History: Grids** graph. Then right click and select **Selected Grid – Transactions**.

  - To view the **Transactions – Selected Level** workspace for a server, select the server in the **Longest Transaction Times: by Server** or the **Highest Transaction Rates: by Server** bar chart. Then right click and select **Selected Server – Transactions**
- To view the **Transactions – Selected Level** workspace for a map set, view it for a grid or server as described in the previous step. Then, select the map set in the **Transactions Summary** table, right click, and select **Selected Map Set – Transactions**.

Transactions – Selected Partition workspace

This workspace shows information about transaction rates and transaction response times in a selected partition within the monitored Extreme Scale environment.

This workspace displays data provided by the "Extreme Scale Grids attributes" on page 570.

The predefined workspace contains the following items:

- Transactions Summary table, which shows the attributes and summary transaction statistics for the partition shards, that is, the shards that belonged to the partition within the history time span for the workspace (by default, the last 2 hours).
- Transaction Rates History graph, displaying the history and real-time values for average transaction rates for the partition shards.
- Transaction Times History graph, displaying the history and real-time values for average transaction times for the partition shards.

To view container details for the server running a shard, select the shard in the **Transactions Summary** table. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view map information for the same partition, select any row in the **Transactions Summary** table. Then right click and select **Selected Level – Maps**. The system displays the "Maps – Selected Partition workspace" on page 559 for the partition.

To view shard information for the same partition, select any row in the **Transactions Summary** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Partition workspace" on page 564 for the partition.

Accessing the Transactions – Selected Partition workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Transactions** node.

- Select a server in the **Longest Transaction Times: by Server** or the **Highest Transaction Rates: by Server** bar chart. Then right click and select **Selected Server – Transactions**.
- Select a shard in the **Highest Transaction Rates** or the **Longest Transaction Times** bar chart. Then right click and select **Selected Partition – Transactions**.

Maps workspace

This workspace shows information about map counts and map hit rates in a monitored Extreme Scale zone node. The information is aggregated on the grid and server levels.

This workspace displays data provided by the "Extreme Scale Map attributes" on page 572.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all map-related Extreme Scale events for this zone node.
- Hit Rates History: Grids graph, displaying the history and real-time values for average map hit rates for every grid in the zone node.
- Map Counts History: Grids graph, displaying the history and real-time values for map counts for every grid in the zone node.
- Lowest Hit Rates: by Server bar chart, displaying hit rate for the server JVMs that have, or recently had, the lowest hit rates.
- Highest Map Counts: by Server bar chart, displaying the map count for the server JVMs that have, or recently had, the highest map counts.

To view detailed map statistics for a grid, select the grid in the **Hit Rates History: Grids** or the **Map Counts History: Grids** graph. Then right click and select **Selected Grid – Maps**. The system displays the "Maps – Selected Level workspace" on page 557 for the grid.

To view detailed map statistics for a server, select the server in the **Lowest Hit Rates: by Server** or the **Highest Map Counts: by Server** bar chart. Then right click and select **Selected Server – Maps**. The system displays the "Maps – Selected Level workspace" on page 557 for the server.

To view container details for a server, select the server in the **Lowest Hit Rates: by Server** or the **Highest Map Counts: by Server** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

Accessing the Maps workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Maps** node.

Maps – Selected Level workspace

This workspace shows information about map counts and hit rates in a selected server, grid, or map set within the monitored Extreme Scale environment.

This workspace displays data provided by the "Extreme Scale Map attributes" on page 572.

The exact content of the workspace depends on whether it is invoked for a server, grid, or map set.

Maps – Selected Grid

This workspace contains map data for a grid, aggregated from all servers.

The predefined workspace contains the following items:

- Maps Summary table, which shows the attributes and statistics for maps available in the grid.
- Hit Rates History graph, displaying the history and real-time values of the map hit rate for every map in the grid.
- Map Counts History graph, displaying the history and real-time values of the map count for every map in the grid.
- Lowest Hit Rates bar chart, displaying the hit rate for the server JVMs that have, or recently had, the lowest hit rates.
- Highest Map Counts bar chart, displaying the map count for the server JVMs that have, or recently had, the highest map counts.

To view detailed map statistics for a map, select the map in the **Maps Summary** table or in the **Hit Rates History** or the **Map Counts History** graph. Then right click and select **Selected Map – Maps**. The system displays the"Maps – Selected Level workspace" for the map.

To view detailed map statistics for a server, select the server in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click and select **Selected Server – Maps**. The system displays the "Maps – Selected Level workspace" for the server.

To view container details for a server, select the server in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same grid, select any row in the **Maps Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the grid.

To view shard information for the same grid, select any row in the **Selected Maps** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Level workspace" on page 561 for the grid.

Maps – Selected Server

This workspace contains map data for a single server.

The predefined workspace contains the following items:

- Maps Summary table, which shows the attributes and statistics for maps available on the server.
- Hit Rates History graph, displaying the history and real-time values of the map hit rate for every map on the server.
- Map Counts History graph, displaying the history and real-time values of the map count for every map on the server.
- Lowest Hit Rates bar chart, displaying the hit rate for the shards on this server that have, or recently had, the lowest map hit rates.
- Highest Map Counts bar chart, displaying map count for the shards on this server that have, or recently had, the highest map counts.

To view detailed map statistics for a map, select the map in the **Maps Summary** table or in the **Hit Rates History** or the **Map Counts History** graph. Then right click and select **Selected Map – Maps**. The system displays the"Maps – Selected Level workspace" on page 557 for the map.

To view detailed map statistics for a shard, select the shard in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click and select **Selected Partition – Maps**. The system displays the "Maps – Selected Partition workspace" on page 559 for the partition corresponding to the shard.

To view container details for the server, select any shard in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same server, select any row in the **Maps Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the server.

To view shard information for the same server, select any row in the **Maps Summary** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Level workspace" on page 561 for the server.

Maps – Selected Map

This workspace displays map data for a map aggregated among all servers (if called from a Selected Grid workspace) or on one server (if called from a Selected Server workspace).

The predefined workspace contains the following items:

- Maps Summary table, which shows the attributes and statistics for the map. This table has one row.
- Hit Rates History graph, displaying the history and real-time value of the map hit rate for the map.
- Map Counts History graph, displaying the history and real-time value of the map count for the map.
- Lowest Hit Rates bar chart, displaying the hit rate for the shards that have, or recently had, the lowest map hit rates.
- Highest Map Counts bar chart, displaying the map count for the shards that have, or recently had, the highest map counts.

To view detailed map statistics for a shard, select the shard in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click, and select **Selected Partition – Maps**. The system displays the "Maps – Selected Partition workspace" for the partition corresponding to the shard.

To view container details for the server running a shard, select the shard in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

Accessing the Maps – Selected Level workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Maps** node.
- Perform one of the following steps:
  - To view the Maps – Selected Level workspace for a grid, select the grid in the **Hit Rates History: Grids** or the **Map Counts History: Grids** graph. Them right click and select **Selected Grid – Maps**.
  - To view the Maps – Selected Level workspace for a server, select the server in the **Lowest Hit Rates: by Server** or the **Highest Map Counts: by Server** bar chart. Then right click and select **Selected Server – Maps**.
- To view the Maps – Selected Level workspace for a map, view it for a grid or server as described in the previous step. Then, select the map in the **Maps Summary** table, right click, and select **Selected Map – Maps**.

Maps – Selected Partition workspace

This workspace shows information about map counts and hit rates in a selected partition within the monitored Extreme Scale environment.

This workspace displays data provided by the "Extreme Scale Map attributes" on page 572.

The predefined workspace contains the following items:

- Maps Summary table, which shows the attributes and summary map statistics for the partition shards, that is, the shards that belonged to the partition within the history time span for the workspace (by default, the last 2 hours).
- Hit Rates History graph, displaying the history and real-time value of the map hit rate for the partition shards.
- Map Counts History graph, displaying the history and real-time value of the map count for the partition shards.
- Batch Update Times History graph, displaying the history and real-time value of the batch update time for the partition shards.
- Used Memory History graph, displaying the history and real-time amount of the JVM memory used by the partition shards.

To view container details for the server running a shard, select the shard in the **Maps Summary** table. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same partition, select any row in the **Maps Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the partition.

To view shard information for the same partition, select any row in the **Selected Maps** table. Then right click and select **Selected Level – Shards**. The system displays the "Shards – Selected Partition workspace" on page 564 for the partition.

Accessing the Maps – Selected Partition workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Maps** node.
- Select a server in the **Lowest Hit Rates: by Server** or the **Highest Map Counts: by Server** bar chart. Then right click and select **Selected Server – Maps**.
- Select a shard in the **Lowest Hit Rates** or the **Highest Map Counts** bar chart. Then right click and select **Selected Partition – Maps**.

Shards workspace

This workspace shows information about shard request rates and active request counts in a monitored Extreme Scale zone node. The information is aggregated on the grid and server levels.

This workspace displays data provided by the "Extreme Scale Shards attributes" on page 575.

The predefined workspace contains the following items:

- Situation Event Console, which shows the event log for all shard-related Extreme Scale events for this zone node.
- Request Rates History: Grids graph, displaying the history and real-time values for average shard request rates for every grid in the zone node.
- Active Requests History: Grids graph, displaying the history and real-time values for average shard active requests count for every grid in the zone node.
- Highest Request Rates: by Server bar chart, displaying the average shard request rates for the server JVMs that have, or recently had, the highest shard request rates.
- Highest Active Requests: by Server bar chart, displaying the average shard active request counts for the server JVMs that have, or recently had, the highest shard active request counts.

To view detailed shard statistics for a grid, select the grid in the **Request Rates History: Grids** or the **Active Requests History: Grids** graph. Then right click and select **Selected Grid – Shards**. The system displays the "Shards – Selected Level workspace" for the grid.

To view detailed shard statistics for a server, select the server in the **Highest Request Rates: by Server** or the **Highest Active Requests: by Server** bar chart. Then right click and select **Selected Server – Shards**. The system displays the "Shards – Selected Level workspace" for the server.

Accessing the Shards workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Shards** node.

Shards – Selected Level workspace

This workspace shows information about shard request rates and active request counts in a selected server, grid, or map set within the monitored Extreme Scale environment.

This workspace displays data provided by the "Extreme Scale Shards attributes" on page 575.

The exact content of the workspace depends on whether it is invoked for a server, grid, or map set.

Shards – Selected Grid

This workspace contains shard data for a grid, aggregated from all servers.

The predefined workspace contains the following items:

- Shards Summary table, which shows the attributes and summary shard statistics for map sets available in the grid.
- Request Rates History graph, displaying the history and real-time values for average shard request rates for every map set in the grid.
- Active Requests History graph, displaying the history and real-time values for average shard active requests count for every map set in the grid.
- Highest Request Rates bar chart, displaying the shard request rate for the server JVMs that have, or recently had, the highest shard request rates.
- Highest Active Requests bar chart, displaying the shard active request count for the server JVMs that have, or recently had, the highest shard active request counts.

To view detailed shard statistics for a map set, select the map set in the Shards Summary table or in the Request Rates History or Active Requests History graph. Then select **Selected Map Set – Shards**. The system displays the"Shards – Selected Level workspace" on page 561 for the map.

To view detailed shard statistics for a server, select the server in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Server – Shards** . The system displays the "Shards – Selected Level workspace" on page 561 for the server.

To view container details for a server, select the server in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same grid, select any row in the **Shards Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the grid.

To view map information for the same grid, select any row in the **Shards Summary** table. Then right click and select **Selected Level – Maps**. The system displays the "Maps – Selected Level workspace" on page 557 for the grid.

Shards – Selected Server

This workspace contains shard data for a single server.

The predefined workspace contains the following items:

- Shards Summary table, which shows the attributes and summary shard statistics for map sets available on the server.
- Request Rates History graph, displaying the history and real-time values for average shard request rates for every map set on the server.
- Active Requests History graph, displaying the history and real-time values for average shard active requests count for every map set on the server.
- Highest Request Rates bar chart, displaying the request rate for the shards that have, or recently had, the highest request rates.
- Highest Active Requests bar chart, displaying the active request count for the shards that have, or recently had, the highest shard active request counts.

To view detailed shard statistics for a map set, select the map set in the **Shards Summary** table or in the **Request Rates History** or the **Active Requests History** graph. Then right click and select **Selected Map Set – Shards**. The system displays the"Shards – Selected Level workspace" on page 561 for the map.

To view detailed shard statistics for a shard, select the shard in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Partition – Shards**. The system displays the "Shards – Selected Partition workspace" on page 564 for the partition corresponding to the shard.

To view container details for the server, select any shard in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same server, select any row in the **Shards Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the server.

To view map information for the same server, select any row in the **Shards Summary** table. Then right click and select **Selected Level – Maps**. The system displays the "Maps – Selected Level workspace" on page 557 for the server.

Shards – Selected Map Set

This workspace displays shard data for a map set aggregated among all servers (if called from a Selected Grid workspace) or on one server (if called from a Selected Server workspace).

The predefined workspace contains the following items:

- Shards Summary table, which shows the attributes and summary shard statistics for the map set. This table has one row.
- Request Rates History graph, displaying the history and real-time value for average shard request rates for this map set.
- Active Requests History graph, displaying the history and real-time value for average shard active requests count for this map set.
- Highest Request Rates bar chart, displaying the request rate for the shards in this map set that have, or recently had, the highest request rates.
- Highest Active Requests bar chart, displaying the active request counts for the shards in this map set that have, or recently had, the highest shard active request counts.

To view detailed shard statistics for a shard, select the shard in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Partition – Shards**. The system displays the "Shards – Selected Partition workspace" on page 564 for the partition corresponding to the shard.

To view container details for the server running a shard, select the shard in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same map set, select the row in the **Shards Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the map set.

Accessing the Shards – Selected Level workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.
- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.

- Under the Extreme Scale zone node of your choice, select the **Shards** node.
- Perform one of the following steps:
  - To view the **Shards – Selected Level** workspace for a grid, select the grid in the **Request Rates History: Grids** or the **Active Requests History: Grids** graph. Then right click and select **Selected Grid – Shards**.
  - To view the **Shards – Selected Level** workspace for a server, select the server in the **Highest Request Rates: by Server** or the **Highest Active Requests: by Server** bar chart. Then right click and select **Selected Server – Shards**.
- To view the Shards – Selected Level workspace for a map set, view it for a grid or server as described in the previous step. Then, select the map set in the **Shards Summary** table, right click, and select **Selected Map Set – Shards**.

Shards – Selected Partition workspace

This workspace shows information about shard request rates and active request counts in a selected partition within the monitored Extreme Scale environment.

This workspace displays data provided by the "Extreme Scale Shards attributes" on page 575.

The predefined workspace contains the following items:

- Shards Summary table, which shows the attributes and summary shard statistics for the partition shards, that is, the shards that belonged to the partition within the history time span for the workspace (by default, the last 2 hours)..
- Request Rates History graph, displaying the history and real-time value for request rates for the partition shards.
- Active Requests History graph, displaying the history and real-time value for active requests count for the partition shards.
- Forwarded Requests History graph, displaying the history and real-time value for forwarded requests count for the partition shards.

To view container details for the server running a shard, select the shard in the **Shards Summary** table. Then right click and select **Selected Server – Containers**. The system displays the "Selected Containers workspace" on page 549 for the server.

To view transaction information for the same partition, select the row in the **Shards Summary** table. Then right click and select **Selected Level – Transactions**. The system displays the "Transactions – Selected Level workspace" on page 552 for the partition.

To view map information for the same partition, select the row in the **Shards Summary** table. Then right click and select **Selected Level – Maps**. The system displays the "Maps – Selected Partition workspace" on page 559 for the partition.

Accessing the Shards – Selected Partition workspace

To access the workspace, complete the following steps:

- Within the Navigator, expand Windows Systems, Linux Systems, z/OS Systems, or UNIX Systems, as appropriate for the node you are monitoring.
- Within the node list, expand the entry that corresponds to the server node.
- Within that node list of monitored applications, expand the WebSphere agent.
- Under the WebSphere agent, expand **WebSphere XS Zone**.

- Under the **WebSphere XS Zone**, expand the Extreme Scale zone node of your choice.
- Under the Extreme Scale zone node of your choice, select the **Shards** node.
- Select a server in the **Highest Request Rates: by Server** or the **Highest Active Requests: by Server** bar chart. Then right click and select **Selected Server – Shards**.
- Select a shard in the **Highest Request Rates** or the **Highest Active Requests** bar chart. Then right click and select **Selected Partition – Shards**.

WebSphere Extreme Scale Attributes

IBM Tivoli Composite Application Manager Agent for WebSphere Applications uses several attribute groups to store data for WebSphere Extreme Scale monitoring.

Extreme Scale Catalog attributes

The **Extreme Scale Catalog** attributes provide information about the catalog service for the WebSphere Extreme Scale grid.

The attributes within this group are used to build the "Catalogs workspace" on page 548.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 32 characters.

Sample Date and Time The date and time at which the agent has collected this data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 79. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time The length of the sample interval in seconds. The valid format is a positive integer.

Server Name The name of the catalog service. The valid format is an alphanumeric string, with a maximum of 48 characters.

Host Name The host name where the catalog service is running. The valid format is an alphanumeric string, with a maximum of 32 characters.

JMX URL The URL for the catalog service JMX connection. The valid format is an alphanumeric string, with a maximum of 192 characters.

Grid Servers The combined number of grid processes (catalog services and container servers) running in the grid. The valid format is a positive integer. For

WebSphere Extreme Scale 7.1 this attribute is collected from the server; for WebSphere Extreme Scale 6.1 and 7.0, the Agent calculates the value by tracking active catalog services in the grid.

Heart Beat Frequency Level The heart beat frequency level, as reported by the WebSphere Extreme Scale MBean. For WebSphere Extreme Scale 6.1 and 7.0, the valid values are `TYPICAL`,`RELAXED` and `AGGRESSIVE`. This information is not available for WebSphere Extreme Scale 7.1; for this version the value is always `LEVEL_OFF`.

Active Catalogs The number of active catalog service processes monitored in the grid, reported by WebSphere Extreme Scale 7.1 MBean or calculated by agent for back-level versions. The valid format is a positive integer. For WebSphere Extreme Scale 7.1 this attribute is collected from the server; for WebSphere Extreme Scale 6.1 and 7.0, the Agent calculates the value.

Quorum Catalogs The number of quorum catalog service processes monitored in the grid. The valid format is a positive integer. This information is only available for WebSphere Extreme Scale 7.1. In WebSphere Extreme Scale 6.1 and 7.0, ignore this value as the WebSphere Extreme Scale MBean does not provide the data.

Is Primary Whether this is the primary catalog service for workspace reports and situation formula checks. The valid values are `YES` and `NO`. The agent discovers the primary catalog service for the grid automatically; the catalog service configured in the agent might not be the primary service.

Attention: For WebSphere Extreme Scale 6.1, this field is set to YES for the randomly selected catalog service that the agent monitors.

Connection Status Agent connection status for the catalog service. The valid values are `ONLINE` and `OFFLINE`; if the value is `OFFLINE`, monitoring information might not be up to date.

Configuration Status Agent configuration status for the catalog service. The valid values are:

- `YES`: this is the catalog service configured in the agent, using the "WebSphere Extreme Scale configuration workspace" on page 544.
- `NO`: this catalog service was automatically discovered via the initially configured service. The agent discovers all catalog services in the grid node.
- `ERROR`: the catalog service connection is not configured. When a zone node is created but not yet configured, or configured incorrectly, no monitoring is performed, but one row is available in the Catalog attributes table with Configuration Status set to `ERROR`.

Quorum Status Indicates the quorum status for the catalog service domain. The valid values are `DISABLE`, `NORMAL`, `WAITING` ,`OVERRIDE`, and `INCONSISTENT`. For WebSphere Extreme Scale 6.1 and 7.0 this attribute is collected from the server; for WebSphere Extreme Scale 7.1, the Agent calculates the value (`DISABLE`, `NORMAL` or `WAITING`).

Monitored Servers The number of container servers on this catalog service that are monitored by the agent. This can be less than the total number of servers on the catalog service; for example, if the catalog service includes servers in multiple zones, but only one zone is configured in the agent, only servers in that zone is included in this value. The valid format is a positive integer.

Extreme Scale Containers attributes

The **Extreme Scale Containers** attributes provide statistics for grid shards running within a Grid Container on a server.

The attributes within this group are not used for any predefined workspace.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time at which the agent has collected data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 80. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time The length of the sample interval in seconds. The valid format is a positive integer.

Server Name The name of the server. The valid format is an alphanumeric string, with a maximum of 48 characters.

Container Name The name of this Grid Container instance. The valid format is an alphanumeric string, with a maximum of 128 characters.

Zone Name The Zone name for this container on this server. The valid format is an alphanumeric string, with a maximum of 128 characters.

Domain Name Domain name of the grid instance within the virtual machine. The valid format is an alphanumeric string, with a maximum of 128 characters. Only available for WebSphere Extreme Scale 7.1.

Core Group The name of core group. The valid format is an alphanumeric string, with a maximum of 32 characters.

Stats Name The name of the map. The valid format is an alphanumeric string, with a maximum of 32 characters.

Active Shard Count The number of active Shards hosted by this container instance. The valid format is a positive integer.

Activated Shard Counts The number of Shards activated in this container instance during the collection interval. The valid format is a positive integer.

Deactivated Shard Count The number of Shards deactivated in this container instance during the collection interval. The valid format is a positive integer.

Stats Level The level of the statistics. The valid values are:

- `CONTAINER`: metrics for a single container.
- `SERVER`: metrics for a single server.
- `COREGROUP`: aggregated metrics for servers in a core group.
- `ZONE`: aggregated metrics for servers in a zone.

Extreme Scale Servers attributes

The **Extreme Scale Servers** attributes provide statistics for a JVM process that hosts all grid components on a server, or aggregated information for JVM processes on several servers.

The attributes within this group are used to build the "WebSphere Extreme Scale Zone Summary workspace" on page 547,"WebSphere Extreme Scale Zone workspace" on page 547"Containers workspace" on page 548 and "Selected Containers workspace" on page 549.

Origin Node The grid zone subnode name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Sample Date and Time The date and time at which the agent has collected data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 81. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time The length of the sample interval in seconds. The valid format is a positive integer.

Server Name The server name. The valid format is an alphanumeric string, with a maximum of 48 characters.

Core Group The name of the core group in which the server is running. The valid format is an alphanumeric string, with a maximum of 32 characters.

Host Name The name of the host on which the server runs.The valid format is an alphanumeric string, with a maximum of 32 characters.

Zone Name The WebSphere Extreme Scale grid zone name. (This is the name configured in Extreme Scale, even if the agent zone node name is different). The valid format is an alphanumeric string, with a maximum of 32 characters.

Available Processors The number of available processors on the server. The valid format is a 4-byte integer.

Free Memory Amount of free memory available on the server JVM process, in megabytes. The valid format is a decimal (formatted to three decimal places).

Max Memory Amount of max memory available on the server JVM process, in megabytes. The valid format is a decimal (formatted to three decimal places).

Used Memory Amount of used memory on the server JVM process, in megabytes. The valid format is a decimal (formatted to three decimal places).

Stats Spec Shows which statistics are available for the server. Normally set to `statsSpec=all=enabled`, meaning that all statistics are available, The valid format is an alphanumeric string, with a maximum of 256 characters. Only available for WebSphere Extreme Scale 7.1.

Is Safe To Shutdown YES if replicas exists for each primary shard hosted by the server. The valid values are `YES` ena `NO`.

Trace Spec The trace specification for the server, in the format Component = Level = [enable|disable]; for example, *=all=enable - tracing is enabled for all components at all levels. The valid format is an alphanumeric string, with a maximum of 256 characters. Only available for WebSphere Extreme Scale 7.1.

WebSphere Server SubNode If the server is also monitored by ITCAM Agent for WebSphere Applications, and is represented by a node in Tivoli Enterprise Portal, this field contains the agent subnode name for this server. Using this value, you can link from Extreme Scale workspaces directly into agent workspaces for the server. The valid format is an alphanumeric string, with a maximum of 32 characters.

Heap Used Percent The percentage of used JVM memory. The valid format is a decimal (formatted to one decimal place).

Connection Status The agent status for the Catalog Service connection. The valid values are `ONLINE` and `OFFLINE`.

Is Leader Whether the server is the leader in a core group. The valid values are `YES` or `NO`.

Stats Level The level of the statistics. The valid values are:

- `SERVER`: metrics for a single server.
- `COREGROUP`: aggregated metrics for servers in a core group.
- `ZONE`: aggregated metrics for servers in a zone.

Hung Threads Blocked Number of hung threads detected in the BLOCKED state. The valid format is a positive integer.

Hung Threads Blocked Number of hung threads detected in the WAITING state. The valid format is a positive integer.

Hung Threads Blocked Number of hung threads detected in the TIMED_WAITING state. The valid format is a positive integer.

Hung Threads Total Total number of hung threads detected on the server or several servers. The valid format is a positive integer.

Extreme Scale Grids attributes

The **Extreme Scale Grids** attributes provide common statistics for a grid instance running on a virtual machine process, or a shard or map set within this grid, or aggregated statistics across all grids in a container. These statistics can also be aggregated from several servers.

The attributes within this group are used to build the "WebSphere Extreme Scale Zone Summary workspace" on page 547, "WebSphere Extreme Scale Zone workspace" on page 547 "Selected Containers workspace" on page 549 "Transactions workspace" on page 551, "Transactions – Selected Level workspace" on page 552 "Transactions – Selected Partition workspace" on page 555

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time at which the agent has collected this data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 82. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time The length of the sample interval in seconds. The valid format is a positive integer.

Statistics Level The level at which statistics are aggregated in the table. The valid values are:

- `SHARD`: raw statistics for an individual shard
- `PARTITION`: summary across all shards in a partition. Not used in the current version.
- `MAP`: summary across all partitions within a map.
- `MAPSET`: summary across all partitions within a map set.
- `GRID`: summary across all map sets within a Grid.
- `DOMAIN`: summary across all Grid Containers within a Grid Domain,

Scope Level The servers from which the statistic information is collected for the table. The valid values are:

- `SHARD`: statistics from a single shard, on one server
- `CONTAINER`: statistics from one container
- `SERVER`: statistics from one server
- `COREGROUP`: statistics from servers within a core group
- `ZONE`: statistics from servers within a zone

Grid Name The grid name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Server Name The server name. The valid format is an alphanumeric string, with a maximum of 48 characters.

Core Group Name The core group name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Statistics Name The name of the unit for which the information is aggregated in this row. For example, if the information is aggregated for a zone, this is the zone name. If the information is aggregated for a map set, this is the map set name, and so on. The valid format is an alphanumeric string, with a maximum of 48 characters.

Statistics ID The unique identifier for this statistics row. Used for identifying the statistic that triggers a situation. The valid format is an alphanumeric string, with a maximum of 96 characters.

Partition ID The partition ID for this Grid Shard, collected for SHARD aggregation level only. The valid format is a positive integer.

Summary Level The aggregation level and scope for this row. This field determines the values of the **Statistics Level** and **Statistics Scope** fields. The valid values are:

- SHARD: raw statistics for an individual shard. Level is SHARD, Scope is SHARD.
- PARTITION: aggregated information for all shards in a partition. Level is PARTITION, Scope is ZONE
- MAP: aggregated information for a map, on all servers within a zone. Level is MAPSET, Scope is ZONE.
- MAPSET: aggregated information for a map set, on all servers within a zone. Level is MAPSET, Scope is ZONE.
- GRID: aggregated information for a grid, on all servers within a zone. Level is GRID, Scope is ZONE.
- DOMAIN: aggregated information for a domain, on all servers within a zone. Level is DOMAIN, Scope is ZONE.
- MAP_SRV: aggregated information for a map, on a single server. Level is MAP, Scope is SERVER.
- MAPSET_SRV: aggregated information for a map set, on a single server. Level is MAPSET, Scope is SERVER.
- GRID_SRV: aggregated information for a grid, on a single server. Level is GRID, Scope is SERVER.
- DOMAIN_SRV: aggregated information for a domain, on a single server. Level is DOMAIN, Scope is SERVER.
- MAP_CG: aggregated information for a map, on all servers within a core group. Level is MAP, Scope is COREGROUP.
- MAPSET_CG: aggregated information for a map set, on all servers within a core group. Level is MAPSET, Scope is COREGROUP.
- GRID_CG: aggregated information for a grid, on all servers within a core group. Level is GRID, Scope is COREGROUP.
- DOMAIN_CG: aggregated information for a domain, on all servers within a core group. Level is DOMAIN, Scope is COREGROUP.

MapSet Name The map set name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Domain Name The domain name for the grid instance within the server JVM. The valid format is an alphanumeric string, with a maximum of 32 characters. Only available for WebSphere Extreme Scale 7.1.

Important: Some name fields might be set to `[Summary]` at various aggregation levels. For example, **Server Name** is `[Summary]` if information is aggregated from multiple servers.

Transaction Count The number of transactions processed by the Grid Shard or all Shards during the collection interval. The valid format is a positive integer.

Transaction Time The total transaction processing time spent by the Grid Shard or all Shards during the collection interval. The valid format is a positive integer.

Transaction Rate The average rate, per second, at which the Grid Shard or all Shards were processing transactions during the collection interval. The valid format is a decimal (formatted to three decimal places).

Extreme Scale Map attributes

The **Extreme Scale Map** attributes provide statistics for a particular map running within a grid container, or aggregated statistics for several maps. A map runs on a particular server, within a grid and a map set; it is associated with one or several partitions on the server.

The attributes within this group are used to build the “WebSphere Extreme Scale Zone Summary workspace” on page 547,“WebSphere Extreme Scale Zone workspace” on page 547“Maps workspace” on page 556“Maps – Selected Level workspace” on page 557 and “Maps – Selected Partition workspace” on page 559.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time at which the agent has collected data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 83. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time Indicates the length of the sample interval in seconds. The valid format is a positive integer.

Statistics Level The level at which statistics are aggregated in the table. The valid values are:

- SHARD: raw statistics for an individual shard
- PARTITION: summary across all shards in a partition. Not used in the current version.
- MAP: summary across all partitions within a map.
- MAPSET: summary across all partitions within a map set.
- GRID: summary across all map sets within a Grid.
- DOMAIN: summary across all Grid Containers within a Grid Domain,

Scope Level The servers from which the statistic information is collected for the table. The valid values are:

- SHARD: statistics from a single shard, on one server
- CONTAINER: statistics from one container
- SERVER: statistics from one server
- COREGROUP: statistics from servers within a core group
- ZONE: statistics from servers within a zone

Grid Name The grid name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Server Name The server name. The valid format is an alphanumeric string, with a maximum of 48 characters.

Core Group Name The core group name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Statistics Name The name of the unit for which the information is aggregated in this row. For example, if the information is aggregated for a zone, this is the zone name; if the information is for an individual map, this is the map name, and so on. The valid format is an alphanumeric string, with a maximum of 48 characters.

Statistics ID The unique identifier for this statistics row. Used for identifying the statistic that triggers a situation. The valid format is an alphanumeric string, with a maximum of 96 characters.

Partition ID The partition ID for this Grid Shard, collected for SHARD aggregation level only. The valid format is a positive integer.

Summary Level The aggregation level and scope for this row. This field determines the values of the **Statistics Level** and **Statistics Scope** fields. The valid values are:

- SHARD: raw statistics for an individual shard. Level is SHARD, Scope is SHARD.
- PARTITION: aggregated information for all shards in a partition. Level is PARTITION, Scope is ZONE
- MAP: aggregated information for a map, on all servers within a zone. Level is MAPSET, Scope is ZONE.
- MAPSET: aggregated information for a map set, on all servers within a zone. Level is MAPSET, Scope is ZONE.
- GRID: aggregated information for a grid, on all servers within a zone. Level is GRID, Scope is ZONE.

- DOMAIN: aggregated information for a domain, on all servers within a zone. Level is DOMAIN, Scope is ZONE.
- MAP_SRV: aggregated information for a map, on a single server. Level is MAP, Scope is SERVER.
- MAPSET_SRV: aggregated information for a map set, on a single server. Level is MAPSET, Scope is SERVER.
- GRID_SRV: aggregated information for a grid, on a single server. Level is GRID, Scope is SERVER.
- DOMAIN_SRV: aggregated information for a domain, on a single server. Level is DOMAIN, Scope is SERVER.
- MAP_CG: aggregated information for a map, on all servers within a core group. Level is MAP, Scope is COREGROUP.
- MAPSET_CG: aggregated information for a map set, on all servers within a core group. Level is MAPSET, Scope is COREGROUP.
- GRID_CG: aggregated information for a grid, on all servers within a core group. Level is GRID, Scope is COREGROUP.
- DOMAIN_CG: aggregated information for a domain, on all servers within a core group. Level is DOMAIN, Scope is COREGROUP.

Map Name The map name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Important: Some name fields can be set to [Summary] at various aggregation levels. For example, **Server Name** is [Summary] if information is aggregated from multiple servers.

Map Count The number of the map entries held in this Grid Partition. The valid format is a positive integer.

Get Count The number of maps gets registered on the collection interval. The valid format is a positive integer. The Get Count attribute is only supported for WebSphere Extreme Scale 7.1. Ignore this attribute value for WebSphere Extreme Scale 6.1 and 7.0; it does not reflect the actual hit count and is only used for internal calculations.

Hit Count The number of map hits registered on the collection interval. The valid format is a positive integer. The Hit Count attribute is only supported for WebSphere Extreme Scale 7.1. Ignore this attribute value for WebSphere Extreme Scale 6.1 and 7.0; it does not reflect the actual get count and is only used for internal calculations.

Hit Rate The map hit rate percentage registered on the collection interval. -1 when **Get Count** is 0. The valid format is a positive integer.

Used KBytes The memory, in kilobytes, used by all entries in the map. The valid format is a positive integer. This information is not available for WebSphere Extreme Scale 6.1 and 7.0; for these versions, the value is always -1.

Batch Update Total Time The total time, in seconds, spent on batch database updates for the Map during the collection interval. The valid format is a positive integer.

Extreme Scale Shards attributes

The **Extreme Scale Shards** attributes provide statistics for a particular shard running within a grid container, or aggregated statistics for several shards. A shard runs on a particular server. A shard belongs to a grid and a map set. A shard corresponds to a partition. One partition can contain several shards on different servers, but only one shard can belong to a partition on any given server.

The attributes within this group are used to build the “Shards workspace” on page 560,“Shards – Selected Level workspace” on page 561 and “Shards – Selected Partition workspace” on page 564.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time at which the agent has collected data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 84. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time Indicates the length of the sample interval in seconds. The valid format is a positive integer.

Server Name The server name. The valid format is an alphanumeric string, with a maximum of 48 characters.

Grid Name The grid name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Core Group Name The core group name. The valid format is an alphanumeric string, with a maximum of 32 characters.

MapSet Name The map set name. The valid format is an alphanumeric string, with a maximum of 32 characters.

Domain Name The domain name for the grid instance within the server JVM. The valid format is an alphanumeric string, with a maximum of 32 characters. Only available for WebSphere Extreme Scale 7.1.

Container Name The name of this Grid Container. The valid format is an alphanumeric string, with a maximum of 32 characters.

Important: Some name fields might be set to `[Summary]` at various aggregation levels. For example, **Server Name** is `[Summary]` if information is aggregated from multiple servers.

Statistics Name The name of the unit for which the information is aggregated in this row. For example, if the information is aggregated for a zone, this is the zone name; if the information is for an individual shard, this is the shard name, and so on. The valid format is an alphanumeric string, with a maximum of 48 characters.

Display name The unique identifier for this statistics row. Used for identifying the statistic that triggers a situation. The valid format is an alphanumeric string, with a maximum of 96 characters.

State The state of the shard, as reported by WebSphere Extreme Scale. The valid format is an alphanumeric string, with a maximum of 32 characters.

Partition The partition ID for this Grid Shard, collected for SHARD aggregation level only. The valid format is a positive integer.

Active Request Count The number of requests currently being processed by this shard. The valid format is a positive integer.

Request Count The total number of requests that this shard has processed or forwarded on the collection interval. The valid format is a positive integer.

Forwarded Request Count The number of requests that this shard has forwarded on the collection interval. The valid format is a positive integer.

Processed Request Count The number of requests that this shard has processed or forwarded on the collection interval. The valid format is a positive integer.

Request Rate The overall request processing rate for this shard, including both processed and forwarded requests, during the collection interval. The valid format is a decimal (formatted to three decimal places).

Forwarded Request Rate The rate of processing forwarded requests for this shard during the collection interval. The valid format is a decimal (formatted to three decimal places).

Processed Request Rate The rate of processing non-forwarded (processed) requests for this shard during the collection interval. The valid format is a decimal (formatted to three decimal places).

Type The type of the shard. The valid values are:

- `PRIMARY`: This is a primary shard in the partition.
- `INACTIVE`: This shard is currently inactive within the partition.
- `SYNC_REPLICA`: This shard is currently a replica within the partition and is synchronized with the primary shard.
- `ASYNC_REPLICA`: This shard is currently a replica within the partition and is not synchronized with the primary shard.

Statistics Level The level at which statistics are aggregated in the table. The valid values are:

- `SHARD`: raw statistics for an individual shard
- `PARTITION`: summary across all shards in a partition. Not used in the current version.
- `MAP`: summary across all partitions within a map.
- `MAPSET`: summary across all partitions within a map set.

- GRID: summary across all map sets within a Grid.
- DOMAIN: summary across all Grid Containers within a Grid Domain,

Scope Level The servers from which the statistic information is collected for the table. The valid values are:

- SHARD: statistics from a single shard, on one server
- CONTAINER: statistics from one container
- SERVER: statistics from one server
- COREGROUP: statistics from servers within a core group
- ZONE: statistics from servers within a zone

Summary Level The aggregation level and scope for this row. This field determines the values of the **Statistics Level** and **Statistics Scope** fields. The valid values are:

- SHARD: raw statistics for an individual shard. Level is SHARD, Scope is SHARD.
- PARTITION: aggregated information for all shards in a partition. Level is PARTITION, Scope is ZONE
- MAP: aggregated information for a map, on all servers within a zone. Level is MAPSET, Scope is ZONE.
- MAPSET: aggregated information for a map set, on all servers within a zone. Level is MAPSET, Scope is ZONE.
- GRID: aggregated information for a grid, on all servers within a zone. Level is GRID, Scope is ZONE.
- DOMAIN: aggregated information for a domain, on all servers within a zone. Level is DOMAIN, Scope is ZONE.
- MAP_SRV: aggregated information for a map, on a single server. Level is MAP, Scope is SERVER.
- MAPSET_SRV: aggregated information for a map set, on a single server. Level is MAPSET, Scope is SERVER.
- GRID_SRV: aggregated information for a grid, on a single server. Level is GRID, Scope is SERVER.
- DOMAIN_SRV: aggregated information for a domain, on a single server. Level is DOMAIN, Scope is SERVER.
- MAP_CG: aggregated information for a map, on all servers within a core group. Level is MAP, Scope is COREGROUP.
- MAPSET_CG: aggregated information for a map set, on all servers within a core group. Level is MAPSET, Scope is COREGROUP.
- GRID_CG: aggregated information for a grid, on all servers within a core group. Level is GRID, Scope is COREGROUP.
- DOMAIN_CG: aggregated information for a domain, on all servers within a core group. Level is DOMAIN, Scope is COREGROUP.

Extreme Scale Placements attributes

The **Extreme Scale Placements** attributes provide information about grids and their placement in the WebSphere Extreme Scale zone.

The attributes within this group are used to build the “Grids workspace” on page 550.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 32 characters.

Sample Date and Time The date and time at which the agent has collected this data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 85. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time The length of the sample interval in seconds. The valid format is a positive integer.

Grid Name The name of the grid. The valid format is an alphanumeric string, with a maximum of 32 characters.

MapSet Name The name of the map set. The valid format is an alphanumeric string, with a maximum of 32 characters.

Placement Strategy The Grid shards placement strategy. The valid values are `PER_CONTAINER` and `FIXED_PARTITIONS`.

Min Sync Replicas The minimum amount of synchronous replicas in the grid. The valid format is a positive integer.

Development Mode The Grid development mode status. The valid format is a positive integer.

Number of Containers The number of containers in the grid. The valid format is a positive integer.

Number of Machines The number of machines in the grid. The valid format is a positive integer.

Outstanding Work Items The number of Outstanding Work Items in the grid. The valid format is a positive integer.

Number of Active Zones The number of Active Zones in the grid. The valid format is a positive integer.

Extreme Scale Thread Pools attributes

The **Extreme Scale Thread Pools** attributes provide information about thread pools on servers within the WebSphere Extreme Scale zone.

The attributes within this group are used to build the “Containers workspace” on page 548 and “Selected Containers workspace” on page 549.

All attributes in this group are only available for WebSphere Extreme Scale 7.1.

Origin Node The server name subnode. The valid format is an alphanumeric string, with a maximum of 32 characters.

Sample Date and Time The date and time at which the agent has collected this data. The valid format is a 16-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 86. Format of the 16-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Interval Time The length of the sample interval in seconds. The valid format is a positive integer.

Thread Pool Name The name of the thread pool. The valid format is an alphanumeric string, with a maximum of 32 characters.

Server Name The name of the server. The valid format is an alphanumeric string, with a maximum of 48 characters.

Core Group The name of core group. The valid format is an alphanumeric string, with a maximum of 32 characters.

Maximum Size The maximum pool size. The valid format is a positive integer. Only available for WebSphere Extreme Scale 7.1.

Minimum Size The minimum pool size. The valid format is a positive integer. Only available for WebSphere Extreme Scale 7.1.

Active Thread Count The active thread count. The valid format is a positive integer. Only available for WebSphere Extreme Scale 7.1.

Stats Level The level at which statistics are aggregated in the table. The valid values are:

- SERVER: information for thread pools on a single server.
- COREGROUP: aggregated information for thread pools on servers within a core group.
- ZONE: aggregated information for thread pools on servers within a zone.

WebSphere XS Configuration attributes

The WebSphere XS Configuration attributes provide data access for WXS cell configuration workspaces.

Origin Node The XD Cell subnode. The valid format is an alphanumeric string, with a maximum of 32 characters.

Sample Date and Time The date and time at which the agent collected data. The valid format is T.

Interval Time The length of the sample interval in seconds. The valid format is a 4-byte integer. Valid fixed values are: =-1

Row Id The row identifier. The valid format is a 2-byte integer.

View Id The view identifier. The valid format is a 2-byte integer.

Act Id The act identifier. The valid format is a 2-byte integer.

Data The data string. The valid format is an alphanumeric string, with a maximum of 4096 characters.

Situations for WebSphere Extreme Scale

Some predefined situations are available for WebSphere Extreme Scale monitoring. You can use them to monitor quorum status and detect the stopping of a catalog server.

These predefined situations display an alert status. When these situations trigger an alert, you can investigate the event by opening the corresponding workspace.

How the situations work

Situations are tests expressed in IF-TRUE format of system conditions that you want to monitor; the tested value is an ITCAM Agent for WebSphere Applications attribute expressed in the form *attribute-group.attribute-name*. If the specified condition occurs or exists, the situation is true, and an alert is issued.

Accessing the situations

A number of predefined situations are shipped to monitor WebSphere Extreme Scale. Access the situations in the Tivoli Enterprise Portal using one of the following methods:

- From the toolbar on the main menu, click the Situation Editor icon  and scroll to the situation that you want to view.
- Right-click a WebSphere Extreme Scale zone node. Click **Manage Situations** to display all the situations available for the zone. If you want to start, stop edit or model a situation right-click the situation and select the option you want.

WXSQuorumWaiting is a critical situation. It is raised when a Grid within the monitored zone or zones is waiting for the quorum.

WXSCatalogOffline is a critical situation. It is raised when a catalog server within the monitored zone or zones goes offline.

WXSContainerOffline is a critical situation. It is raised when a container server within the monitored zone or zones goes offline.

WXSHighUsedMemoryPercent is a critical situation. It is raised when the used heap memory goes over 80% of the maximum on a container server within the monitored zone or zones.

WXSHungThreadsDetected is a critical situation. It is raised when the used hung threads are detected on a container server within the monitored zone or zones.

WebSphere Extreme Scale Take Action Commands

Use Take Action commands to manage Extreme Scale zone nodes, to trigger zone topology discovery, to override the quorum on the Extreme Scale catalog service, to shut down servers in the Extreme Scale grid, and to trigger map set placement in the grid.

WXS_Add_Zone: add a zone node

Use the WXS_Add_Zone Take Action command, available on the Agent for WebSphere Applications node level, to add a new WebSphere Extreme Scale zone node for this server.

Command syntax

YN:WXS_Add_Zone *Node_Name*

Parameters:

Node_Name
The name of the new zone node

Important: A newly created node contains no data. To monitor a WebSphere Extreme Scale zone or several zones, configure the node. To do this, right-click the node and select **WebSphere XS Zone – Configuration**. See "WebSphere Extreme Scale configuration workspace" on page 544.

WXS_Rename_Zone: rename a zone node

Use the WXS_Rename_Zone Take Action command, available on the WebSphere Extreme Scale zone node level, to change the name of the zone node in the navigator.

Attention: When you rename a zone node, Tivoli Monitoring loses all history information for it.

Command syntax

YN:WXS_Rename_Zone *New_Node_Name*

Parameters:

New_Node_Name
The new name of the zone node.

Important: The node with the old name is disabled. Use the **Clear Offline Nodes** command in Tivoli Enterprise Portal to remove the inactive zone from the tree.

WXS_Remove_Zone: remove a zone node

Use the WXS_Remove_Zone Take Action command, available on the WebSphere Extreme Scale zone node level, to remove a zone node.

Command syntax

YN:WXS_Remove_Zone

Important: The zone is disabled. Use the **Clear Offline Nodes** command in Tivoli Enterprise Portal to remove the inactive zone from the tree.

WXS_Remove_Offline: remove a catalog or container in the offline state from monitoring

Use the WXS_Remove_Offline Take Action command, available on the WebSphere Extreme Scale zone node level, to stop monitoring a catalog or container (server) that is now offline. If you do not use this Take Action command, the offline catalog or container remains in the monitoring tables for a while. It is listed in workspaces and might trigger false events.

Command syntax

YN:WXS_Remove_Offline *Server_Name*

Parameters:

Server_Name
The name of the server to remove.

WXS_Discover_Zone: discover the zone topology

Use the WXS_Discover_Zone Take Action command, available on the WebSphere Extreme Scale zone node level, to discover the topology of the monitored zone. This action updates monitoring of the zone to the new topology.

Zone topology is discovered automatically once in the Discovery interval (set in the "WebSphere Extreme Scale configuration workspace" on page 544). If container servers or catalog services were added, removed, or restarted, you can use this Take Action command to update monitoring to the new topology without waiting for expiration of the Discovery interval.

Command syntax

YN:WXS_Discover_Zone

WXS_Override_Quorum: override quorum for the catalog server

Use the WXS_Override_Quorum Take Action command, available on the WebSphere Extreme Scale zone node level, to override quorum on the monitored zone.

To perform this action, the agent uses the OverrideQuorum method on the primary catalog server.

Command syntax

YN:WXS_Override_Quorum

WXS_Tear_Down_Servers: stop Extreme Scale container server

Use the WXS_Tear_Down_Servers Take Action command to stop an Extreme Scale container server. This command is available on the WebSphere Extreme Scale zone node, or in the Container Servers Summary table (Containers workspace).

To perform this action, the agent uses the tearDownServers method on PlacementServiceMBean.

When the action is invoked from the Container Servers Summary table ("Containers workspace" on page 548), the server name is taken from the selected row.

Command syntax

YN:WXS_Tear_Down_Servers *Server_Name*

Parameters:

Server_Name
The name of the server to stop.

WXS_Trigger_Placement: trigger the placement of a map set on an Extreme Scale grid

Use the WXS_Trigger_Placement Take Action command to trigger the placement of a map set on an Extreme Scale grid. By default, a grid be placed automatically, but you can use this command to start its placement earlier. This command is available on the WebSphere Extreme Scale zone node, or in the Grids Summary table (Grids workspace).

To perform this action, the agent uses the triggerPlacement method on PlacementServiceMBean.

When the action is invoked from the Grids Summary table ("Grids workspace" on page 550), the grid and map set names are taken from the selected row.

Command syntax

YN:WXS_Trigger_Placement *Grid_Name MapSet_Name*

Parameters:

Grid_Name
The name of the grid.

MapSet_Name
The name of the map set. The agent triggers placement of this map set on the grid.

WebSphere PMI Attribute Mapping

The following tables show how the data displayed in the ITCAM for Application Diagnostics resource workspaces map to their corresponding WebSphere PMI categories and their attributes. It also provides the monitoring overhead incurred when turning on these attributes.

By default, ITCAM changes the PMI collection level based on its monitoring level. See the *Advanced configuration of the Agent* chapter of *ITCAM Agent for WebSphere Applications Installation and Configuration Guide* to learn more about this behavior and how to turn it off for custom monitoring.

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping.

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

AppServer--High Availability Manager	ITM Table Name: KYNHAMGMT			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: HAManager			
Local Groups	LocalGroupCount	LocalGroupCount	All	High
Group State Rebuild Time	GroupStateRebuildTime	GroupStateRebuildTime	All	High
Bulletin-Board Subjects	BulletinBoardSubjectCount	BulletinBoardSubjectCount	All	High
Bulletin-Board Subscriptions	BulletinBoardSubcriptionCount	BulletinBoardSubcriptionCount	All	High
Bulletin-Board Rebuild Time	BulletinBoardRebuildTime	BulletinBoardRebuildTime	All	High
Local Bulletin-Board Subjects	LocalBulletinBoardSubjectCount	LocalBulletinBoardSubjectCount	All	High
AppServer-DCS Stacks	**ITM Table Name: KYNDCSSTK**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: DCS Statistics			
Message Buffer Reallocations	MessageBufferReallocationCount	MessageBufferReallocationCount	All	Medium
Sent Messages	SentMessageCount	SentMessageCount	All	High
Average Outgoing Message Size	OutgoingMessageSize	OutgoingMessageSize	All	High
Minimum Outgoing Message Size	OutgoingMessageSize	OutgoingMessageSize	All	High
Maximum Outgoing Message Size	OutgoingMessageSize	OutgoingMessageSize	All	High
Outgoing Messages	SentMessageCount	SentMessageCount	All	High
Average Incoming Message Size	IncomingMessageSize	IncomingMessageSize	All	High
Minimum Incoming Message Size	IncomingMessageSize	IncomingMessageSize	All	High
Maximum Incoming Message Size	IncomingMessageSize	IncomingMessageSize	All	High
Incoming Messages	ReceivedMessageCount	ReceivedMessageCount	All	High
Synchronization Completion Time	SynchronizationCompleteTime	SynchronizationCompleteTime	All	High
Synchronization Timeouts	SynchronizationTimeoutCount	SynchronizationTimeoutCount	All	Medium
High Severity Congestion Events	HighSeverityCongestionEventCount	HighSeverityCongestionEventCount	All	Medium
Coalesce Time	CoalesceTime	CoalesceTime	All	Medium
Join View Change Time	JoinViewChangeTime	JoinViewChangeTime	All	High
Remove View Change Time	RemoveViewChangeTime	RemoveViewChangeTime	All	High

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Suspicions	SuspicionCount	SuspicionCount	All	High
View Changes	ViewChangeCount	ViewChangeCount	All	Medium
Group Size	ViewGroupSize	ViewGroupSize	All	Medium
Web Applications	**ITM Table Name: KYNAPP**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Web Applications			
Request Count	RequestCount (Servlet Info)	RequestCount (Servlet Info)	Basic	Low
Request Rate (per sec)	RequestCount	RequestCount	Basic	Low
Error Count	ErrorCount (Servlet Info)	ErrorCount (Servlet Info)	Extended	Low
Error Rate (per sec)	ErrorCount (Servlet Info)	ErrorCount (Servlet Info)	Extended	Low
Average Response Time (ms)	ServiceTime (Servlet Info)	ServiceTime (Servlet Info)	Basic	Medium
Average Concurrent Requests	ConcurrentRequests (Servlet Info)	ConcurrentRequests (Servlet Info)	Extended	High
Servlets Loaded	LoadedServletCount	LoadedServletCount	All	Low
Servlets Reloaded	ReloadCount	ReloadCount	All	Low
WebApplications --ServletSessions	**ITM Table Name: KYNSERVS**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Servlet Session Manager			
Sessions Created	CreateCount	CreateCount	All	Low
Session Creation Rate (per sec)	CreateCount	CreateCount	All	Low
Sessions Invalidated	InvalidateCount	InvalidateCount	All	Low
Session Invalidation Rate (per sec)	InvalidateCount	InvalidateCount	All	Low
Average Session Lifetime (ms)	LifeTime	LifeTime	Extended	Medium
Average Concurrently Active Sessions	ActiveCount	ActiveCount	All	High
Average Concurrently Live Sessions	LiveCount	LiveCount	Basic	High
Failed Session Requests	NoRoomForNewSessionCount	NoRoomForNewSessionCount	Extended	Low
Failed Session Request Rate (per sec)	NoRoomForNewSessionCount	NoRoomForNewSessionCount	Extended	Low
Cache Discards	CacheDiscardCount	CacheDiscardCount	All	Low
Cache Discard Rate (per sec)	CacheDiscardCount	CacheDiscardCount	All	Low
External Read Time (ms)	ExternalReadTime	ExternalReadTime	Extended	Medium
External Read Size (bytes)	ExternalReadSize	ExternalReadSize	Extended	Medium

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

External Write Time (ms)	ExternalWriteTime	ExternalWriteTime	Extended	Medium
External Write Size (bytes)	ExternalWriteSize	ExternalWriteSize	Extended	Medium
Broken Session Affinities	AffinityBreakCount	AffinityBreakCount	All	Low
Broken Session Affinity Rate (per sec)	AffinityBreakCount	AffinityBreakCount	All	Low
Time since Last Activated	TimeSinceLastActivated	TimeSinceLastActivated	All	Medium
Nonexistent Session Requests	ActivateNonExistSessionCount	ActivateNonExistSessionCount	All	Low
Nonexistent Session Request Rate (per sec)	ActivateNonExistSessionCount	ActivateNonExistSessionCount	All	Low
Total Serializable Session Object Size (bytes)	SessionObjectSize	SessionObjectSize	All	Max
Average Serializable Session Object Size (bytes)	SessionObjectSize	SessionObjectSize	All	Max
Min Serializable Session Object Size (bytes)	SessionObjectSize	SessionObjectSize	All	Max
Max Serializable Session Object Size (bytes)	SessionObjectSize	SessionObjectSize	All	Max
EJB Containers	**ITM Table Name: KYNCONTNR**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Enterprise Beans			
Method Average Response Time (ms)	MethodResponseTime	MethodResponseTime	Basic	High
Method Invocation Count	MethodCallCount	MethodCallCount	Basic	High
Method Invocation Rate (per sec)	MethodCallCount	MethodCallCount	Basic	High
Create Average Time (ms)	CreateTime	CreateTime	All	Max
Remove Average Time (ms)	RemoveTime	RemoveTime	All	Max
Average Concurrently Ready Beans	ReadyCount	ReadyCount	Basic	Low
Average Concurrently Live Beans	LiveCount	LiveCount	Extended	High
Active Method Count	ActiveMethodCount	ActiveMethodCount	All	High
Create Count	CreateCount	CreateCount	Basic	Low
Creation Rate (per sec)	CreateCount	CreateCount	Basic	Low
Remove Count	RemoveCount	RemoveCount	Basic	Low
Removal Rate per sec)	RemoveCount	RemoveCount	Basic	Low
Activate Count	ActivateCount	ActivateCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Activation Rate (per sec)	ActivateCount	ActivateCount	All	Low
Passivate Count	PassivateCount	PassivateCount	Basic	Low
Passivation Rate (per sec)	PassivateCount	PassivateCount	Basic	Low
Entity Bean Load Count	LoadCount	LoadCount	All	Low
Entity Bean Load Rate (per sec)	LoadCount	LoadCount	All	Low
Entity Bean Store Count	StoreCount	StoreCount	All	Low
Entity Bean Store Rate (per sec)	StoreCount	StoreCount	All	Low
Instantiate Count	InstantiateCount	InstantiateCount	All	Low
Instantiation Rate (per sec)	InstantiateCount	InstantiateCount	All	Low
Destroy Count	FreedCount	FreedCount	All	Low
Destruction Rate (per sec)	FreedCount	FreedCount	All	Low
EJB Containers ---Enterprise Java Beans	**ITM Table Name: KYNEJB**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Enterprise Beans			
Method Invocations	MethodCallCount	MethodCallCount	Basic	High
Method Invocation Rate (per sec)	MethodCallCount	MethodCallCount	Basic	High
Method Average Response Time (ms)	MethodResponseTime	MethodResponseTime	Basic	High
Create Count	CreateCount	CreateCount	Basic	Low
Creation Rate (per sec)	CreateCount	CreateCount	Basic	Low
Create Average Time (ms)	CreateTime	CreateTime	All	Max
Remove Count	RemoveCount	RemoveCount	Basic	Low
Removal Rate (per sec)	RemoveCount	RemoveCount	Basic	Low
Remove Average Time (ms)	RemoveTime	RemoveTime	All	Max
Activate Count	ActivateCount	ActivateCount	All	Low
Activation Rate (per sec)	ActivateCount	ActivateCount	All	Low
Passivate Count	PassivateCount	PassivateCount	All	Low
Passivation Rate	PassivateCount	PassivateCount	All	Low
Entity Bean Load Count	LoadCount	LoadCount	All	Low
Entity Bean Load Rate (per sec)	LoadCount	LoadCount	All	Low
Entity Bean Store Count	StoreCount	StoreCount	All	Low
Entity Bean Store Rate (per sec)	StoreCount	StoreCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Instantiate Count	InstantiateCount	InstantiateCount	All	Low
Destroy Count	FreedCount	FreedCount	All	Low
Destruction Rate (per sec)	FreedCount	FreedCount	All	Low
Find Count	RetrieveFromPoolSuccessCount	RetrieveFromPoolSuccessCount	All	Low
Find Rate (per sec)	RetrieveFromPoolSuccessCount	RetrieveFromPoolSuccessCount	All	Low
Get Count	RetrieveFromPoolCount	RetrieveFromPoolCount	All	Low
Get Rate (per sec)	RetrieveFromPoolCount	RetrieveFromPoolCount	All	Low
Return Count	ReturnsToPoolCount	ReturnsToPoolCount	Extended	Low
Return Rate (per sec)	ReturnsToPoolCount	ReturnsToPoolCount	Extended	Low
Discard Count	ReturnsDiscardCount	ReturnsDiscardCount	Extended	Low
Discard Rate (per sec)	ReturnsDiscardCount	ReturnsDiscardCount	Extended	Low
Drain Count	DrainsFromPoolCount	DrainsFromPoolCount	All	Low
Drain Rate (per sec)	DrainsFromPoolCount	DrainsFromPoolCount	All	Low
Average Concurrently Ready Beans	ReadyCount	ReadyCount	Basic	High
Average Concurrently Live Beans	LiveCount	LiveCount	Extended	High
Active Method Count	ActiveMethodCount	ActiveMethodCount	All	High
Average Objects Discarded	DrainSize	DrainSize	All	Medium
Average Objects in Pool	PooledCount	PooledCount	Basic	High
EJB Containers ---Container Transactions	**ITM Table Name: KYNTRANS**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Transaction Manager			
Global Transactions Begun	GlobalBegunCount	GlobalBegunCount	Extended	Low
Global Transactions Begin Rate (per sec)	GlobalBegunCount	GlobalBegunCount	Extended	Low
Local Transactions Begun	LocalBegunCount	LocalBegunCount	Extended	Low
Local Transactions Begin Rate (per sec)	LocalBegunCount	LocalBegunCount	Extended	Low
Global Transactions Active	ActiveCount	ActiveCount	Basic	Low
Local Transactions Active	LocalActiveCount	LocalActiveCount	All	Low
Global Transactions Committed	CommittedCount	CommittedCount	Basic	Low
Global Transaction Commit Rate (per sec)	CommittedCount	CommittedCount	Basic	Low
Local Transactions Committed	LocalCommittedCount	LocalCommittedCount	All	Low
Local Transaction Commit Rate (per sec)	LocalCommittedCount	LocalCommittedCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Global Transactions Rolled Back	RolledbackCount	RolledbackCount	Basic	Low
Global Transaction Rollback Rate (per sec)	RolledbackCount	RolledbackCount	Basic	Low
Local Transactions Rolled Back	LocalRolledbackCount	LocalRolledbackCount	All	Low
Local Transaction Rollback Rate (per sec)	LocalRolledbackCount	LocalRolledbackCount	All	Low
Global Transaction Timeouts	GlobalTimeoutCount	GlobalTimeoutCount	Extended	Low
Global Transaction Timeout Rate (per sec)	GlobalTimeoutCount	GlobalTimeoutCount	Extended	Low
Local Transaction Timeouts	LocalTimeoutCount	LocalTimeoutCount	Extended	Low
Local Transaction Timeout Rate (per sec)	LocalTimeoutCount	LocalTimeoutCount	Extended	Low
Global Transactions Optimized	OptimizationCount	OptimizationCount	All	Low
Global Transaction Optimize Rate (per sec)	CommittedCount	CommittedCount	Basic	Low
Global Transactions Involved	GlobalInvolvedCount	GlobalInvolvedCount	All	Low
Global Transactions Involve Rate (per sec)	GlobalInvolvedCount	GlobalInvolvedCount	All	Low
Global Transaction Duration (ms)	GlobalTranTime	GlobalTranTime	Extended	Medium
Local Transaction Duration (ms)	LocalTranTime	LocalTranTime	Extended	Medium
Global Transaction before Completion Duration (ms)	GlobalBeforeCompletionTime	GlobalBeforeCompletionTime	All	Medium
Local Transaction before Completion Duration (ms)	LocalBeforeCompletionTime	LocalBeforeCompletionTime	All	Medium
Global Transaction Commit Duration (ms)	GlobalCommitTime	GlobalCommitTime	All	Medium
Local Transaction Commit Duration (ms)	LocalCommitTime	LocalCommitTime	All	Medium
Global Transaction Prepare Duration (ms)	GlobalPrepareTime	GlobalPrepareTime	All	Medium
EJB Containers ---Container Object Pools	**ITM Table Name: KYNCNTROP**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
Average Objects in Pool	PooledCount	PooledCount	Basic	High
Average Objects Discarded	DrainSize	DrainSize	All	Medium
Find Count	RetrieveFromPoolSuccessCount	RetrieveFromPoolSuccessCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Find Rate	RetrieveFromPoolSuccessCount	RetrieveFromPoolSuccessCount	All	Low
Get Count	RetrieveFromPoolCount	RetrieveFromPoolCount	All	Low
Get Rate	RetrieveFromPoolCount	RetrieveFromPoolCount	All	Low
Return Count	ReturnsToPoolCount	ReturnsToPoolCount	Extended	Low
Return Rate	ReturnsToPoolCount	ReturnsToPoolCount	Extended	Low
Discard Count	ReturnsDiscardCount	ReturnsDiscardCount	Extended	Low
Discard Rate	ReturnsDiscardCount	ReturnsDiscardCount	Extended	Low
Drain Count	DrainsFromPoolCount	DrainsFromPoolCount	All	Low
Drain Rate	DrainsFromPoolCount	DrainsFromPoolCount	All	Low
DB Connection Pools	**ITM Table Name: KYNDBCONP**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: JDBC Connection Pools			
Maximum Pool Size	PoolSize	PoolSize	Basic	High
Average Pool Size	PoolSize	PoolSize	Basic	High
Average Waiting Threads	WaitingThreadCount	WaitingThreadCount	Basic	High
Average Wait Time (ms)	WaitTime	WaitTime	Basic	Medium
Average Usage Time (ms)	UseTime	UseTime	Basic	Medium
Average Free Pool Size	FreePoolSize	FreePoolSize	Basic	High
JDBC Time(ms)	JDBCTime	JDBCTime	Extended	Medium
Percent Used	PercentUsed	PercentUsed	Basic	High
Percent of Time Pool at Max	PercentMaxed	PercentMaxed	All	High
Connections Created	CreateCount	CreateCount	Basic	Low
Connection Creation Rate (per sec)	CreateCount	CreateCount	Basic	Low
Connections Allocated	AllocateCount	AllocateCount	Extended	Low
Connection Allocation Rate (per sec)	AllocateCount	AllocateCount	Extended	Low
Connections Destroyed	CloseCount	CloseCount	Basic	Low
Connection Destruction Rate (per sec)	CloseCount	CloseCount	Basic	Low
Threads Timed Out	FaultCount	FaultCount	Extended	Low
Thread Timeout Rate (per sec)	FaultCount	FaultCount	Extended	Low
Prep Statement Cache Discards	PrepStmtCacheDiscardCount	PrepStmtCacheDiscardCount	Extended	Low
Prep Statement Cache Discard Rate (per sec)	PrepStmtCacheDiscardCount	PrepStmtCacheDiscardCount	Extended	Low
Return Count	ReturnCount	ReturnCount	Extended	Low
Return Rate(per sec)	ReturnCount	ReturnCount	Extended	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

J2C Connection Pools	**ITM Table Name: KYNJ2C**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: J2C Connection Pools			
Maximum Pool Size	PoolSize	PoolSize	Basic	High
Average Pool Size	PoolSize	PoolSize	Basic	High
Average Free Connections	FreePoolSize	FreePoolSize	Basic	High
Connections Used	ManagedConnectionCount	ManagedConnectionCount	All	Low
Connection Handles	ConnectionHandleCount	ConnectionHandleCount	All	Low
Average Wait Time (ms)	WaitTime	WaitTime	Basic	Medium
Concurrent Waiting Threads	WaitingThreadCount	WaitingThreadCount	Basic	High
Average Usage Time (ms)	UseTime	UseTime	Basic	Medium
Pool Used (%)	PercentUsed	PercentUsed	All	High
Percent of Time Pool at Max	PercentMaxed	PercentMaxed	All	High
Connections Created	CreateCount	CreateCount	Basic	Low
Connection Creation Rate (per sec)	CreateCount	CreateCount	Basic	Low
Connections Allocated	AllocateCount	AllocateCount	All	Low
Connection Allocation Rate (per sec)	AllocateCount	AllocateCount	All	Low
Connections Returned	FreedCount	FreedCount	All	Low
Connection Return Rate (per sec)	FreedCount	FreedCount	All	Low
Connections Destroyed	CloseCount	CloseCount	Basic	Low
Connection Destruction Rate (per sec)	CloseCount	CloseCount	Basic	Low
Connection Pool Timeouts	FaultCount	FaultCount	All	Low
Connection Pool Timeout Rate (per sec)	FaultCount	FaultCount	All	Low
Thread Pools	**Table Name: KYNTHRDP**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Thread Pools			
Maximum Pool Size	PoolSize	PoolSize	Basic	High
Average Pool Size	PoolSize	PoolSize	Basic	High
Average Active Threads	ActiveCount	ActiveCount	Extended	High
Average Free Threads	PoolSize - ActiveCount	PoolSize - ActiveCount	Extended	High
Percent of Time Pool at Max	PercentMaxed	PercentMaxed	All	High
Threads Created	CreateCount	CreateCount	All	Low
Thread Creation Rate (per sec)	CreateCount	CreateCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Threads Destroyed	DestroyCount	DestroyCount	All	Low
Thread Destruction Rate (per sec)	DestroyCount	DestroyCount	All	Low
Thread Pools ---Alarm Manager	**ITM Table Name: KYNALARMM**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Alarm Manager			
Alarms Cancelled	AlarmsCancelledCount	AlarmsCancelledCount	All	High
Alarms Latency Duration	AlarmLatencyDuration	AlarmLatencyDuration	All	High
Alarms Rate	AlarmRate	AlarmRate	All	High
Alarms Created	AlarmsCreatedCount	AlarmsCreatedCount	All	High
Alarms Fired	AlarmsFiredCount	AlarmsFiredCount	All	High
Alarms Pending Size	AlarmsPendingSize	AlarmsPendingSize	All	High
Dynamic Cache	**ITM Table Name: KYNCACHE**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Dynamic Caching			
Maximum In-Memory Cache Size	MaxInMemoryCacheEntryCount	MaxInMemoryCacheEntryCount	All	Low
Current In-Memory Cache Size	InMemoryCacheEntryCount	InMemoryCacheEntryCount	All	Low
In-Memory and Disk Timeouts	TimeoutInvalidationCount	TimeoutInvalidationCount	All	Low
In-Memory and Disk Timeout Rate (per sec)	TimeoutInvalidationCount	TimeoutInvalidationCount	All	Low
Dynamic Cache Templates	**ITM Table Name: KYNCACHT**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Templates			
Current Cache Size	InMemoryAndDiskCacheEntry Count	InMemoryAndDiskCacheEntry Count	All	Low
Disk Hits	HitsOnDiskCount	HitsOnDiskCount	All	Low
Disk Hit Rate(per sec)	HitsOnDiskCount	HitsOnDiskCount	All	Low
Memory Hits	HitsInMemoryCount	HitsInMemoryCount	Extended	Low
Memory Hit Rate(per sec)	HitsInMemoryCount	HitsInMemoryCount	Extended	Low
Remote Hits	RemoteHitCount	RemoteHitCount	All	Low
Remote Hit Rate(per sec)	RemoteHitCount	RemoteHitCount	All	Low
Cache Misses	MissCount	MissCount	Extended	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Cache Miss Rate(per sec)	MissCount	MissCount	Extended	Low
Remote Cache Entries Received	RemoteCreationCount	RemoteCreationCount	All	Low
Remote Cache Entry Receive Rate(per sec)	RemoteCreationCount	RemoteCreationCount	All	Low
Client Requests	ClientRequestCount	ClientRequestCount	All	Low
Client Request Rate(per sec)	ClientRequestCount	ClientRequestCount	All	Low
Cluster Requests	DistributedRequestCount	DistributedRequestCount	All	Low
Cluster Request Rate(per sec)	DistributedRequestCount	DistributedRequestCount	All	Low
Total Explicit Invalidations	ExplicitInvalidationCount	ExplicitInvalidationCount	All	Low
Total Explicit Invalidation Rate(per sec)	ExplicitInvalidationCount	ExplicitInvalidationCount	All	Low
Timeout Invalidations	TimeoutInvalidationCount	TimeoutInvalidationCount	All	Low
Timeout Invalidation Rate(per sec)	TimeoutInvalidationCount	TimeoutInvalidationCount	All	Low
Least Recently Used Invalidations	LruInvalidationCount	LruInvalidationCount	All	Low
Least Recently Used Invalidation Rate(per sec)	LruInvalidationCount	LruInvalidationCount	All	Low
Explicit Memory Invalidations	ExplicitMemoryInvalidationCount	ExplicitMemoryInvalidationCount	All	Low
Explicit Memory Invalidation Rate(per sec)	ExplicitMemoryInvalidationCount	ExplicitMemoryInvalidationCount	All	Low
Explicit Disk Invalidations	ExplicitDiskInvalidationCount	ExplicitDiskInvalidationCount	All	Low
Explicit Disk Invalidations Rate(per sec)	ExplicitDiskInvalidationCount	ExplicitDiskInvalidationCount	All	Low
Explicit Local Invalidations	LocalExplicitInvalidationCount	LocalExplicitInvalidationCount	All	Low
Explicit Local Invalidation Rate(per sec)	LocalExplicitInvalidationCount	LocalExplicitInvalidationCount	All	Low
Explicit Remote Invalidations	RemoteExplicitInvalidationCount	RemoteExplicitInvalidationCount	All	Low
Explicit Remote Invalidation Rate(per sec)	RemoteExplicitInvalidationCount	RemoteExplicitInvalidationCount	All	Low
Workload Management Client	**ITM Table Name: KYNWLMCL**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Workload Management client			

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Outgoing Requests	OutgoingIIOPRequestCount	OutgoingIIOPRequestCount	All	Low
Outgoing Request Rate (per sec)	OutgoingIIOPRequestCount	OutgoingIIOPRequestCount	All	Low
Client Cluster Updates	ClientClusterUpdateCount	ClientClusterUpdateCount	All	Low
Client Cluster Update Rate (per sec)	ClientClusterUpdateCount	ClientClusterUpdateCount	All	Low
Client Response Time	ClientResponseTime	ClientResponseTime	All	Medium
Workload Management Server	**ITM Table Name: KYNWLMSR**			
		WebSphere PMI Level		
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Workload Management server			
Incoming Requests	IIOPRequestCount	IIOPRequestCount	Extended	Low
Incoming Request Rate (per sec)	IIOPRequestCount	IIOPRequestCount	Extended	Low
Incoming Strong Affinity Requests	StrongAffinityIIOPRequestCount	StrongAffinityIIOPRequestCount	All	Low
Incoming Strong Affinity Request Rate (per sec)	StrongAffinityIIOPRequestCount	StrongAffinityIIOPRequestCount	All	Low
Incoming Nonaffinity Requests	NoAffinityIIOPRequestCount	NoAffinityIIOPRequestCount	All	Low
Incoming Nonaffinity Request Rate (per sec)	NoAffinityIIOPRequestCount	NoAffinityIIOPRequestCount	All	Low
Incoming Non-WLM Object Requests	NonWLMEnabledIIOPRequestCount	NonWLMEnabledIIOPRequestCount	All	Low
Incoming Non-WLM Object Request Rate (per sec)	NonWLMEnabledIIOPRequest Count	NonWLMEnabledIIOPRequestCount	All	Low
Server Cluster Updates	ServerClusterUpdateCount	ServerClusterUpdateCount	All	Low
Server Cluster Update Rate (per sec)	ServerClusterUpdateCount	ServerClusterUpdateCount	All	Low
WLM Clients Serviced	WLMClientsServicedCount	WLMClientsServicedCount	All	Low
WLM Clients Serviced Rate (per sec)	WLMClientsServicedCount	WLMClientsServicedCount	All	Low
Concurrent Requests	ConcurrentRequestCount	ConcurrentRequestCount	Extended	High
Server Response Time (ms)	ServerResponseTime	ServerResponseTime	Extended	Medium
Scheduler	**ITM Table Name: KYNSCHED**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Schedulers			
Poll Count	PollCount	PollCount	All	High
Poll Duration	PollDuration	PollDuration	All	High

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Poll Query Duration	PollQueryDuration	PollQueryDuration	All	High
Run Duration	RunDuration	RunDuration	All	High
Task Collision Rate	TaskCollisionRate	TaskCollisionRate	All	High
Task Delay Duration	TaskDelayDuration	TaskDelayDuration	All	High
Task Expiration Rate	TaskExpirationRate	TaskExpirationRate	All	High
Task Failure Count	TaskFailureCount	TaskFailureCount	All	High
Task Finish Count	TaskFinishCount	TaskFinishCount	All	High
Task Finish Rate	TaskFinishRate	TaskFinishRate	All	High
Task Run Rate	TaskRunRate	TaskRunRate	All	High
Web Services	**ITM Table Name: KYNWEBSVC**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Web services			
Received Requests	ReceivedRequestCount	ReceivedRequestCount	All	Low
Dispatched Requests	DispatchedRequestCount	DispatchedRequestCount	All	Low
Processed Requests	ProcessedRequestCount	ProcessedRequestCount	All	Low
Response Time	ResponseTime	ResponseTime	All	High
Request Response Time	RequestResponseTime	RequestResponseTime	All	Medium
Dispatch Response Time	DispatchResponseTime	DispatchResponseTime	All	Medium
Reply Response Time	ReplyResponseTime	ReplyResponseTime	All	Medium
Payload Size	PayloadSize	PayloadSize	All	Medium
Reply Payload Size	ReplyPayloadSize	ReplyPayloadSize	All	Medium
Request Payload Size	RequestPayloadSize	RequestPayloadSize	All	Medium
WebServices Gateway	**ITM Table Name: KYNWEBSGW**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Web services Gateway			
Synchronous Requests	SynchronousRequestCount	SynchronousRequestCount	All	Low
Synchronous Responses	SynchronousResponseCount	SynchronousResponseCount	All	Low
Asynchronous Requests	AsynchronousRequestCount	AsynchronousRequestCount	All	Low
Asynchronous Responses	AsynchronousResponseCount	AsynchronousResponseCount	All	Low
Messaging Engines	**ITM Table Name: KYNMSGENG**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: SIB Service > SIB Messaging Engines			
Average Local Wait Time (ms)	LocalMessageWaitTime	LocalMessageWaitTime	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Expired Messages	ReportEnabledMessagesExpiredCount	ReportEnabledMessagesExpiredCount	All	Low
Incomplete Topic Publications	IncompletePublicationCount	IncompletePublicationCount	All	Low
Total Published	TotalMessagesPublishedCount	TotalMessagesPublishedCount	All	Low
Client Communications	**ITM Table Name: KYNCLICOM**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: Standard Statistics			
Clients Attached	ClientsAttachedCount	ClientsAttached	All	Low
API Connections	APIConnectionsCount	APIConnections	All	Low
Errors	ErrorsCount	Errors	All	Low
Writes	WritesCount	Writes	All	Low
Reads	ReadsCount	Reads	All	Low
Writes Blocked	WritesBlockedCount	WritesBlocked	All	Low
Reads Blocked	ReadsBlockedCount	ReadsBlocked	All	Low
Multicast Write (bytes)	MulticastWriteBytesCount	MulticastWriteBytes	All	Low
Multicast Send Messages	MulticastSendMessageCount	MulticastSendMessage	All	Low
Buffered Write (bytes)	BufferedWriteBytesCount	BufferedWriteBytes	All	Low
Buffered Read (bytes)	BufferedReadBytesCount	BufferedReadBytes	All	Low
Message Written (bytes)	MessagesBytesWrittenCount	ClientStats.MessageBytesWritten	All	Low
Message Read (bytes)	MessageBytesReadCount	MessageBytesRead	All	Low
Total Written (bytes)	TotalBytesWrittenCount	TotalBytesWritten	All	Low
Total Read (bytes)	TotalBytesReadCount	TotalBytesRead	All	Low
	Category: Detailed Statistics			
Sent at Highest Priority (bytes)	BytesSentAtHighestPriorityCount	BytesSentAtHighestPriority	All	Low
Sent at Very High Priority (bytes)	BytesSentAtVeryHighPriorityCount	BytesSentAtVeryHighPriority	All	Low
Sent at High Priority (bytes)	BytesSentAtHighPriorityCount	BytesSentAtHighPriority	All	Low
Sent at JMS 9 Priority (bytes)	BytesSentAtJMS9PriorityCount	BytesSentAtJMS9Priority	All	Low
Sent at JMS 8 Priority (bytes)	BytesSentAtJMS8PriorityCount	BytesSentAtJMS8Priority	All	Low
Sent at JMS 7 Priority (bytes)	BytesSentAtJMS7PriorityCount	BytesSentAtJMS7Priority	All	Low
Sent at JMS 6 Priority	BytesSentAtJMS6PriorityCount	BytesSentAtJMS6Priority	All	Low
Sent at JMS 5 Priority (bytes)	BytesSentAtJMS5PriorityCount	BytesSentAtJMS5Priority	All	Low
Sent at JMS 4 Priority (bytes)	BytesSentAtJMS4PriorityCount	BytesSentAtJMS4Priority	All	Low
Sent at JMS 3 Priority (bytes)	BytesSentAtJMS3PriorityCount	BytesSentAtJMS3Priority	All	Low
Sent at JMS 2 Priority (bytes)	BytesSentAtJMS2PriorityCount	BytesSentAtJMS2Priority	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Sent at JMS 1 Priority (bytes)	BytesSentAtJMS1PriorityCount	BytesSentAtJMS1Priority	All	Low
Sent at JMS 0 Priority (bytes)	BytesSentAtJMS0PriorityCount	BytesSentAtJMS0Priority	All	Low
Sent at Low Priority (bytes)	BytesSentAtLowPriorityCount	BytesSentAtLowPriority	All	Low
Sent at Very Low Priority (bytes)	BytesSentAtVeryLowPriorityCount	BytesSentAtVeryLowPriority	All	Low
Sent at Lowest Priority (bytes)	BytesSentAtLowestPriorityCount	BytesSentAtLowestPriority	All	Low
Received at Highest Priority (bytes)	BytesReceivedAtHighestPriorityCount	BytesReceivedAtHighestPriority	All	Low
Received at Very High Priority (bytes)	BytesReceivedAtVeryHighPriority Count	BytesReceivedAtVeryHighPriority	All	Low
Received at High Priority (bytes)	BytesReceivedAtHighPriorityCount	BytesReceivedAtHighPriority	All	Low
Received at JMS 9 Priority (bytes)	BytesReceivedAtJMS9PriorityCount	BytesReceivedAtJMS9Priority	All	Low
Received at JMS 8 Priority (bytes)	BytesReceivedAtJMS8PriorityCount	BytesReceivedAtJMS8Priority	All	Low
Received at JMS 7 Priority (bytes)	BytesReceivedAtJMS7PriorityCount	BytesReceivedAtJMS7Priority	All	Low
Received at JMS 6 Priority (bytes)	BytesReceivedAtJMS6PriorityCount	BytesReceivedAtJMS6Priority	All	Low
Received at JMS 5 Priority (bytes)	BytesReceivedAtJMS5PriorityCount	BytesReceivedAtJMS5Priority	All	Low
Received at JMS 4 Priority (bytes)	BytesReceivedAtJMS4PriorityCount	BytesReceivedAtJMS4Priority	All	Low
Received at JMS 3 Priority (bytes)	BytesReceivedAtJMS3PriorityCount	BytesReceivedAtJMS3Priority	All	Low
Received at JMS 2 Priority (bytes)	BytesReceivedAtJMS2PriorityCount	BytesReceivedAtJMS2Priority	All	Low
Received at JMS 1 Priority (bytes)	BytesReceivedAtJMS1PriorityCount	BytesReceivedAtJMS1Priority	All	Low
Received at JMS 0 Priority (bytes)	BytesReceivedAtJMS0PriorityCount	BytesReceivedAtJMS0Priority	All	Low
Received at Low Priority (bytes)	BytesReceivedAtLowPriorityCount	BytesReceivedAtLowPriority	All	Low
Received at Very Low Priority (bytes)	BytesReceivedAtVeryLowPriority Count	BytesReceivedAtVeryLowPriority	All	Low
Received at Lowest Priority (bytes)	BytesReceivedAtLowestPriorityCount	BytesReceivedAtLowestPriority	All	Low
Messages Sent at JMS 9 Priority	MessagesSentAtJMS9PriorityCount	MessagesSentAtJMS9Priority	All	Low
Messages Sent at JMS 8 Priority	MessagesSentAtJMS8PriorityCount	MessagesSentAtJMS8Priority	All	Low
Messages Sent at JMS 7 Priority	MessagesSentAtJMS7PriorityCount	MessagesSentAtJMS7Priority	All	Low
Messages Sent at JMS 6 Priority	MessagesSentAtJMS6PriorityCount	MessagesSentAtJMS6Priority	All	Low
Messages Sent at JMS 5 Priority	MessagesSentAtJMS5PriorityCount	MessagesSentAtJMS5Priority	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Messages Sent at JMS 4 Priority	MessagesSentAtJMS4PriorityCount	MessagesSentAtJMS4Priority	All	Low
Messages Sent at JMS 3 Priority	MessagesSentAtJMS3PriorityCount	MessagesSentAtJMS3Priority	All	Low
Messages Sent at JMS 2 Priority	MessagesSentAtJMS2PriorityCount	MessagesSentAtJMS2Priority	All	Low
Messages Sent at JMS 1 Priority	MessagesSentAtJMS1PriorityCount	MessagesSentAtJMS1Priority	All	Low
Messages Sent at JMS 0 Priority	MessagesSentAtJMS0PriorityCount	MessagesSentAtJMS0Priority	All	Low
Messages Received at JMS 9 Priority	MessagesReceivedAtJMS9Priority Count	MessagesReceivedAtJMS9Priority	All	Low
Messages Received at JMS 8 Priority	MessagesReceivedAtJMS8Priority Count	MessagesReceivedAtJMS8Priority	All	Low
Messages Received at JMS 7 Priority	MessagesReceivedAtJMS7Priority Count	MessagesReceivedAtJMS7Priority	All	Low
Messages Received at JMS 6 Priority	MessagesReceivedAtJMS6Priority Count	MessagesReceivedAtJMS6Priority	All	Low
Messages Received at JMS 5 Priority	MessagesReceivedAtJMS5Priority Count	MessagesReceivedAtJMS5Priority	All	Low
Messages Received at JMS 4 Priority	MessagesReceivedAtJMS4Priority Count	MessagesReceivedAtJMS4Priority	All	Low
Messages Received at JMS 3 Priority	MessagesReceivedAtJMS3Priority Count	MessagesReceivedAtJMS3Priority	All	Low
Messages Received at JMS 2 Priority	MessagesReceivedAtJMS2Priority Count	MessagesReceivedAtJMS2Priority	All	Low
Messages Received at JMS 1 Priority	MessagesReceivedAtJMS1Priority Count	MessagesReceivedAtJMS1Priority	All	Low
Messages Received at JMS 0 Priority	MessagesReceivedAtJMS0Priority Count	MessagesReceivedAtJMS0Priority	All	Low
Messaging Engine Communications	**ITM Table Name: KYNMECOM**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: SIB Communications > Messaging Engines > Standard Statistics			
Messaging Engine Attached	MEAttachedCount	MEAttached	All	Low
API Connections	APIConnectionsCount	APIConnections	All	Low
Errors	ErrorsCount	Errors	All	Low
Writes	WritesCount	Writes	All	Low
Reads	ReadsCount	Reads	All	Low
Writes Blocked	WritesBlockedCount	WritesBlocked	All	Low
Reads Blocked	ReadsBlockedCount	ReadsBlocked	All	Low
Buffered Write (bytes)	BufferedWriteBytesCount	BufferedWriteBytes	All	Low
Buffered Reads (bytes)	BufferedReadBytesCount	BufferedReadBytes	All	Low
Message Written (bytes)	MessageBytesWrittenCount	MessageBytesWritten	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Message Read (bytes)	MessageBytesReadCount	MessageBytesRead	All	Low
Total Written (bytes)	TotalBytesWrittenCount	TotalBytesWritten	All	Low
Total Read (bytes)	TotalBytesReadCount	TotalBytesRead	All	Low
Durable Subscriptions	**ITM Table Name: KYNDURSUB**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: SIB Service > SIB Messaging Engines > Destinations > Topicspaces > Durable Subscriptions			
Available Message	AvailableMessageCount	AvailableMessageCount	All	Low
Total Messages Consumed	TotalMessagesConsumedCount	TotalMessagesConsumedCount	All	Low
Best Effort Non-persistent Messages Consumed	BestEffortNonPersistentMessages ConsumedCount	BestEffortNonPersistentMessages ConsumedCount	All	Low
Express Non-persistent Messages Consumed	ExpressNonPersistentMessages ConsumedCount	ExpressNonPersistentMessages ConsumedCount	All	Low
Reliable Non-persistent Messages Consumed	ReliableNonPersistentMessages ConsumedCount	ReliableNonPersistentMessages ConsumedCount	All	Low
Reliable Persistent Messages Consumed	ReliablePersistentMessages ConsumedCount	ReliablePersistentMessages ConsumedCount	All	Low
Assured Persistent Messages Consumed	AssuredPersistentMessages ConsumedCount	AssuredPersistentMessages ConsumedCount	All	Low
Aggregate Message Wait Time	AggregateMessageWaitTime	AggregateMessageWaitTime	All	High
Local Message Wait Time	LocalMessageWaitTime	LocalMessageWaitTime	All	High
Local Oldest Message Age	LocalOldestPublicationAge	LocalOldestPublicationAge	All	Max
Queue	**ITM Table Name: KYNMSGQUE**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: SIB Service > SIB Messaging Engines > Destinations > Queues			
Available Message	AvailableMessageCount	AvailableMessageCount	All	Low
Unavailable Message	UnavailableMessageCount	UnavailableMessageCount	All	Low
Local Producer Attaches	LocalProducerAttachesCount	LocalProducerAttachesCount	All	Low
Local Producer	LocalProducerCount	LocalProducerCount	All	Low
Local Consumer Attaches	LocalConsumerAttachesCount	LocalConsumerAttachesCount	All	Low
Local Consumer	LocalConsumerCount	LocalConsumerCount	All	Low
Total Messages Produced	TotalMessagesProducedCount	TotalMessagesProducedCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Best Effort Non-persistent Messages Produced	BestEffortNonPersistentMessages ProducedCount	BestEffortNonPersistentMessages ProducedCount	All	Low
Express Non-persistent Messages Produced	ExpressNonPersistentMessages ProducedCount	ExpressNonPersistentMessages ProducedCount	All	Low
Reliable Non-persistent Messages Produced	ReliableNonPersistentMessages ProducedCount	ReliableNonPersistentMessages ProducedCount	All	Low
Reliable Persistent Messages Produced	ReliablePersistentMessages ProducedCount	ReliablePersistentMessages ProducedCount	All	Low
Assured Persistent Messages Produced	AssuredPersistentMessages ProducedCount	AssuredPersistentMessages ProducedCount	All	Low
Total Messages Consumed	TotalMessagesConsumedCount	TotalMessagesConsumedCount	All	Low
Best Effort Non-persistent Messages Consumed	BestEffortNonPersistentMessages ConsumedCount	BestEffortNonPersistentMessages ConsumedCount	All	Low
Express Non-persistent Messages Consumed	ExpressNonPersistentMessages ConsumedCount	ExpressNonPersistentMessages ConsumedCount	All	Low
Reliable Non-persistent Messages Consumed	ReliableNonPersistentMessages ConsumedCount	ReliableNonPersistentMessages ConsumedCount	All	Low
Reliable Persistent Messages Consumed	ReliablePersistentMessages ConsumedCount	ReliablePersistentMessages ConsumedCount	All	Low
Assured Persistent Messages Consumed	AssuredPersistentMessages ConsumedCount	AssuredPersistentMessages ConsumedCount	All	Low
Report Enabled Messages Expired	ReportEnabledMessages ExpiredCount	ReportEnabledMessages ExpiredCount	All	Low
Aggregate Message Wait Time	AggregateMessageWaitTime	AggregateMessageWaitTime	All	Low
Local Message Wait Time	LocalMessageWaitTime	LocalMessageWaitTime	All	Low
Local Oldest Message Age	LocalOldestMessageAge	LocalOldestMessageAge	All	Low
Topic Spaces	**ITM Table Name: KYNTOPICSP**			
TEP Console Column Name	**WebSphere 7 PMI Attribute**	**WebSphere 8 PMI Attribute**	**WebSphere PMI Level**	**Overhead**
	Category: SIB Service > SIB Messaging Engines > Destinations > Topicspaces			
Incomplete Publication	IncompletePublicationCount	IncompletePublicationCount	All	Low
Local Publisher Attaches	LocalPublisherAttachesCount	LocalPublisherAttachesCount	All	Low
Local Publisher	LocalPublisherCount	LocalPublisherCount	All	Low
Total Local Subscription	TotalLocalSubscriptionCount	TotalLocalSubscriptionCount	All	Low
Non-durable Local Subscription	NonDurableLocalSubscriptionCount	NonDurableLocalSubscriptionCount	All	Low
Durable Local Subscription	DurableLocalSubscriptionCount	DurableLocalSubscriptionCount	All	Low
Total Messages Published	TotalMessagesPublishedCount	TotalMessagesPublishedCount	All	Low

Table 87. TEP Console Workspace Columns to WebSphere PMI Attribute Mapping (continued).

TEP Console Workspace Columns to WebSphere PMI Attribute Mapping

Best Effort Non-persistent Messages Published	BestEffortNonPersistentMessages PublishedCount	BestEffortNonPersistentMessages PublishedCount	All	Low
Express Non-persistent Messages Published	ExpressNonPersistentMessages PublishedCount	ExpressNonPersistentMessages PublishedCount	All	Low
Reliable Non-persistent Messages Published	ReliableNonPersistentMessages PublishedCount	ReliableNonPersistentMessages PublishedCount	All	Low
Reliable Persistent Messages Published	ReliablePersistentMessages PublishedCount	ReliablePersistentMessages PublishedCount	All	Low
Assured Persistent Messages Published	AssuredPersistentMessages PublishedCount	AssuredPersistentMessages PublishedCount	All	Low
Total Local Subscription Hits	TotalLocalSubscriptionHitCount	TotalLocalSubscriptionHitCount	All	Low
Best Effort Non-persistent Local Subscription Hits	BestEffortNonPersistentLocal SubscriptionHitCount	BestEffortNonPersistentLocal SubscriptionHitCount	All	Low
Express Non-persistent Local Subscription Hits	ExpressNonPersistentLocal SubscriptionHitCount	ExpressNonPersistentLocal SubscriptionHitCount	All	Low
Reliable Non-persistent Local Subscription Hits	ReliableNonPersistentLocal SubscriptionHitCount	ReliableNonPersistentLocal SubscriptionHitCount	All	Low
Reliable Persistent Local Subscription Hits	ReliablePersistentLocal SubscriptionHitCount	ReliablePersistentLocal SubscriptionHitCount	All	Low
Assured Persistent Local Subscription Hits	AssuredPersistentLocal SubscriptionHitCount	AssuredPersistentLocal SubscriptionHitCount	All	Low
Report Enabled Publication Expired	ReportEnabledPublications ExpiredCount	ReportEnabledPublications ExpiredCount	All	Low
Local Oldest Publication Age	LocalOldestPublicationAge	LocalOldestPublicationAge	All	Max

Important: Some attributes were renamed between WebSphere Application Server versions 7 and 8. The names for version 8 also apply to version 8.5.

Data sources for workspaces

ITCAM Agent for Application Diagnostics collects data from several sources on the application server. Different workspaces use data from different sources.

Types of data

The Agent collects the following four types of data:

- Resource data
- Request data
- Data from WebSphere log files
- Process data from the operating system

These four categories are based on the source from which the monitoring data is collected. The workspaces in the Tivoli Enterprise Portal display data from these sources. Usually, each workspace displays data from one of the sources. Some workspaces contain summary information collected from one or more sources and processed by the Agent.

Resource Data

ITCAM Agent for WebSphere Applications obtains resource data from the Performance Monitoring Infrastructure (PMI) component that WebSphere Application Server provides. This data mostly contains usage and performance information about a resource in the system. For example, the "DB Connection Pools" workspace provides data about connection pool resources, including how long a connection is checked out, how many threads are waiting for connection, and so on.

Table 88. Workspaces that display resource data, which is obtained from PMI.

Workspace	Description
"Web Applications workspace" on page 335	Displays performance data for each web application (.war). This data includes number of requests, average response time, and number of errors. Note that one Web application may have multiple URLs for user requests and the response times for all the requests are aggregated in this window.
"Servlets/JSPs - Selected Web Application workspace" on page 330	Breaks down the performance data for a particular Web application into the servlets or JavaServer pages (JSPs) that it contains. Again, each servlet/JSP can contain or be mapped to multiple URLs and all user requests are combined.
"Sessions workspace" on page 332	Provides information regarding HTTP sessions created by each Web application. HTTP sessions are used to maintain a user session between multiple invocations and stores user specific data. This workspace also displays the total, average, maximum, and minimum sizes of the user data stored for each Web application.
"EJB Containers workspace" on page 297	Provides an overview of the performance of different types of Enterprise JavaBeans (EJBs) deployed in the application server.
"Enterprise Java Beans workspace" on page 299	Displays the performance data for each EJB deployed in the application. Based on the Bean type, data is available in the appropriate columns.
"Container Transactions workspace" on page 289	Provides performance data regarding Java® Transaction API (JTA) transactions the EJBs are involved in.
"Container Object Pools workspace" on page 288	Provides information about the behavior of stateless and entity bean pools.
"DB Connection Pools workspace" on page 292	Displays the usage data regarding Database connection pools such as the number of connections available, checked out, threads waiting for a connection, and so on. This workspace is useful in understanding the pool usage and identifying potential bottlenecks in application performance when a thread is waiting or getting timed out for a connection from the pool.

Table 88. Workspaces that display resource data, which is obtained from PMI. (continued)

Workspace	Description
"J2C Connection Pools workspace" on page 304	Displays the usage data for connection pools set up for Java Connector Architecture (JCA) based resource adapters. The WebSphere Application Server creates connection pools for the resource adapters that are deployed (some of the adapters are provided by application server installation itself) and this workspace is useful in monitoring performance bottlenecks in terms of wait times to obtain a connection.
"Pool Analysis workspace" on page 313	Provides information about the usage of several types of pools associated with each application server, including Web container pools, ORB pools, J2C connection pools, and database connection pools. This workspace helps you detect resource constraints and potential performance congestion.
"Thread Pools workspace" on page 333	Displays the usage data for thread pools. The data can be used to determine whether the pool is configured correctly to service user requests. For example, the Web Container thread pool is used to execute servlet requests from the users and if the number of threads in the pool is small compared to the number of incoming requests, there will be a delay in servicing them leading to slower response times.
"Thread Pool Trend workspace" on page 334	Displays trend information about thread pool size and usage.
Cache Analysis: Dynamic Cache"Cache Analysis workspace" on page 286	Provides the usage data of the various dynamic caches that have been configured in the system. In WebSphere Application Server 6.x and later, multiple caches can be configured. Also provides the usage data for each dynamic cache template. The templates are unique IDs that are specified in the cachespec.xml files to identify different URLs that must be cached.

Table 88. Workspaces that display resource data, which is obtained from PMI. (continued)

Workspace	Description
"Workload Management workspace" on page 344	Workload management distributes the user requests made through the Object Request Broker (ORB) to different servers in a cluster. This usually means that this feature will come into play only when remote EJB calls are made over the ORB. This feature is applicable only in the ND (Network Deployment) environment when clusters are set up, so that the workload can be distributed when remote EJB calls are invoked over the ORB. If the ORB is not used, this feature is not exercised and there will be no data in this workspace. If a server has both Servlets and EJBs on the same server instance, the calls are usually configured to be local calls instead of remote calls for better performance, and hence the ORB will not come into play. The server side data provides information about the number of requests received on the server side (for example, the EJB container receiving the user requests).
"Scheduler workspace" on page 331	Scheduler service runs periodic tasks. Schedulers are persistent transactional timer services that run Enterprise JavaBean methods or send Java Message Service messages using any JEE server application.
"Web Services workspace" on page 337	Provides performance data on the Web Services hosted by the application server instance. Web Services provides data regarding number of Web services loaded, number of requests delivered, size of the requests, and so on. Also includes information on the Web services gateway, which is used to map an existing service - either an inbound or an outbound service - to a new Web service that appears to be provided by the gateway. The gateway acts as a proxy: your gateway service users need not know whether the underlying service is being provided internally or externally. The gateway provides you with a single point of control, access and validation of Web service requests, and allows you to control which Web services are available to different groups of Web service users.
"Workplace Mail workspace" on page 345	Provides aggregated statistics of the usage information about the incoming message traffic.
"Messages Queues workspace" on page 309	Provides information about message delivery state, including ready retry, unprocessed, and dead.

Table 88. Workspaces that display resource data, which is obtained from PMI. (continued)

Workspace	Description
"Messaging workspace" on page 311	The messaging engines defined in various Service Integration Buses in this application server. Provides performance numbers on each messaging engine such as number of messages published. This workspace displays data only when a Service Integration Bus (SIB) is available in the system.
"Client Communications workspace" on page 287	Provides communication details with distinct client processes that are currently network-connected to the Service Integration Buses of this application server. The data deals with messages sent and received from and to the various client processes.
"Messaging Engine Communications workspace" on page 310	Provides information about other application servers that are hosting messaging engines and are network-connected to this application server. The Service Integration Bus is used to send and receive messages from these messaging engines.
"Destinations workspace" on page 295	Provides performance data and counters for the destinations of a selected messaging engine. This includes the numbers of available and unavailable messages, numbers of messages produced and consumed, aggregated wait times and so on.
"WMQ Client Link Communications workspace" on page 342	Provides information regarding communication to WebSphere MQ JMS clients that are connected to this application server.
"WMQ Link Communications workspace" on page 343	Provides information regarding communication to WebSphere MQ JMS queue managers that are connected to this application server.
"Alarm Manager workspace" on page 281	Provides aggregated information about the alarms for each work manager. This includes the number of alarms created, fired, canceled and so on.
"DCS Stacks workspace" on page 294	Provides aggregated information about each DCS stack within the entire WebSphere Application Server domain, including multiple nodes and servers. This includes the incoming and outgoing message size, the number of incoming and outgoing messages, congestion events, and message buffer re-allocations.
"Durable Subscriptions workspace" on page 296	Provides statistic counters for the durable subscriptions of a selected topic.
"High Availability Manager workspace" on page 302	Provides aggregated information about high availability managers.

Table 88. Workspaces that display resource data, which is obtained from PMI. (continued)

Workspace	Description
"IMAP/POP workspace" on page 303	Provides aggregated statistics of the usage information about the IMAP service and the POP3 service connectivity, especially for the performance-related connectivity.
"Service Components workspace" on page 329	Provides overview performance of the key service components. WebSphere servers feature their own service components, and each of these components has its own set of event points that can be monitored.
"Service Component Elements workspace" on page 327	Lists performance metrics for all the service components and their elements. Service components contain one or more elements, which are sets of different steps processed in each service component. In turn, each element has its own set of event natures, which are key points that are reached when processing a service component element.

The availability of data in these workspaces depends on whether PMI is turned on in the WebSphere Application Server, and on the level of PMI data collection that is turned on.

PMI might be enabled at None, Basic, Extended, All, Custom levels. Each of these levels turns on a set of data counters. The higher level is inclusive of all the counters turned on at the lower level. As you go higher, the cost overhead for the collection increases. The "low" level incurs the least cost (usually obtains data that displays various "counts") and the "maximum" level incurs the highest cost (this, for example, returns time average data that requires extra memory and processing).

There is an additional "custom" level where individual data attributes can be turned on in addition to the data collection levels based on cost.

Hence, to view the Resource data in the portal workspaces, you will need to complete the following steps:

1. Have the appropriate resources defined. For example, JDBC providers and Datasources defined with connection pools to see data in DB Connection pools, Schedulers defined to see any data in the Schedulers workspace, and so on.
2. Turn on the PMI monitoring in the WebSphere Application Server (either through the admin console, wsadmin, or custom scripts).
3. Select the appropriate level of PMI data collection based on the desired data that needs to be monitored.

For a list of the PMI categories and attributes that the Agent workspaces and tables use, see "WebSphere PMI Attribute Mapping" on page 583.

Request Data

The Request data traces user transactions and measures the time to perform various activities (such as database access and JMS operations). Depending on the Request Data workspace, this information is grouped into various categories. For example, in the Request Analysis workspace the data is grouped by the request

name (either the URL for servlets/JSPs or the EJB name) and in DataSources workspaces by the name of the Datasource.

ITCAM Agent for WebSphere Applications tracks the user transactions by modifying application and WebSphere system Java classes by adding method entry and exit "hooks". These hooks are callback methods into the ITCAM Data collector for WebSphere module that will track a user transaction as it travels through the classes in various modules (servlets, EJBs, Datasources, JMS, and so on). The application and WebSphere Application Server system Java classes are modified when they are loaded by WebSphere classloaders. During this load time, based on the type of class - Servlets, EJB, Datasource, JMS, JCA, JNDI, and so on - different callback methods are injected into the class using "Byte Code Modification" (BCM).

Table 89. Workspaces that display request data, which is obtained through Byte Code Modification.

Workspace	Monitoring level	Description
"Request Analysis workspace" on page 317	Level 1	This workspace displays the number of user requests that were tracked during the sampling interval and their average response times. The requests are grouped by URLs for servlet calls and EJB names for direct EJB calls.
"Request Baseline workspace" on page 319	Level 1	This workspace displays aggregated information about the request baseline. In the baselining process, the agent collects statistical information about a request completion times and uses this information to assign "fair" and "bad" thresholds for requests.
"Request Analysis workspace" on page 317	Level 2	In addition to Level 1 data, the workspace also shows the breakdown of the response times into application, SQL Query, SQL Update, JCA, JMS, JNDI, SCA, web services processing times. This breakdown can be displayed in milliseconds and as a percentage of average response time.

Table 89. Workspaces that display request data, which is obtained through Byte Code Modification. (continued)

Workspace	Monitoring level	Description
"Selected Request - Datasources workspace" on page 325 "Selected Request - JMS Queues workspace" on page 325 "Selected Request - Resource Adapters workspace" on page 326 "Selected Request - Service Components workspace" on page 327	Level 2	To access this workspace, clicking the link icon on the row header of each row in the Request Analysis workspace. This workspace provides more details on the resources accessed by the selected user transaction - including the resource names, the average response time, and the longest response time during the sampling interval.
"Data sources workspace" on page 290	Level 2	This workspace displays the time spent on various Data source operations. The operations include obtaining a connection, and executing a query or update. The workspace also shows the number of connections used during the sampling interval. The data in this workspace supplements the "DB Connection Pools workspace" on page 292 by identifying the reason for long connection checkout times.
"JMS Summary workspace" on page 306	Level 2	This workspace provides count and time measurements for JMS operations: send, receive, browse, and publish. It displays data on a queue and topic basis. The data provides a breakdown of the JMS delay shown in the Request Analysis workspace.

The *Monitoring Level* determines the amount of collected request data. At Level 2 the Agent collects more detailed data and incurs additional overhead in terms of memory and processing time. The default is Level 1. To change the value, use the "Start_Request_Monitoring : Begin reporting request data" on page 506 Take Action command.

Another parameter that determines the amount of data collected is the Request Data Sampling Rate. It specifies the percentage of user requests that the Agent tracks and monitors. When the load on the system is very large, tracking every single user request becomes very expensive. Therefore, the Agent uses a sampling approach to monitor the performance of the applications. The default is 2% (i.e. only 2 out of every 100 user requests are tracked).

However, the load on your system might be very low (for example, in a test system you might apply a 10 user load). In this case, you might see no data in the Request Data workspaces with the default setting of 2%, as the probability of tracking one of the 10 requests will be very low. Hence, you need to increase the sampling rate in systems with low load. Use the "Set_Request_Sampling_Rate: Set the sampling rate for request data" on page 505 Take Action command to do this.

Log File Data

ITCAM Agent for WebSphere Applications reads the application server log files and analyzes them to display data in several workspaces.

Table 90. Workspaces that display log file data.

Workspace	Log file	Description
"Garbage Collection Analysis workspace" on page 300	Native error file specified in application server configuration. The file name is set when configuring the Data Collector (the default is `gc.log`).	ITCAM Agent for WebSphere Applications scans the verbose garbage collection (GC) log to provide information about how the GC is performing. This workspace can identify application performance issues when the GC consumes a lot of time. Such issues usually happen when the heap size parameters in the JVM are not set correctly. You can stop and start the collection of GC data using the Start_GC_Monitoring and Stop_GC_Monitoring Take Action commands. **Important:** the Take Action command does not turn on and off the verbose GC flag in WebSphere JVM. This flag is required for collection of GC data. You might need to enable or disable it using the application server admin console.
"Allocation Failures workspace" on page 282	Same as Garbage Collection Analysis	Provides information about heap allocation failures that cause the JVM to invoke garbage collection.
"JVM Stack Trend workspace" on page 307	Same as Garbage Collection Analysis	Provides trend data regarding JVM CPU usage, JVM garbage collection, and JVM heap usage. Displays Operating System data for CPU usage.

Table 90. Workspaces that display log file data. (continued)

Workspace	Log file	Description
"Log Analysis workspace" on page 308	SystemOut.log	The Agent scans the application server SystemOut.log file and retrieves the messages for display in this workspace. Use the workspace to be informed about the health of the application server and also to create situations when certain message IDs are encountered.
"Log Analysis workspace" on page 308	Data Collector message events	Displays the Data Collector diagnostic messages, informing whether the Data Collector has initialized and is working correctly.

Application server process data from the operating system

ITCAM Agent for WebSphere Applications obtains the system CPU usage for the application server process by making operating system calls. This data is displayed in the "Application Server Summary and Business Process Manager Summary workspaces" on page 260 in the **CPU Used (ms)** and **CPU Used (%)** columns. The "JVM Stack Trend workspace" on page 307 also uses this data.

The "OS Stack workspace" on page 312 displays detailed operating system performance information for the application server.

WebSphere summary and aggregated information

The following workspaces contain summary or aggregated information for WebSphere Application Server:

Table 91. Workspaces that display summary and aggregated information.

Workspace	Description
"Application Server Summary and Business Process Manager Summary workspaces" on page 260	This workspace displays overall statistic information for each application server that is monitored by the agent. The information includes history of heap usage, response times, request rates, and % CPU used.
"WebSphere Agent workspace" on page 338	This workspace displays events occurring within the agent and all the application servers on the host. It also displays status information about the agent and the online/offline status of the monitored servers.
"EJB Tier Analysis workspace" on page 320	The workspace displays information about the health of a WebSphere application based on response time and completion rates for EJB requests. It also displays ORB thread pool information and PMI information related to EJB requests.

Table 91. Workspaces that display summary and aggregated information. (continued)

Workspace	Description
"Backend Tier Analysis workspace" on page 322	The workspace displays information about the health of a WebSphere application based on response time for Backend Tier requests. It also displays information about backend tier resources, including Data Sources, JMS resources, JDBC and JCA Pools, and JVM statistics.
"Web Tier Analysis workspace" on page 323	The workspace displays information about the health of a WebSphere application based on response time and completion rates for Web Tier requests. It also displays WebContainer thread pool information and PMI information related to HTTP sessions.

z/OS region workspaces

In a z/OS environment, most workspace tables report data at both a region and server instance level. The advantage is that you can view metrics collected at both levels and switch between server instance lever and region level.

Data Display Problems

If you do not see data in the portal workspaces, use the following checklist to verify the settings.

Workspace	Checkpoint	Comments
All Workspaces	The application server is running and the monitoring agent is connected to it	Check that the **Status** column in the "Application Server Summary and Business Process Manager Summary workspaces" on page 260 shows **Connected**.
All Workspaces	There is user load applied on the application server	The Request data workspaces display data only when user load is applied. The Resource data workspaces still display data rows even if load is not applied, but the columns will either be empty or display zeros. The data rows in this case are created from the application server configuration information about the resource type.

Workspace	Checkpoint	Comments
Request Data Workspaces	Request Data collection is turned on	Verify by checking the Request Data Monitoring Level column in the Application Server Summary workspace is not **Disabled**. If it is **Disabled**, start the monitoring by using the "Start_Request_Monitoring : Begin reporting request data" on page 506 Take Action command.
Request Data Workspaces	Sampling rate is high enough	The default sampling rate is 2%. If the load on the application server is low, the Agent might fail to track user transactions. Verify the sampling rate in the **Request Data Sampling Rate(%)** column in the "Application Server Summary and Business Process Manager Summary workspaces" on page 260. Set the rate higher by using the "Set_Request_Sampling_Rate: Set the sampling rate for request data" on page 505 Take Action command.
"Data sources workspace" on page 290 "JMS Summary workspace" on page 306	Request Data Monitoring Level is set to Level 2	The **Request Data Monitoring Level** column in "Application Server Summary and Business Process Manager Summary workspaces" on page 260 workspace displays the value. To change the monitoring level, use the "Start_Request_Monitoring : Begin reporting request data" on page 506 Take Action command.
"Data sources workspace" on page 290	Data sources are configured in the application server; user requests are accessing these Data sources	To check the presence of Data sources, use the application server admin console. You need to know application logic to verify whether the user requests are accessing the Data sources. Check "DB Connection Pools workspace" on page 292 for activity with the Data sources.

Workspace	Checkpoint	Comments
"JMS Summary workspace" on page 306	JMS resources (queues, topics) are configured and applications are accessing these resources	To check the existence of JMS resources, use the admin console. You need to know application logic to verify whether these resources are used.
Resource Data Workspaces	Feature is available	Verify the application server feature is being used. For example, Dynamic cache data will be available only when the cache feature has been set up. All "Platform Messaging" workspaces will be available only when the Service Integration Bus has been configured and used.
Resource Data Workspaces	PMI is enabled in the application server	Verify PMI is enabled through the administration console of WebSphere Application Server. Make sure PMI is enabled for the desired modules (for example, WebApplications, ServletSessions, EJBContainer, and so on). To do this, check the **Runtime** tab of the **Performance Monitoring Infrastructure** page in the application server admin console.
Resource Data Workspaces	Data not available in certain columns	Verify PMI level is high enough to capture data for these columns. Check the **Instrumentation Level** column in the workspace that you need to view. **Tip:** Change the PMI level through the application server admin console. The contextual help on the data columns in portal workspace specifies the PMI level at which data will be available.
Resource Data Workspaces	Resource Data Monitoring is turned on	The **Resource Data Monitoring** column in "Application Server Summary and Business Process Manager Summary workspaces" on page 260 must say **Enabled**. If it is set to **Disabled**, turn it on using the"Start_Resource_Monitoring: Begin reporting PMI data" on page 507 Take Action command.

Workspace	Checkpoint	Comments
"Garbage Collection Analysis workspace" on page 300	Verbose GC flag is turned on in WAS	Verify the Verbose GC box is checked in the **Application Server->Process Definition->Java Virtual Machine** page of the admin console.
"Garbage Collection Analysis workspace" on page 300	Garbage Collection Monitoring is enabled	The **Garbage Collection Monitoring** column in"Application Server Summary and Business Process Manager Summary workspaces" on page 260 must be **Enabled**. If it set to **Disabled**, use the "Start_GC_Monitoring: Begin reporting garbage-collection data" on page 506 Take Action command to start it. **Important:** the Take Action command will not enable the verbose GC flag in the application server. You have to do it separately.

Glossary for ITCAM Agent for WebSphere Applications

C

class file A file containing Java object code for a single Java class of objects.

class loader A Java component that loads Java object classes into the heap.

component In object-oriented programming environments like Java, a component is a reusable program building block that can be combined with other components running on the same or different computers in a distributed network to form an application. Examples of a component include a single button in a graphical user interface, an interface to a database manager, and an EJB. A component runs within a container.

connection pool A group of host connections that are maintained in an initialized state, ready to be used without having to create and initialize them.

container In the Enterprise JavaBeans (EJB) architecture, a container is an application program that executes the program building block known as a component (a JavaBean). You write a Java container (such as a GUI button or a database requester) that runs both prewritten and site-developed JavaBeans. Examples of containers include pages on a website as well as the web browsers themselves.

contention In the database manager, a situation in which a transaction attempts to lock a row or table that is already locked.

CPU Central Processing Unit. The part of the computer that controls the interpretation and execution of instructions.

CTG CICS Transaction Gateway. CTG integrates your site WebSphere application server applications with key business systems running on CICS servers. CTG integrates middle-tier application servers with CICS by providing a multiuser gateway that supports web applications written in Java, C/C++, Visual Basic, and COBOL. CTG supports three programming interfaces: the External Call Interface (ECI), the External Presentation Interface (EPI), and the External Security Interface (ESI). The latest version of CTG implements the JEE Connector Architecture (JCA), which allows enterprise Java technology to use CICS applications.

D

data sources Data pertaining to JDBC data streams. These are logical connections to database subsystems.

DCE Distributed Computing Environment. A specification from the Open Software Foundation (OSF) that supports remote function execution across a network. WebSphere application server uses DCE to provide certain types of security, for example, authentication of tokens, tickets, or credentials in an untrusted network.

design patterns Written narratives that define a recurring problem, outline a solution, and describe the trade offs involved in implementing that solution. In object-oriented programming environments like Java, a design pattern also describes certain objects and object classes to be used when creating the architecture for such a solution.

DHCP Dynamic Host Configuration Protocol. A communications protocol that lets network administrators centrally manage the assignment of Internet Protocol (IP) addresses across an organization network. Without DHCP, each computer IP address must be entered manually at each computer. If computers move to another location in another part of the network, a new IP address must be chosen. DHCP lets a network administrator supervise and distribute IP addresses from a central point and automatically sends a new IP address when a computer is plugged into a different location in the network.

dynamic cache A consolidation of several caching activities, including servlets, web services, and WebSphere commands into one service where these activities work together to improve performance and share configuration parameters.

E

EJB Enterprise Java Bean. A component architecture for the development and deployment of object-oriented, distributed, enterprise-level applications. Applications written using the Enterprise JavaBeans architecture are scalable, transactional, multi-user, and secure.

G

garbage collection A facility of the JVM that reallocates and frees up memory lost when objects are created and later destroyed. Garbage collection usually occurs when insufficient heap space is available for allocating new objects, although you can trigger it manually by invoking the System.gc method.

Garbage collection can be expensive since it involves identifying all the unreferenced objects, removing them from the heap, and then possibly compacting the heap.

H

heap A memory storage area known formally as the System Resources area that stores system information about running applications. The main storage area used for allocating Java classes and object instances, as well as executable code. The allocation of objects on the heap creates a live or active object, which exists as long as the JVM maintains a pointer to it. When all references to an object cease to exist, it is subject to garbage collection.

HTTP Hypertext Transfer Protocol. A suite of protocols for the Internet that transfer and display hypertext documents.

HTTP sessions Data related to sessions of specific World Wide Web browsers.

I

IIOP Internet Inter-ORB Protocol. A protocol that distributed programs written in different programming languages can use to communicate over the Internet. IIOP, a part of the Common Object Request Broker Architecture (CORBA), is based on the client/server computing model, in which a client program makes requests of a server program that waits to respond to client requests. With IIOP, you can write client programs that communicate with your existing site server programs wherever they are located and without having to understand anything about the server other than the service it performs and its address (called the Interoperable Object Reference, IOR, which comprises the server port number and IP address).

instrumentation The process of modifying program code so the time it takes the code to run, along with other statistics, can be measured. These modifications insert hooks into these Java methods so that the real time and CPU time spent executing the methods can be measured. Instrumentation is key to certain data-collection features (such as lock analysis) of IBM Tivoli Composite Application Manager Agent for WebSphere Applications.

IBM Tivoli Monitoring platform The software architecture and foundation that support the development and operations of Tivoli Enterprise Portal and its Tivoli Enterprise Monitoring Agents.

J

J2C WebSphere application server implementation of the connection-management and pool-management components of JCA. Applications needing a backend resource (such as a CICS transaction) reference a connection factory suitable for use with that resource. The connection factory in turn calls the WebSphere application server DB2 Relational Resource Adapter, DB2 RRA (which, despite its name, also supports Oracle, SQL Server, and other database management systems). The DB2 RRA passes the connection request to a J2C connection manager, which tries to provide an available connection from a predefined pool. But when all connections are in use, the J2C connection manager calls back to the DB2 RRA to obtain a new connection from the backend system. When this connection is returned, the connection manager adds it to the connection pool for future reuse.

J2C container The third type of container within WebSphere application server (EJB containers and web containers are the other two types). A J2C container contains connection factories that represent backend resources like CICS. It also manages access to these resources via a connection manager associated with each connection factory.

JEE Java Platform, Enterprise Edition. An environment for developing and deploying enterprise applications using the Java language. The JEE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functions needed for developing multi-tiered, Web-based applications.

JAR file A Java archive file, which is a compressed file containing many class files along with their directory structure.

JavaBean The Java implementation of component-based software architecture, which defines how small, tested software entities called components can be integrated to build complete software solutions (called containers).

JCA JEE Connector Architecture. A template for writing your own connectors, which is a binding between a JEE application server and a backend enterprise information system (EIS) or data repository such as CICS, IMS, or DB2.

JDBC Java Database Connectivity. An application programming interface (API) for connecting Java programs to the data in a relational database. With this API, you can encode data requests in Structured Query Language (SQL) that JDBC then passes to the database manager for interpretation and processing.

JMS Java Message Service. A programming interface that connects Java programs to middleware messaging applications, for example, WebSphere MQ.

JMX Java Management Extensions. A standard technology for management and monitoring of Java applications and Java environments that defines architecture, application programming interfaces (APIs), and services for application and network management. JMX provides tools for building distributed, Web-based solutions for managing and monitoring Java devices and applications.

A JMX agent is an Mbean container.

JNDI Java Naming and Directory Interface. A programming interface that connects Java applications to naming and directory servers such as LDAP.

JSP JavaServer Page. A web page that specifies one or more servlets whose execution on the web server modifies the page content or appearance before it is presented to the user.

JTA Java Transaction API. An application programming interface developed by Sun Microsystems that specifies standard Java interfaces between a transaction manager (such as Tuxedo) and other players in a distributed transaction system: the resource manager, the application server, the transactional applications, and often the database manager as well.

JVM Java virtual machine. A software implementation of a central processing unit (CPU) that runs Java applets and applications. The JVM provides a software execution engine that safely and compatibly executes the byte codes in Java class files on various microprocessors, whether embedded in a computer or in another electronic device.

JVMPI Java Virtual Machine Profiling Interface. An experimental interface that provides profiling hooks into the Java virtual machine. This interface supports two-way function calls between the JVM and a profiler agent: the virtual machine

notifies the agent of various events, and the agent issues requests for control information.

K

Kerberos A security system from the Massachusetts Institute of Technology that provides security services for networking.

L

LDAP Lightweight Directory Access Protocol. A protocol that uses TCP/IP to access directories. Applications can use these directories for storing common data and for retrieving data. For example, applications can use LDAP to access such information as email address, service configuration parameters, or public keys.

lock A semaphore created through the use of the Java synchronized keyword that prevents simultaneous access to a Java object or section of code.

logstream A system logger application that records error information, unexpected conditions, or failures detected within WebSphere application server code. Logstreams are specific to the z/OS platform.

M

MBeans Managed Beans. A set of standard instrumentation methods for use in Java programs and by Java management applications (such as IBM Tivoli Composite Application Manager Agent for WebSphere Applications) that allows monitoring and management of Java-based applications. An MBean is a JavaBean that represents a JMX-manageable resource.

monitor 1) A transaction environment for maintaining large quantities of data in a consistent state and that controls which users and clients can access data through authorized servers. 2) A programming primitive created so multiple program threads can share the same resource (such as an object). A program creates a monitor for a given resource by requesting it from the system; the system returns a unique ID for that monitor. After that, any thread needing the resource must use the monitor to lock the resource while the thread is using it. If the monitor is already locked, a thread requesting the resource is queued by the system and then given control when the monitor becomes unlocked. Also called a mutex.

MVC The Model/View/Controller design pattern is based on the separation of the application into three objects: the user interface ("view") , the logical structure of the data the application requires ("model"), and the user commands that change the view or the model ("controller"). Design patterns themselves are written narratives that define a recurring problem, outline a solution, and describe the trade offs involved in implementing that solution. (In object-oriented programming environments like Java, a design pattern also describes certain objects and object classes to be used when creating the architecture for such a solution).

N

node Any managed system, such as a WebSphere application server, that the Tivoli Enterprise Portal is monitoring. A node can also be a managed system of subnodes being managed under a primary node.

O

OLT Object Level Trace. A tool for testing distributed applications that include a graphical trace facility and a remote debugger.

ORB Object Request Broker. In object-oriented programming, software that serves as an intermediary by transparently enabling objects to exchange requests and responses.

P

PMI Performance Monitoring Infrastructure. The application programming interface (API) that IBM provides for extracting WebSphere application server performance data. PMI must be enabled and the appropriate instrumentation levels set for the Tivoli Enterprise Monitoring Agent to extract PMI data and populate certain workspaces. PMI runs as an interface to JMX.

portal A Web-based application that commonly provides personalization, single signon, and content aggregation from different sources, and hosts the presentation layer of Information Systems (IS). Portals are becoming the next-generation desktop, delivering ebusiness applications over the web to many types of client devices.

portlet A Java technology-based web component, managed by a portlet container, that process requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to various heterogeneous Enterprise Information Systems (EIS) or Information Systems (IS).

R

request The entry point into an application whose processing or response time directly affects the response time the end user perceives. In IBM Tivoli Composite Application Manager Agent for WebSphere Applications, a request is a servlet, JSP, or EJB method. Also called a transaction.

request analysis A configuration technique for a request (that is, a transaction) within WebSphere application server that allows you to see how often the request runs and what factors about the request degrade its performance. Request analysis involves collecting and aggregating request data on each request that runs. Its primary purpose is to report on requests running in the application server, how often they run, their response time, and what is causing performance degradation.

Request analysis works by dynamic object code instrumentation, which modifies the Java object code for the application classes as they are being loaded into the JVM.

resource adapter An implementation of the JEE Connector Architecture Specification that provides applications with access to resources outside of the server or provides server applications with access to an Enterprise Information System (EIS). A resource adapter can provide applications access to resources such as DB2, CICS, SAP, and PeopleSoft. It can provide an EIS with the ability to communicate with message-driven beans configured on the server. A resource adapter is implemented in a resource adapter archive file, which has an extension of .rar. A resource adapter can be provided as a stand-alone adapter or as part of an application, in which case it is referred to as an embedded adapter.

RMI Remote Method Invocation. A Java standard from Sun Microsystems that performs a remote procedure call (RPC) to allow Java objects stored in the network to be accessed remotely in a client/server environment.

RPC Remote Procedure Call. A protocol based on the Distributed Computing Environment (DCE) of the Open Software Foundation that allows one program to request services from a program running on another computer in a network. (A procedure call is also known as a function call or a subroutine call.) RPC uses the client/server model: the requesting program is the client, and the responding program is the server. As with a local procedure call, an RPC is a synchronous operation: the requesting program is suspended until the remote procedure returns its results.

S

SCA Service Component Architecture. An architecture in which all elements of a business transaction, such as web service access, Enterprise Information System (EIS) service assets, business rules, workflows, databases, and so on, are represented in a service-oriented way.

servlet A Java application that runs in an application server or web server, and that provides server-side processing.

session A series of requests to a servlet that originate from the same user at the same browser. Sessions allow applications running in a web container to tracks individual users.

signature The name of an operation or method and its parameters.

SSL Secure Sockets Layer. A security protocol for communication privacy to provide secure client-server conversations.

struts A widely used framework for developing servlet- or JSP-based web applications that is distributed as open source by the Apache Jakarta Project. Struts encourages application architectures based on the Model 2 approach, a variation of the Model-View-Controller (MVC) design paradigm.

T

thread A dispatchable unit of work.

thread pool The threads that are being used by or are available to a computer program.

transaction A single execution of a workload.

U

URI Uniform Resource Identifier. An identifier for a point of content on the Internet, be it a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the web page address, which is a particular form of URI called a Universal Resource Locator (URL). A URI typically describes the mechanism used to access the resource, the computer that houses the resource, and the name of the resource (such as a file name) on that computer.

URL Universal Resource Locator. The unique address for a file accessible via the Internet. Such a file might be a web page (usually the home page), an image file, or a program such as a Java applet or servlet. The URL comprises the protocol used to access the file, a domain name that identifies a specific computer on the Internet, and a path name that specifies the file location on that computer.

W

Web container A special JEE container that manages web applications in a JEE architecture. A web container specifies a runtime environment for web components, including security, concurrency, lifecycle management, transaction, deployment, and other services. It also enables a web application to access external resources such as relational databases (via JDBC) and Enterprise Java Beans. A web container provides the same services as a JSP container and provides support for managing JavaServer Pages.

Every JEE server contains at least one web container.

workload management The optimization of the distribution of incoming work requests to the application servers, enterprise beans, servlets, and other objects that can effectively process the request.

Chapter 5. ITCAM Agent for J2EE

IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE provides a Systems Management solution for the J2EE Application Server Version 6.2 for distributed platforms. Using ITCAM Agent for J2EE, you can monitor multiple J2EE application servers running on the same physical node. Each application server must be configured with its own IBM Tivoli Composite Application Manager (ITCAM) for J2EE data collector.

The Tivoli Enterprise Monitoring Agent collects performance data from the following four primary sources:

- Response time data for application server requests from the ITCAM for J2EE data collector
- Resource data from the Performance Monitoring Infrastructure (PMI) of J2EE
- J2EE Application Server log messages
- Garbage-collector activity recorded by the JVM verboseGC trace

Attributes within the product collect data about the inner workings of an application server and performance information about user applications running under its control.

Initiating data collection and reporting of data

Because of high processor usage, some data items are not automatically collected and reported. The collection of some data and statistics depends upon the setting of instrumentation levels for certain attributes. If the instrumentation levels are not set appropriately, certain information is not collected and displayed in the workspaces. Similarly, those attributes that collect request and application trace data require you to complete several configuration steps. To collect this data, use one of the following methods to reconfigure data collection:

- Complete configuration steps (as explained in the ITCAM Agent for J2EE installation and customization guide).
- Issue Take Action commands, with which you can take specific action against your J2EE application server using the Tivoli Enterprise Portal.
- Use Manage Tivoli Enterprise Services (as explained in the various IBM Tivoli Monitoring installation manuals and the ITCAM Agent for J2EE installation and customization guide).

Automatic baselining

To display application health status, ITCAM monitors request response times (averaged over a sampling interval, by default 60 seconds) for every application. Every top-level request available in an application is monitored separately.

For every request, two *thresholds* are set, known as *fair* and *bad*. When at least one average request response time for an application rises over the fair threshold, a health warning (yellow) for this application is reported. Similarly, when at least one average request response time rises over the bad threshold, an application health alarm (red) is reported.

© Copyright IBM Corp. 2006, 2015

ITCAM also monitors the nested requests (for example, database calls) within every top-level request. In the event of a warning or alarm, ITCAM checks which of the nested requests is taking more than its usual share of time. Depending on the type of such nested requests, ITCAM shows whether the client, application, or backend tier is the likely cause of the warning/alarm. Servlet and Portal request types are assigned to the client tier. EJB and User (Custom) request types are assigned to the application tier. All other request types (JNDI, JDBC, JCA, JMS) are assigned to the backend tier.

When ITCAM starts to monitor a new application, it automatically starts a *baselining process*. This process normally runs for seven days but provides updated information every hour from the beginning. During the process, ITCAM collects statistical data for all requests in this application. When the data is collected, ITCAM sets the thresholds automatically; it also records the typical share of response time for each nested request type.

In most cases, this automatic setting is adequate. During the baselining process, the baselines get updated periodically, and the alarms/warnings start to correspond to real problems. You do not have to adjust baselining settings when things are working normally. (The automatic thresholds usually become usable earlier, after the application has been observed through its typical load patterns). If you need to acquire thresholds, based on whatever data is available, before the hourly automatic update, you can manually update baselining.

However, in some situations the threshold levels can become inadequate. This results in either too many false alarms/warnings, or in real problems going undetected. Such situations can be split broadly into two categories:

- If time has passed since the baselining process for an application, its response times might have changed because of configuration alteration, database growth, changing load patterns, and so on. In this case, you might have to run the baselining process again. It is good practice to do it after any configuration or infrastructure change.
- If the thresholds are incorrect immediately after the baselining process has been completed, you might have to adjust the auto threshold settings.

As a last resort, you can also override the thresholds with fixed values. However, do not do this unless you know a lot about the monitored application, or unless instructed by IBM Level 3 Support.

If you need the thresholds set before they are updated automatically for the first time, you can trigger a baseline update. This immediately sets the thresholds based on the request data collected so far.

Additional information

For additional usage information about this agent, see:

- Workspaces
- Attributes
- Situations
- Take Action commands

Workspaces for ITCAM Agent for J2EE

As part of the IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE product integration with the Tivoli Enterprise Portal, the workspaces offer views of monitoring data that provide detailed, current data about the J2EE application servers running on the UNIX and Windows platforms on your site. In addition to reports and graphs, a workspace can contain other views (that is, windows), such as a Notepad editor session, a browser session, a telnet session, an event console, or a Take Action view from which you can issue commands.

Several views of high-level information

Several workspaces provide high-level information to help you meet the monitoring and administrative needs of your site. These workspaces report on status and availability for both the J2EE administrative server and its application server instances. They enable you to easily monitor the availability of your enterprise, the J2EE Application Server, and application server instances.

Primary and secondary workspaces

The workspaces listed in the Navigator are directly accessible and are called *primary workspaces*. Some of these also contain *secondary workspaces*, which are not accessible directly from the Navigator. Instead, you must select and display the primary workspace and then use either a menu option or a special link icon in the primary workspace views to reach the secondary workspaces (sometimes called subsidiary workspaces).

Workspaces with historical data links

Several workspaces provide secondary workspaces that display historical data. You can specify a time span over which to collect historical data, which accumulates and summarizes the data in the primary workspaces that generate them. (The default setting is 15 minutes; you can modify this setting to suit your needs.) The descriptions of the historical workspaces follow the descriptions of the primary workspaces that generate them in the workspace helps.

Available Tivoli Enterprise Portal workspaces

For an overview of the organization of the available workspaces, see Organization of the predefined workspaces.

Organization of the predefined workspaces

The IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE workspaces for the Tivoli Enterprise Portal define data shows that display in the physical view of the Navigator. In addition to the workspaces that the Navigator lists, you can reach their subsidiary (or secondary) workspaces from the primary workspaces (those listed in the Navigator).

Accessing the subsidiary workspaces

You can access the subsidiary workspaces of a primary workspace by using one or more of the following methods:

- From the Navigator:
  1. Select the primary workspace.

2. Right-click the name of the selected workspace in the Navigator.
3. Select **Workspaces** from the menu.
4. Select the desired subsidiary workspace.

- From the View menu:
  1. Select the primary workspace.
  2. Select **View > Workspaces**.
  3. Select the desired subsidiary workspace.
- From a report:
  1. Select the primary workspace.
  2. If the workspace report contains a link, as shown in the following example:



     You can either click the icon to select the row or right-click the icon and select a subsidiary workspace from the menu.
- From a chart view:

  The data displayed in some bar charts and plot charts is linked to subsidiary workspaces. To search for a link, right-click a bar or data point in the chart. If **Link to** displays in the menu, you can select a subsidiary workspace pertaining to the data in the chart.

Workspace organization

The hierarchy levels shown in the Navigator depend upon the customization of the Tivoli Enterprise Portal by your enterprise. However, ITCAM Agent for J2EE provides a set of predefined workspaces, which do not require customization. The following list shows the order and hierarchy of the predefined workspaces provided by the IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE Tivoli Enterprise Monitoring Agent. It is a representation of how the predefined workspaces are organized in the Navigator. For more detailed information about a workspace, click its name in the list.

operating system [for example, Windows]

- *system* [or nodename]
  –
  – J2EE Agent
    - JBoss App Server
      - Application Health Summary
        – Selected Application - Application Tier Analysis
        – Selected Application - Configuration
        – Selected Application - Backend Tier Analysis
        – Selected Application - Request Analysis
        – Selected Application - Health History
        – Selected Application - Client Tier Analysis
        – Application Registry
          - Selected Request - Baseline

    - Request Analysis
      - Selected Request - Datasources
      - Selected Request - JMS Queues
      - Selected Request - Resource Adapters
      - Selected Request - History
    - Garbage Collection Analysis
      - Allocation Failures
      - Garbage Collections - Selected Allocation Failure
    - Log Analysis
    - Datasources
      - Selected Datasource - History
    - JMS Summary
    - Web Applications
      - Servlets / JSPs - Selected Web Application
    - EJB Modules
      - Enterprise Java Beans
    - JCA Connection Pools
    - JTA Resources
- SAP NetWeaver Server Workspace
    - Application Health Summary
      - Selected Application - Application Tier Analysis
      - Selected Application - Configuration
      - Selected Application - Backend Tier Analysis
      - Selected Application - Request Analysis
      - Selected Application - Health History
      - Selected Application - Client Tier Analysis
      - Application Registry
        - Selected Request - Baseline
    - Request Analysis
      - Selected Request - Datasources
      - Selected Request - JMS Queues
      - Selected Request - Resource Adapters
      - Selected Request - History
    - Garbage Collection Analysis
      - Allocation Failures
      - Garbage Collections - Selected Allocation Failure
    - Log Analysis
    - Datasources
      - Selected Datasource - History
    - JMS Summary
    - Web Container
    - Enterprise Java Beans
    - DB Connection Pools
      - Selected DB Connection Pool - History
    - JTA Summary

- Tomcat Server Workspace
  - Application Health Summary
    - Selected Application - Application Tier Analysis
    - Selected Application - Configuration
    - Selected Application - Backend Tier Analysis
    - Selected Application - Health History
    - Selected Application - Client Tier Analysis
    - Application Registry
      - Selected Request - Baseline
  - Request Analysis
    - Selected Request - Datasources
    - Selected Request - JMS Queues
    - Selected Request - Resource Adapters
    - Selected Request - History
  - Garbage Collection Analysis
    - Allocation Failures
    - Garbage Collections - Selected Allocation Failure
  - Log Analysis
  - Datasources
    - Selected Datasource - History
  - JMS Summary
- Oracle App Server Workspace
  - Application Health Summary
    - Selected Application - Application Tier Analysis
    - Selected Application - Configuration
    - Selected Application - Backend Tier Analysis
    - Selected Application - Health History
    - Selected Application - Client Tier Analysis
    - Application Registry
      - Selected Request - Baseline
  - Request Analysis
    - Selected Request - Datasources
    - Selected Request - JMS Queues
    - Selected Request - Resource Adapters
    - Selected Request - History
  - Garbage Collection Analysis
    - Allocation Failures
    - Garbage Collections - Selected Allocation Failure
  - Log Analysis
  - Datasources
    - Selected Datasource - History
  - JMS Summary
  - Web Applications
    - Servlets / JSPs - Selected Web Application
  - EJB Modules

      - Enterprise Java Beans
- BEA WebLogic App Server Workspace
  - Application Health Summary
    - Selected Application - Application Tier Analysis
    - Selected Application - Configuration
    - Selected Application - Backend Tier Analysis
    - Selected Application - Request Analysis
    - Selected Application - Health History
    - Selected Application - Client Tier Analysis
    - Application Registry
      - Selected Request - Baseline
  - Request Analysis
    - Selected Request - Datasources
    - Selected Request - JMS Queues
    - Selected Request - Resource Adapters
    - Selected Request - History
  - Garbage Collection Analysis
    - Allocation Failures
    - Garbage Collections - Selected Allocation Failure
  - Log Analysis
  - Datasources
    - Selected Datasource - History
  - JMS Summary
  - Web Applications
    - Servlets / JSPs - Selected Enterprise Application
  - EJB Components Workspace
    - EJBs - Selected Enterprise Application
  - JDBC Connection Pools
    - Selected JDBC Connection Pool - History
  - JCA Connection Pools
  - JMS Sessions
  - JTA Resources
- WebSphere App Server CE
  - Application Health Summary
    - Selected Application - Application Tier Analysis
    - Selected Application - Configuration
    - Selected Application - Backend Tier Analysis
    - Selected Application - Request Analysis
    - Selected Application - Health History
    - Selected Application - Client Tier Analysis
    - Application Registry
      - Selected Request - Baseline
  - Request Analysis
    - Selected Request - Datasources
    - Selected Request - JMS Queues

        - Selected Request - Resource Adapters
        - Selected Request - History
      - Garbage Collection Analysis
        - Allocation Failures
        - Garbage Collections - Selected Allocation Failure
      - Log Analysis
      - Datasources
        - Selected Datasource - History
      - JMS Summary
      - Web Applications
        - Servlets / JSPs - Selected Web Application
    - J2SE Application
      - Request Analysis
        - Selected Request - Datasources
        - Selected Request - JMS Queues
        - Selected Request - Resource Adapters
        - Selected Request - History
      - Garbage Collection Analysis
        - Allocation Failures
        - Garbage Collections - Selected Allocation Failure
      - DC Message Events
      - Datasources
        - Selected Datasource - History
      - JMS Summary
      - JVM Statistics

For additional information, see:

"Attribute groups used by the predefined workspaces" on page 666

Configuration workspaces

Monitoring Agent configuration and tuning is facilitated in the Tivoli Enterprise Portal through *configuration workspaces*. There are two configuration workspaces: one for the Monitoring Agent level and one for the server level:

- J2EE Agent Configuration workspace
- application server Configuration workspace

Both configuration workspaces have the same tabs and configuration settings.

The J2EE Agent Configuration workspace settings are initial settings applied to all data collectors managed by the Monitoring Agent. For example, when a data collector connects to the Monitoring Agent for the first time or if the data collector configuration was deleted. In both these examples, the configuration settings specified in the J2EE Agent Configuration workspace are applied.

The application server Configuration workspace contains individual server settings and the settings in this workspace override the settings in the J2EE Agent Configuration workspace.

Settings for the J2EE Agent Configuration and application server Configuration workspaces are saved in the following files:

Table 92. Workspace configuration files

Workspace	Configuration file
J2EE Agent Configuration workspace on Windows systems:	*CANDLE_HOME*\TMAITM6*hostname_productcode*.xml
Application server Configuration workspace on Windows systems:	*CANDLE_HOME*\TMAITM6\ *hostname_productcode_servervendor.servernode.server name*.xml
J2EE Agent Configuration workspace on other platforms:	*CANDLE_HOME*/config/*hostname_productcode*.xml
Application server Configuration workspace on other platforms:	*CANDLE_HOME*/config/ *hostname_productcode_servervendor.servernode.server name*.xml

The J2EE Agent Configuration workspace configuration settings are initial settings which are applied to all data collectors managed by the selected J2EE agent. The application server Configuration workspace configuration settings override agent level configuration settings.

Both configuration workspaces have the following two views:

- **Application Diagnostics Configuration view** - this view has the following four tabs
  - Collection (Basic)
  - Collection (Advanced)
  - Application Dashboard (Basic)
  - Application Dashboard (Auto Threshold)
- **Application Servers view** - this view has one tab which lists all application servers monitored by the selected J2EE agent, for each application server, there is a link to application server Configuration workspace.

Application Diagnostics Configuration - Basic Tab

The Basic tab contains the following fields:

- **Request Data Monitoring** - in this field, specify the monitoring level for requesting data from connected data collectors. The following values can be entered into this field:
  - **Disable** - request data is not monitored and is not displayed
  - **Level 1** - only monitors edge request data, for example, servlets, JSPs, EJBs, SCA requests, and web service requests
  - **Level 2** - monitors nested request data in addition to edge request data, for example, JNDI, JMS, JDBC, and JCA requests

Important: This monitoring level is applied by default to newly discovered applications. However, you can set the monitoring level for any operation separately, using the "Set_Application_Monitoring: Set monitoring" on page 746 Take Action command.

- **Request Data Monitoring Method** - in this field, specify the monitoring method used by the Monitoring Agent to govern when it uploads request and garbage collection data from connected data collectors. The following values can be entered into this field:
  - **On Demand** - when the monitoring method is set to On Demand, data is uploaded only when requested by the user. The exception to this is if the cache of data is still current. Whether this data is current is determined by the Request Data On Demand Maximum Sample Age (sec) field in the Collection Advanced tab. With On Demand monitoring, the Tivoli Enterprise Portal response time is slower because the data is collected as requested. However, the retrieved data is the most current available. CPU and memory usage on the monitored systems is lower for On Demand monitoring.
  - **Fixed Interval** - when the monitoring method is set to Fixed Interval, the Monitoring Agent will upload sample data from the data collectors at regular fixed intervals, and respond to a user request using the latest cache of sample data gathered at the last interval. With Fixed Interval monitoring, CPU and memory usage can be higher because more data is collected more frequently but Tivoli Enterprise Portal response time is faster as the data is more readily available. Data samples are calculated for the same time interval and as a result are more consistent.
- **Resource Data Monitoring** - in this field, specify whether resource data is monitored from connected data collectors. The following values can be entered in this field:
  - **Disable** - resource data is not collected by the Monitoring Agent and is not displayed in the Tivoli Enterprise Portal.
  - **Enable** - resource data is collected by the Monitoring Agent and is displayed in the Tivoli Enterprise Portal.
- **Resource Data Monitoring Method** - in this field, specify the monitoring methodology used by the agent to govern when it uploads resource data from connected data collectors. The following values can be entered into this field
  - **On Demand** - when the monitoring method is set to On Demand, data is uploaded only when requested by the user. The exception to this is if the cache of data is still current. Whether this data is current is determined by the Resource Data On Demand Maximum Sample Age (sec) field in the Collection Advanced tab. With On Demand monitoring, the Tivoli Enterprise Portal response time is slower as the data is collected as requested. However, the retrieved data is the most current available. CPU and memory usage on the monitored systems is lower for On Demand monitoring.
  - **Fixed Interval** - when the monitoring method is set to Fixed Interval, the Monitoring Agent will upload sample data from the data collectors at regular fixed intervals, and respond to a user request using the latest cache of sample data gathered at the last interval. With Fixed Interval monitoring, CPU and memory usage can be higher because more data is collected more frequently but Tivoli Enterprise Portal response time is faster as the data is more readily available. Data samples are calculated for the same time interval and as a result are more consistent.
- **Garbage Collection Monitoring** - in this field, specify if verbose garbage collection output monitoring is enabled. The following values can be entered in this field:

- **Disable** - verbose garbage collection output data is not collected by the Monitoring Agent and is not displayed in the Tivoli Enterprise Portal.
- **Enable** - verbose garbage collection output data is collected by the Monitoring Agent and is displayed in the Tivoli Enterprise Portal.

Application Dashboard (Basic) tab

The Application Dashboard (Basic) tab contains the following fields:

- **Application Fair Completion Rate Threshold (%)** - Defines the default completion percentage for application fair availability threshold.
- **Application Bad Completion Rate Threshold (%)** - Defines the default completion percentage for application bad availability threshold.
- **Application Fair Resource Usage Threshold (%)** - Defines the threshold percentage for fair usage level of an application resource.
- **Application Bad Resource Usage Threshold (%)** - Defines the threshold percentage for bad usage level of an application resource.
- **Application Resource Usage Monitoring Cutoff Threshold (%)** - Defines the cutoff threshold percentage for application resources usage monitoring.
- **Request Monitoring Control Level** - Specifies the request monitoring control level for the server. The following entries can be entered to this field:
  - **Application** Request monitoring settings are defined for each application independently.
  - **Server** Request monitoring settings are defined on the server level

The Request Monitoring Control Level option gives you more control over request monitoring settings. In certain sequences, you can benefit from locking the request monitoring control on the server level, because you can change data collector monitoring level in one place.

In ITCAM For Application Diagnostics, the request data monitoring level (Level1 or Level2) that displays on the Tivoli Enterprise Portal depends on the **Request Monitoring Control Level** setting you choose.

- If you select the **Request Monitoring Control Level** as **Application** (the default setting), the monitoring level is set separately for every application. For a newly discovered application, the current monitoring level for the agent is applied. Therefore, to display Level2 data for an application in the Tivoli Enterprise Portal, you must set the monitoring level for the application using the “Set_Application_Monitoring: Set monitoring” on page 746 Take Action command.
- If you select the **Request Monitoring Control Level** as **Server**, the level is set for the server and affects all applications. If you set the Request Data Monitoring Level at Level2 using the “Start_Request_Monitoring : Begin reporting request data” on page 749 Take Action command, the Request Data Monitoring Level for all the applications in this server is Level2. While you can still use the **Set_Application_Monitoring** Take Action command to set the monitoring level for an individual application, the level does not apply as a server-wide level is in operation. To apply the levels set for individual applications, change the **Request Monitoring Control Level** setting to **Application**.

Collection Advanced tab

The Collection Advanced tab contains the following fields:

- **Request Data On Demand Maximum Sample Age (sec)** - The maximum allowed age of sample request data in seconds before collecting a new sample of data. If the monitoring method is set to On Demand, when a user request is

received, and the current sample cache is older than the value specified, then the Monitoring Agent uploads a new sample before servicing the request. Two successive on-demand requests received from users within the period specified by the maximum sample age return the same results without incurring the CPU and memory cost of a new data sample.

- **Request Data Fixed Interval between Collections (sec)** - The amount of time in seconds between uploads of sample request data from the data collectors to the Monitoring Agent when the monitoring method is set to Fixed Interval. When a user request is received, it is serviced from the latest uploaded sample.
- **Request Data Sampling Rate (%)** - The percentage of requests that are sampled for request data monitoring.
- **Resource Data On Demand Maximum Sample Age (sec)** - The maximum allowed age of sample resource data in seconds before collecting a new sample of data. If the monitoring method is set to On Demand, when a user request is received and the current sample cache is older than the value specified in this field, then the Monitoring Agent uploads a new sample before servicing the request. Two successive on-demand requests received from a user within the period specified by the maximum sample age return the same results without incurring the CPU and memory cost of a new data sample.
- **Resource Data Fixed Interval between Collections (sec)** - The amount of time in seconds between uploads of sample resource data from the data collectors to the agent. When a Tivoli Enterprise Portal request is received, it is serviced from the latest uploaded sample.
- **Garbage Collection Polling Interval (sec)** - The interval in seconds between the Monitoring Agent scanning the verbose Garbage Collection output.
- **Log Scan Polling Interval (sec)** - The interval in seconds between the Agent scanning the application server standard output log for changes.

Application Dashboard (Auto Threshold) tab

The Application Dashboard (Auto Threshold) tab contains the fields:

- **Response Time Selection (%)** - Defines the percentage from baseline to be used for response time auto-thresholding.
- **Response Time Deviation (%)** - Defines the deviation for baseline selection to be used for response times auto-thresholding.
- **Fair Response Time Projection (%)** - Defines the percentage to derive the fair response time threshold from the baseline selection.
- **Bad Response Time Projection (%)** - Defines the percentage to derive the bad response time threshold from the baseline selection.

Allocation Failures workspace

This workspace summarizes all the heap-allocation failures that occurred within the Java Virtual Machine (JVM) over the current interval and that caused the JVM to initiate garbage collection.

This workspace displays data provided by the Allocation Failure attributes.

Note to Solaris and HP-UX users: Allocation-failure information is not recorded on these platforms. Consequently this workspace is always empty.

The predefined workspace contains the following items:

- Allocation Failure Elapsed Times bar chart, which displays the number of allocation failures during the current interval.

- Heap Usage bar chart, which displays the heap usage for this JVM. The hover help associated with the bar gives the allocation-failure ID number followed by a range of recording times. This allocation-failure number displays in the Allocation Failures report and associates each bar with that particular row within the report.
- Allocation Failures report, which displays information about the heap-allocation failure that caused the JVM hosting the application server to start its garbage-collection routine.

Accessing the Allocation Failures workspace

To access this workspace from the "Garbage Collection Analysis workspace" on page 643, right-click the link icon, and from the menu, click **Allocation Failures**.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Health Summary workspace

The workspace displays the information about the real-time health status of applications monitored by the Tivoli Enterprise Monitoring Agent.

The health status information is collected from the following sources:
- Request Metrics - performance data that measures request execution time collected from the ITCAM instrumentation points in the application code.
- Garbage Collection Metrics - metrics on garbage collection frequency and performance collected from parsing of the GC verbose log file when it is enabled for the application server JVM.
- Operating System metrics - metrics collected about the JVM process and the whole system execution, such as CPU used percentage, paging rate, and so on.

Additionally, the monitoring agent uses thresholds, called Application Health Indicators, to determine the quality of the application service. These thresholds are assigned automatically during baselining or you can manually customize them. There are three monitored application tiers evaluated for health status.
- Client Tier provides performance data and status of application execution in servlets/JSPs or portal containers as well as corresponding thread pools servicing these containers.
- Application Tier provides application execution metrics of EJB containers and custom requests.
- Backend Tier provides application execution in JDBC, JCA, JMS, and JNDI API calls.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:
- Situation Event console view, which shows the event console with activity associated with the Application Health Summary Navigator item and any other workspaces in the group, as well as linked workspaces. The Navigator displays an event icon overlaid on the Application Health Summary node when a situation becomes true. The report is useful when multiple alerts are raised as you can see them all in a single filtered view.

- Application Health Summary report, which shows the report of the application name, status, and health indicator for client, application, and backend tiers health status.

Accessing the Application Health Summary workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, or z/OS Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server entry of your choice.
5. Within the list of available J2EE application Server workspaces on that server, click the **Application Health** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Registry workspace

This workspace displays the information about the server configuration for the application.

This workspace displays data provided by the Application Monitoring Configuration attributes.

The predefined workspace contains:

- Situation Event Console report, which shows the event activity for situations associated with the current Navigator item. The Navigator alerts you when a situation becomes true by displaying an event indicator on the Navigator item. This report is useful when multiple alerts are raised and you might not know newly arrived alerts just by looking at the indicator.
- Application Configuration report, which shows the configurations that are discovered, stored and managed for J2EE applications running within that application server.

Accessing the Application Registry workspace

To access this workspace from the Application Health Summary workspace, use one of the following procedures:

- Within the Navigator, right-click the **Application Health** entry; and select **Workspace** -> **Application Registry**.
- From the primary Tivoli Enterprise Portal menu, select **View** -> **Workspace** -> **Application Registry**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

BEA WebLogic Application Server workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in KB) within the J2EE Application Server heap over time. The hover help associated with the chart displays the exact values.

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over time.

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time.

  This view displays data provided by the Request Times and Rates attributes .
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time.

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this BEA WebLogic Application Server, including JVM statistics and CPU usage statistics.

  This view displays data provided by the Application Server attributes .

Accessing BEA WebLogic Application Server Workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, select the **BEA WebLogic Application Server** entry.

See also:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Data sources workspace

This workspace displays statistical data for the data sources that your applications reference when accessing databases.

This workspace displays data provided by the Datasources attributes.

The predefined workspace contains the following items:

- Worst Data source Query Times bar chart, which shows the longest times (in milliseconds) the application spent waiting to retrieve data from the database during the specified interval
- Worst Data source Update Times bar chart, which shows the longest times (in milliseconds) the application spent updating data within the database during the specified interval

- Data sources - Current Interval report, which displays database usage information. For example, this report shows traffic information, such as the time the application spent trying to connect to the database and the total and average processing times for database queries and updates

Accessing the data sources workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server entry of your choice.
5. Within the list of available J2EE application server workspaces on that server, click the **Datasources** entry.

Selected data source - History workspace

This workspace displays the historical information that corresponds to the information in the data source workspace for a selected data source. Historical information is collected over a specific time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected data source - History workspace

To access this workspace from the data source workspace, use one of the following procedures:

- From the Data sources - Current Interval report, right-click, and from the menu, click **Selected Datasource - History**.
- From Worst Data source Query Times bar chart or the Worst Data source Update Times bar chart, right-click any bar; then, from the menu, select **Link To > Selected Datasource - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DB Connection Pools workspace

This workspace displays information about the database connection pools associated with SAP NetWeaver application server.

This workspace displays data provided by the “DB Connection Pools - NetWeaver attributes” on page 682.

The predefined workspace contains the following items:

- Pool Sizes bar chart, shows the current size of data source pool
- DB Connection Pools report, which displays information about the database connection pool for each defined data source, and an aggregated value that aggregates over all data sources. For example, this report displays the number of threads waiting for a connection and the number of connections created and released

Accessing the DB Connection Pools workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the SAP NetWeaver application server entry of your choice.
5. Within the list of available J2EE application server workspaces on the SAP NetWeaver server, select the **DB Connection Pools** entry.

Selected DB Connection Pool - History workspace

This workspace displays the historical information that corresponds to the information in the DB Connection Pools workspace for a selected connection pool. Historical information is collected over a particular measured time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

This workspace displays data provided by the "DB Connection Pools - NetWeaver attributes" on page 682.

The predefined workspace contains the following items:

- Active and Available Connections - History table, which displays the history of active and available connections
- Highest Wait Times - History table, which displays the history of the highest wait times (in milliseconds) for each database connection pool
- Selected DB Connection Pool - History report, which displays historical data and statistics in the DB connection pools for a selected connection pool

Accessing the Selected DB Connection Pool - History workspace

To access this workspace from the DB Connection Pools workspace, use one of the following procedures:

- From the DB Connection Pools report, right-click the link icon and select **Selected DB Connection Pool - History**.
- From Pool Size bar chart, right-click any bar, and select **Link To > Selected DB Connection Pool - History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DC Message Events workspace

This workspace displays the information about messages generated by the ITCAM for J2EE data collector and the event activity for situations associated with the Navigator item.

This workspace displays data provided by the "DC Messages - J2EE attributes" on page 683. This workspace is available only for J2SE servers.

The predefined workspace contains the following items:

- Situation Event Console report, which shows the event activity for situations associated with the current Navigator item. The Navigator alerts you when a situation becomes true by overlaying the Navigator item with an event indicator. This report is useful when multiple alerts are raised and you might not know newly arrived alerts just by looking at the indicator.
- DC Message Events report, which displays the messages generated by the ITCAM for J2EE data collector and the related information of the messages. This information includes message identifier, sequence number, and the component from which the messages were generated.

Accessing the DC Message Events workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2SE application server of your choice.
5. Within the list of available J2SE application server workspaces on that server, select the **DC Message Events** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces
- "Situations for ITCAM Agent for J2EE" on page 738

EJB Components workspace

This workspace displays runtime information for an EJB component in the Oracle/BEA WebLogic Server.

This workspace displays data provided by the "Enterprise Java Bean Components - WebLogic attributes" on page 684.

The predefined workspace contains the following items:

- Activated EJBs bar chart, which displays the number of activated EJBs
- EJB Components report, which displays runtime information for an EJB component

Accessing the EJB Components workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the Oracle/BEA WebLogic application server entry.

5. Within the list of available J2EE application server workspaces on the Oracle/BEA WebLogic server, select the **EJB Components** entry.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Enterprise Java Beans workspace

This workspace reports information about each Enterprise Java Bean (EJB) defined for an EJB module. The workspace provides information about these beans regarding their identity, instrumentation level settings, creation and destruction of bean objects, response times, invocations, calls, and rates for retrievals, returns, and discards.

For JBoss and Oracle application servers, this workspace displays data provided by the "Enterprise Java Bean Modules - J2EE attributes" on page 685.

The predefined workspace contains the following items:
- Message Driven Beans report, which displays performance statistics for message driven beans in the given EJB module
- Entity Beans report, which displays performance statistics for entity beans in the given EJB module
- Stateful Session Beans report, which displays performance statistics for stateful session beans in the given EJB module
- Stateless Session Beans report, which displays performance statistics for stateless session beans in the given EJB module

Accessing the Enterprise Java Beans - JBoss and Oracle application servers workspace

You access this workspace from the "EJB Modules workspace" on page 642. To list the EJBs, use one of the following procedures:
- From the primary Tivoli Enterprise Portal menu, select **View** ->**Workspace > Enterprise Java Beans**.
- In the Navigator, right-click the **EJB Modules** entry and select **Workspace> Enterprise Java Beans**.

To see the EJBs referenced by a specific EJB module, from the EJB modules report, right-click the link icon and select **Enterprise Java Beans**.

Enterprise Java Beans - NetWeaver workspace

In SAP NetWeaver server, this workspace displays data provided by the "Enterprise Java Bean Service - NetWeaver attributes" on page 687.

The predefined workspace contains the following items:
- Highest Creation Counts bar chart, which displays the highest count of times of a "create" method was invoked on the bean
- Most Active Session timeouts, which display the most timeout for the active sessions. If a session stays idle and not passivated for this long, it is removed
- Enterprise Java Beans report, which displays information about each Enterprise Java Bean (EJB)

Accessing the Enterprise Java Beans - NetWeaver workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the SAP NetWeaver application server entry of your choice.
5. Within list of available J2EE application server workspaces on the SAP NetWeaver server, click the **Enterprise Java Beans** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

EJB Modules workspace

This workspace displays aggregated information about each defined EJB. It displays aggregated bean performance data for all Enterprise beans deployed to an EJB module. It also displays aggregated information for the application server that aggregates bean performance data for all Enterprise beans deployed to the application server.

This workspace displays data provided by the "Enterprise Java Bean Modules - J2EE attributes" on page 685.

Note to OracleAS9 users: This workspace is not supported on OracleAS9 and no data is provided, hence this workspace is always empty.

The predefined workspace contains the following items:

- Highest Creation Count bar chart, which displays the largest number of times that beans were created during the interval
- EJB Modules report, which displays aggregated information for each defined EJB module that aggregates bean performance data for all Enterprise beans deployed to that module. The report also displays aggregated information for the application server that aggregates bean performance data for all Enterprise beans deployed to the application server

Accessing the EJB Modules workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server entry of your choice.
5. Within the list of available J2EE application server workspaces on that server, select **EJB Modules**.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

EJBs - Selected Enterprise Application workspace

This workspace displays resource statistics for selected EJB module. It displays performance statistics about individual EJBs deployed to the J2EE application.

This workspace displays data provided by the Enterprise Java Bean Components - WebLogic attributes.

The predefined workspace contains the following items:

- Highest Activation Rates bar chart, which displays the highest number of beans from this EJB Home that have been activated per second for the interval since the previous sample
- EJBs - Selected Enterprise Application report, which displays resource statistics for selected EJB module.

Accessing the EJBs - Selected Enterprise Application workspace

This workspace is accessed from the EJB Components workspace. To list the EJBs, use one of the following procedures:

- From the primary Tivoli Enterprise Portal menu, open the **View** menu, then click **Workspace > EJBs - Selected Enterprise Application**.
- Within the Navigator, right-click the **EJB Components** entry and select **Workspace> EJBs - Selected Enterprise Application**.
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collection Analysis workspace

This workspace summarizes all the Java Virtual Machine garbage-collector activity over a user-defined interval. The JVM generates detailed garbage collection logs for an application server when started with the verbose:gc runtime parameter.

This workspace displays data provided by the Garbage Collection Analysis attributes.

The predefined workspace contains the following items:

- Garbage Collection Rate - History graph, which displays the rate at which the garbage-collection algorithm is being invoked
- Heap Usage - History bar chart, which displays the high water mark of free storage (in kilobytes) available in the heap after each garbage-collector run
- Percentage of Time Garbage Collector Running - History graph, which displays the percentage of real time the garbage collector was running during the current interval, for each server region
- Garbage Collection Analysis report, which displays information about the garbage-collection activities within the Java Virtual Machine that is hosting the application server. For example, this report displays the number of times the collector ran during the interval and the resulting number of objects that the collector freed

Accessing the Garbage Collection Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.
5. Within the list of available J2EE application server workspaces on that server, click the **Garbage Collection Analysis** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collections - Selected Allocation Failure workspace

This workspace provides detailed information about the garbage-collection cycles that occurred in response to a specific heap-allocation failure that occurred within the Java Virtual Machine.

This workspace displays data provided by the Garbage Collection Cycle attributes .

Note to Solaris and HP-UX users: Allocation-failure information is not recorded on these platforms; hence this workspace is always empty.

The predefined workspace contains the following items:

- GC Elapsed Times bar chart, which breaks down the mark, sweep, and compact times (in milliseconds) for each garbage-collection cycle that occurred for the selected allocation failure
- Heap Usage bar chart, which displays the JVM heap usage (kilobytes in use, freed, and free at start of garbage collection) for each garbage-collection cycle
- Garbage Collections - Selected Allocation Failure report, which displays information about a single garbage-collection cycle that the Java Virtual Machine hosting the application server performed. For example, this report displays the free heap space both before and after garbage collection, the heap space freed, and the number of objects moved during garbage collection

Accessing the Garbage Collections - Selected Allocation Failure workspace

To access this workspace from the "Allocation Failures workspace" on page 634, use one of the following procedures:

- From the Allocation Failures report, right-click the link icon, and select **Garbage Collections - Selected Allocation Failure**.
- From the Allocation Failure Elapsed Times bar chart or the Heap Usage - History bar chart, right-click any bar and select **Link To -> Garbage Collections - Selected Allocation Failure**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

J2EE Agent workspace

This workspace displays product events that affect the ability of the J2EE Application Server agent to collect data. This workspace displays events occurring within the J2EE Application Server agent and J2EE application servers that are installed on the host computer. It also displays the status of the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- J2EE Agent Events report, which displays information about agent-level events that affect the ability of the Tivoli Enterprise Monitoring Agent to collect data for the J2EE application server. You can use this view to see exception and error messages, their IDs, and their severity.

  Agent Events report also shows the result of issuing a Take Action command. Place your cursor over a truncated message to display the text of the complete message.

  This report displays data reported by the "J2EE Agent Events attributes" on page 696.
- Application Servers Summary report displays information about status of the J2EE server.

  This report displays data reported by the Application Server Status attributes .

Accessing the J2EE Agent workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of available Tivoli Enterprise Monitoring Agents on that node, click the **J2EE Agent** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces
- "Situations for ITCAM Agent for J2EE" on page 738

J2SE Application workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the J2SE Application Server heap over time. The hover help associated with the chart displays the exact values

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes .

- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this J2SE Application Server, including JVM statistics and CPU usage statistics

  This view displays data provided by the Application Server attributes .

Accessing J2SE Application Workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications on that node, expand the list of J2EE agents.
4. Within the list of available agents, click the **J2SE Application** entry of your choice.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JBoss App Server workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the J2EE Application Server heap over time. The hover help associated with the chart displays the exact values

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over a time period

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over a time period

  This view displays data provided by the Request Times and Rates attributes .
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over a time period

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this JBoss application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the Application Server attributes .

Accessing the JBoss App Server workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.

3. Within the list of monitored applications on that node, expand the list of J2EE agents.
4. Within the list of available agents, select the **JBoss App Server** entry of your choice.

For additional information, see:

- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

JCA Connection Pools workspace

This workspace displays information about resource adapters and connectors that adhere to JCA, the J2EE Connector Architecture (JCA).

In JBoss Application Server, this workspace displays data provided by the "JCA Connection Pools - J2EE attributes" on page 699.

The predefined workspace contains the following items:

- Highest Pool Usage bar chart, which shows the largest pool usage for each JCA connection pool. The Y-axis headings correspond to the row number of the JCA Connection Pools report
- Worst Wait Times bar chart, which shows the worst wait time (in milliseconds) for each of the JCA connection pools. The Y-axis headings correspond to the row number of the JCA Connection Pools report
- Worst Use Times bar chart, which shows the worst use time (in milliseconds) for each of the JCA connection pools. The Y-axis headings correspond to the row number of the JCA Connection Pools report
- JCA Connection Pools report, which displays information about the JCA connection pool for each Connection Factory

In the Oracle/BEA WebLogic Application Server, this workspace displays data provided by the "J2EE Connector Connection Pools - WebLogic attributes" on page 697.

The predefined workspace contains the following items:

- Highest Active Connections bar chart, which shows the current highest active connections.
- Worst Connection Rejection Rates, which shows the worst connection rejection rates (in milliseconds) for each JCA connection pools.
- JCA Connection Pools report, which displays information about the JCA connection pool for each Connection Factory

Accessing the JCA Connection Pools workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications on that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.

5. Within the list of available J2EE application server workspaces in that server, select the **JCA Connection Pools** entry.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JDBC Connection Pools workspace

This workspace provides usage information about the JDBC connection pools for a database in the Oracle/BEA WebLogic Server.

This workspace displays data provided by the "JDBC Connection Pools - WebLogic attributes" on page 701.

The predefined workspaces contains the following items:
- Highest Active Connections bar chart, which shows the current highest active connections
- Worst Wait Times bar chart, which shows the worst wait times (in milliseconds) for each database connection pool
- DB Connection Pools report, which displays information about the database connection pool for each defined data source, and an aggregated value that aggregates over all data sources. For example, this report displays the number of threads waiting for a connection and the number of connections created and released

Accessing the JDBC Connection Pools workspace

To access this workspace, complete the following steps:
1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the Oracle/BEA WebLogic application server entry.
5. Within the list of available J2EE application server workspaces on the Oracle/BEA WebLogic server, click the **JDBC Connection Pools** entry.

Selected JDBC Connection Pool - History workspace

This workspace displays the historical information that corresponds to the information in the JDBC Connection Pools workspace for a selected connection pool. Historical information is collected over a particular measured time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

The predefined workspace contains the following items:
- Active and Available Connections - History table, which displays the history of active and available connections.
- Highest Wait Times - History table, which displays the history of the highest wait times (in milliseconds) for each database connection pool.
- Selected JDBC Connection Pool - History report, which displays historical data and statistics in the JDBC connection pools for a selected connection pool.

Accessing the Selected JDBC Connection Pool - History workspace

To access this workspace from the DC Connection Pools workspace, use one of the following procedures:

- From the JDBC Connection Pools report, right-click the link icon, and from the menu, click **Selected JDBC Connection Pool - History**.
- From the Highest Active Connection bar chart, the Worst Wait Times bar chart, right-click any bar, and select **Link To -> Selected JDBC Connection Pool - History**.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JMS Sessions workspace

This workspace displays statistics for Java Message Service (JMS) sessions in Oracle/BEA WebLogic Server.

In Oracle/BEA WebLogic Server, the workspace displays data provided by the "JMS Sessions - WebLogic attributes" on page 709.

The predefined workspace contains the following items:

- Most Messages Sent bar chart, which displays the largest number of messages sent
- Most Messages Received bar chart, which displays the largest number of messages received
- JMS Sessions report, which displays statistics for JMS sessions
- Message Producers report, which displays information about message producers
- Message Consumers report, which displays information about message consumers

Accessing the JMS Sessions workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the Oracle/BEA WebLogic application server.
5. Within the list of available J2EE application workspaces on that server, click the **JMS Sessions** entry.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JMS Summary workspace

The JMS Summary workspace displays information about queues being used by your applications via the JMS interface and about how J2EE Application Server

applications are using J2EE MQ. It displays such information as the number of messages read and written and which queue managers and queues were used during the interval.

This workspace displays data provided by the JMS Summary attributes .

The predefined workspace contains the following items:

- Worst JMS Send Times bar chart, which displays the longest times (in milliseconds) that your application spent putting messages into a queue during the interval
- Worst JMS Receive Times bar chart, which displays the longest times (in milliseconds) that your application spent getting messages from a queue during the interval
- Worst JMS Browse Times bar chart, which displays the longest times (in milliseconds) that your application spent browsing messages on a queue during the interval
- JMS Summary - Current Interval report, which displays detailed information about the send, receive, browse, and publish times for the messaging middleware (J2EE MQ) by your J2EE Application Server applications using JMS. It includes about the queue managers and queues that are being used and how many messages are being read and written

Accessing the JMS Summary workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.
5. Within the list of available J2EE application server workspaces on that server, click the **JMS Summary** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JTA Resources workspace

This workspace displays information about the Java Transaction API (JTA) resources.

This workspace displays data provided by the "JTA Resources - J2EE attributes" on page 716 for JBoss and Oracle application servers. For WebLogic application server, this workspace displays data provided by "Java Transaction Service - WebLogic attributes" on page 713.

The predefined workspace contains the following items:

- Active Transactions bar chart, which shows the number of active transactions
- JTA Resources report, which displays information about the JTA resources

Accessing the JTA Resources workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.
5. Within the list of available J2EE application server workspaces on that server, click the **JTA Resources** entry.

For additional information, see:

- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

JTA Summary workspace

This workspace displays the performance summary statistics information about transactions in SAP NetWeaver application server.

This workspace displays data provided by "JTA Summary - NetWeaver attributes" on page 717.

The predefined workspace contains the following items:

- Transactions - History graph, which shows the history of transactions
- Transaction Service report, which displays performance data of the Transaction Service

Accessing the JTA Summary workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the SAP NetWeaver application server entry of your choice.
5. Within the list of available J2EE application server workspaces on the SAP NetWeaver server, click the **JTA Summary** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JVM Statistics workspace

This workspace reports the detailed statistics of the operating system on which the J2SE application server is running and the Java Virtual Machine (JVM) information.

This workspace displays data provided by "JDK - Operation System attributes" on page 703,"JDK - Memory attributes" on page 705"JDK - JVM attributes" on page 706 and "JDK - Threading attributes" on page 708.

The predefined workspace contains the following items:

- Memory Usage - History graph, which displays the amount of used heap memory and non-heap memory (in kilobytes) in the JVM over time
- Pending Objects - History graph, which displays the number of objects that are not finalized
- Heap Sizes - History graph, which displays the maximum heap memory size and non-heap memory size (in kilobytes) in the JVM over time
- Threads - History graph, which displays the information about threads, such as peak threads and daemon threads
- JVM Statistics report, which shows the overall JVM information that the current J2SE application server uses, such as JVM name, version, and vendor
- Operating System report, which shows the overall information about the operating system on which the J2SE application server is running

Accessing the JVM Statistics workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2SE Application entry of your choice.
5. Within the list of available workspaces on the J2SE Application, click the **JVM Statistics** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Log Analysis workspace

This workspace reports application server error and exception conditions as recorded in the application server log file.

This workspace displays data provided by the Log Analysis attributes .

The predefined workspace contains the following items:

- DC Message Events report, which displays information about the data collector Messages

  This report displays data reported by the "DC Messages - J2EE attributes" on page 683.
- Log Analysis report, which displays application server error and exception conditions as recorded in the application server log file. This information includes the exception severity as well as the ID and text of the associated message

Accessing the Log Analysis workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.
5. Within the list of available J2EE application server workspaces in that server, click the **Log Analysis** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces
- "Situations for ITCAM Agent for J2EE" on page 738

Oracle App Server workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilobytes) within the J2EE Application Server heap over time. The hover help associated with the chart displays the exact values

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes .
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this Oracle Application Server, including JVM statistics and CPU usage statistics

  This view displays data provided by the Application Server attributes .

Accessing Oracle App Server Workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, click the **Oracle App Server** entry of your choice.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Request Analysis workspace

The workspace reports response times and functional decomposition information about requests (including servlets, JSPs, and EJB methods) that completed during the interval. A historical version of this workspace provides a long-term view of a single request that you select.

This workspace displays data provided by the Request Analysis attributes .

The predefined workspace contains the following items:
- Worst Response Times bar chart, which displays the five worst response times for requests processed during the current interval
- Worst Completion Rates bar chart, which displays the 10 requests that have the worst completion rates
- Requests - Current Interval report, which displays detailed information about the response times recorded for each request

Accessing the Request Analysis workspace

To access this workspace, complete the following steps:
1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.
5. Within the list of available J2EE application server workspaces on that server, select the **Request Analysis** entry.

Selected Request - History workspace

The Selected Request - History workspace displays the historical information that corresponds to the information in the Request Analysis workspace for a single request type that you select. Historical information is collected over a specific time span. See the online help for Tivoli Enterprise Portal for a detailed explanation of historical reporting.

Accessing the Selected Request - History workspace

To access this workspace from the Request Analysis workspace, use one of the following procedures:
- From the Requests - Current Interval report, right-click the link icon, then from the menu, click **Selected Request - History**.
- From the Worst Response Times bar chart, right-click any bar, then from the menu, click **Link To > Selected Request - History**.

For additional information, see:
- Organization of the predefined workspaces

- Attribute groups used by the predefined workspaces

Selected Request - Baseline workspace

This workspace displays aggregated information about the request baseline. The baselining collects statistical information about an application requests completion times and uses this information to assign fair and bad thresholds on the application requests. The product divides the whole request response times into buckets and collects individual hits into each bucket. Use these attributes to get statistics from individual requests collected during baselining interval.

This workspace displays data provided by the Baseline attributes.

The predefined workspace contains:

- Baseline Data report, which shows lower and upper boundaries for each bucket request as well as the breakdown of nested request types in percentage.
- Request Label report, which shows the monitoring configuration settings for selected requests, including auto-threshold settings and actual thresholds calculated from the baseline data.
- Nested Delays Distribution bar chart, which displays a bar for each bucket of response times across the different nested types (JDBC, JCA, JMS, and so on.). This chart provides you with additional hints and insight about how to interpret response times distribution displayed in the distribution chart.
- Response Time Distribution bar chart, which displays the distribution of the servlet response times on the baselining interval, also called zones.

Accessing the Selected Request - Baseline workspace

To access this workspace from the Application Registry workspace, use the following procedure:

- From the Application Configurations report, right-click the link icon to the left of any row and select **Selected Request - Baseline**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Application Tier Analysis workspace

The workspace displays detailed information about application tier health for a selected J2EE application.

The application tier health is derived from the following performance statistics:

- Calculated application request delays in EJB container or custom requests delays compared against corresponding thresholds assigned in application configuration.
- Completion rates for application edge EJB requests.
- Application server ORB thread pool utilization level.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Application Tier Analysis report, which shows the overall health status of the Application tier for a selected application.

- Worst Application Tier Delays - Top 10 bar chart, which displays the top 10 delayed requests in the application tier. This view displays data provided by the Request Analysis attributes.
- Worst Application Tier Completion Rates - Top 10 bar chart, which displays the top 10 worst requests in the application tier. This view displays data provided by the Request Analysis attributes.
- JVM Health - CPU Used % graph, which displays the percentage of the CPU used by the Java Virtual Machine (JVM) during the interval. This view displays data provided by the Application Server attributes.
- JVM Health - Heap Used % graph, which displays the current heap usage for the monitored JVM. This view displays data provided by the Garbage Collection Analysisattributes.
- JVM Health - GC Time % graph, which displays the percentage of real time that the garbage collector was active during the interval. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Selected Application - Application Tier Analysis workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon and select **Selected Application - Application Tier Analysis**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Configuration workspace

This workspace displays the information about the configuration details of the selected application. The workspace contains information about application requests and the corresponding thresholds assigned to them, and information about status and configuration for application baseline activity. An entry is created for each application in the configuration report when a J2EE application is discovered by the monitoring agent. The data is also stored in a context file local to the monitoring agent where it can persist between monitoring agent restarts.

This workspace displays data provided by the Application Monitoring Configuration attributes.

The predefined workspace contains:

- Longest Request Thresholds - Top 10 bar chart, which displays the 10 longest (in time) request thresholds configured for the given application (Servlet/JSP URL or EJB class/method call).
- Application Requests report, which shows the discovered application requests and thresholds assigned to them.
- Application report, which shows the common details about application configuration, including custom requests monitoring levels for application and current baseline status.

Accessing the Selected Application - Configuration workspace

Use the following steps to access this workspace.

1. Click **Application Health Summary > Application Health Summary Report**.
2. Right-click the link icon and select **Selected Application - Configuration**

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Backend Tier Analysis workspace

This workspace displays information about the details of the Backend tier for a selected application.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Backend Tier Analysis report, which shows the overall health status of the backend tier for a selected application.
- Worst Backend Tier Delays - Top 10 bar chart, which displays the top 10 delayed requests in the backend tier. This view displays data provided by the Request Analysis attributes.
- Most Used Data sources - Top 10 bar chart, which displays the average time per request used by queries and updates to the data source. This view displays data provided by the Datasources attributes.
- Most Used JMS Resources - Top 10 bar chart, which displays the longest times your application spent in getting messages from a queue, putting messages onto a queue, publishing messages to a queue, or browsing messages on a queue during the interval. This view displays data provided by the JMS Summary attributes.
- JVM Health - CPU Used % graph, which displays the percentage of the CPU used by the Java Virtual Machine (JVM) during the interval. This view displays data provided by the Application Server attributes.
- JVM Health - Heap Used % graph, which displays the current heap usage for the monitored JVM. This view displays data provided by the Garbage Collection Analysis attributes.
- JVM Health - GC Time % graph, which displays the percentage of real time that the garbage collector was active during the interval. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Selected Application - Backend Tier Analysis workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon and, from the menu, click **Selected Application - Backend Tier Analysis**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Health History workspace

The workspace displays the information about the historical health status of a selected application. By default, the history data is collected for the last 24 hours.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Availability/Throughput - History graph, which displays average request processing rate by application over the time. This view displays data provided by the Request Times and Rates attributes.
- Availability/Completion Rate - History graph, which displays the average request completion rate by application over the time. Completion rate is defined as the ratio of successfully completed requests count to the total count of requests processed by application on the interval. This view displays data provided by the Request Times and Rates attributes.
- Availability/Average Load- History graph, which displays the average number of concurrent application requests over the time. This view displays data provided by the Request Times and Rates attributes.
- Response Time - History graph, which displays the average application response time over the time. This view displays data provided by the Request Times and Rates attributes.
- Server Resources/CPU Used - History graph, which displays the percent of CPU time used by the application JVM process over the time. This view displays data provided by the Application Server attributes.
- Server Resources/Paging Rate - History graph, which displays the system paging rate in kilobytes per second over the time. This view displays data provided by the Application Server attributes.
- Server Resources/GC Active Time - History graph, which displays the percentage of total CPU time for which the garbage collector was active over the time. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Selected Application - Health History workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon and select **Selected Application - Health History**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Application - Client Tier Analysis workspace

This workspace displays detailed information about the client tier health for a selected J2EE application.

The client tier health indicator is derived from the following performance statistics:

- Calculated application request delays inside Servlet/JSP or Portal container compared against corresponding thresholds assigned in application configuration.
- Completion rates for edge Servlet/JSP and Portal application requests.

This workspace displays data provided by the Application Health Status attributes.

The predefined workspace contains:

- Client Tier Analysis report, which shows the overall health status of application execution in web or portal containers.
- Worst Client Tier Delays - Top 10 bar chart, which displays the top 10 requests with biggest delays (threshold violations) in the client tier. This view displays data provided by the Request Analysis attributes.
- Worst Client Tier Completion Rates - Top 10 bar chart, which displays the top 10 Servlet/JSP/Portal edge requests with the worst completion rates. This view displays data provided by the Request Analysis attributes.
- JVM Health - CPU Used % graph, which displays the percentage of the CPU used by the Java Virtual Machine (JVM) during the interval. This view displays data provided by the Application Server attributes.
- JVM Health - Heap Used % graph, which displays the current heap usage for the monitored JVM. This view displays data provided by the Garbage Collection Analysis attributes.
- JVM Health - GC Time % graph, which displays the percentage of real time that the garbage collector was active during the interval. This view displays data provided by the Garbage Collection Analysis attributes.

Accessing the Selected Application - Client Tier Analysis workspace

To access this workspace from the Application Health Summary workspace, use the following procedures:

- From the Application Health Summary report, right-click the link icon to the left of any row and select **Selected Application - Client Tier Analysis**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - data sources workspace

The Selected Request - data sources workspace displays information about JDBC activity performed by the request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes .

The predefined workspace contains the following items:

- Worst Data Sources Response Times bar chart, which shows the worst response times (in milliseconds) for data sources accessed by this request
- Selected Request - Data sources report, which displays detailed information about the data sources accessed for the selected request

Accessing the Selected Request - data sources workspace

To access this workspace from the “Request Analysis workspace” on page 654, use one of the following procedures:

- From the Requests - Current Interval report, right-click the link icon, and select **Selected Request - Datasources**.

- From Worst Average Response Times bar chart, right-click any bar and select **Link To > Selected Request - Datasources**.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - JMS Queues workspace

The Selected Request - JMS Queues workspace displays information about message queues owned by messaging middleware and accessed by the request that you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes .

The predefined workspace contains the following items:
- Worst JMS Queues Response Times bar chart, which shows the worst response times (in milliseconds) for JMS resources accessed by this request
- Selected Request - JMS Queues report, which displays detailed information about the JMS resources accessed by the selected request

Accessing the Selected Request - JMS Queues workspace

To access this workspace from the Request Analysis workspace , use one of the following procedures:
- From the Requests - Current Interval report, right-click the link icon, and select **Selected Request - JMS Queues**.
- From Worst Average Response Times bar chart, right-click any bar and select **Link To > Selected Request - JMS Queues**.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - Resource Adapters workspace

The Selected Request - Resource Adapters workspace displays response-time information about the JCA resources adapters referenced by the request you selected in the primary Request Analysis workspace.

This workspace displays data provided by the Selected Request attributes .

The predefined workspace contains the following items:
- Worst Resource Adapter Response Times bar chart, which shows the worst-performing JCA resource adapter nested requests, in milliseconds
- Selected Request - Resource Adapters report, which displays detailed information about each JCA resource adapter that was accessed by the selected request

Accessing the Selected Request - Resource Adapter

To access this workspace from the "Request Analysis workspace" on page 654, use one of the following procedures:
- From the Requests - Current Interval report, right-click the link icon, and select **Selected Request - Resource Adapters**.

- From Worst Average Response Times bar chart, right-click any bar and select **Link To > Selected Request - Resource Adapters**.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

SAP NetWeaver Server workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilobytes) within the J2EE Application Server heap over time. The hover help associated with the chart displays the exact values

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes .
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this SAP NetWeaver Server, including JVM statistics and CPU usage statistics

  This view displays data provided by the Application Server attributes .

Accessing SAP NetWeaver Server workspace

To access this workspace, complete the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, click the **SAP NetWeaver Server** entry of your choice.

For additional information, see:
- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

Servlets/JSPs - Selected Enterprise Application workspace

This workspace displays performance statistics for Servlets/JSPs running in the given Enterprise Application.

This workspace displays data provided by the "Servlets and JSPs - WebLogic attributes" on page 733.

The predefined workspace contains the following items:

- Worst Execution Times bar chart, which displays the worst amount of time the invocations of the servlet have executed for the interval since the previous sample
- Servlets/JSPs - Selected Enterprise Application report, which displays performance information for servlets and JavaServer pages

Accessing the Servlets/JSPs - Selected Enterprise Application workspace

To access this workspace from the"Web Applications workspace" on page 663 in BEA WebLogic application server, use one of the following procedures:

- From the web Applications report, right-click the link icon, and select **Servlets/JSPs - Selected Enterprise Application**.
- From the Worst Response Times bar chart, the Most Popular web Applications bar chart, the Worst Error Rates bar chart, or the Worst Execution Time bar chart, right-click any bar and select **Link To > Servlets/JSPs - Selected Enterprise Application**.

For additional information, see:

- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

Servlets/JSPs - Selected Web Application workspace

This workspace displays statistical data regarding the servlets and JSPs invoked by a single web application.

This workspace displays data provided by the "Servlets JSPs - J2EE attributes" on page 731.

The predefined workspace contains the following items:

- Worst Response Times bar chart, which displays the worst average response times (in milliseconds) for servlets invoked by the selected web application
- Servlets/JSPs - Selected Web Application report, which displays performance information about the servlets and JSPs invoked by the application. For example, this report displays the average number of concurrent requests for a servlet and the time it takes a servlet to respond to a request

Accessing the Servlets/JSPs - Selected Web Application workspace

To access this workspace from the "Web Applications workspace" on page 663, use one of the following procedures:

- From the Web Applications report, right-click the link icon, and select **Servlets/JSPs - Selected Web Application**.
- From the Worst Response Times bar chart, the Most Popular Web Applications bar chart, the Worst Error Rates bar chart, or the Worst Execution Time bar chart, right-click any bar and select **Link To -> Servlets/JSPs - Selected Web Application**.

For additional information, see:

- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

Tomcat Server workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the J2EE Application Server heap over time. The hover help associated with the chart displays the exact values

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes .
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this Tomcat server, including JVM statistics and CPU usage statistics

  This view displays data provided by the Application Server attributes .

Accessing Tomcat Server Workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, click the **Tomcat Server** entry of your choice.

For additional information, see:

- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

Web Applications workspace

This workspace displays information about the web applications running in J2EE application servers.

Note to OracleAS9 users: This workspace is not supported on OracleAS9 and no data is provided, hence this workspace is always empty.

This workspace displays data provided by the "Web Applications - J2EE attributes" on page 735.

The predefined workspace contains the following items:

- Worse Response Times bar chart, which shows the worst servlet response times (in milliseconds) during the interval
- Most Popular Web Applications bar chart, which shows the servlet exception and request rates (in events per second)

- Web Applications report, which displays aggregated performance data for each web application about all servlets and JSPs deployed to that web application, including response and error rates and response times

In the BEA WebLogic Application Server, this workspace displays data provided by the "Web Applications - WebLogic attributes" on page 737.

The predefined workspace contains the following items:

- Most Active Sessions bar chart, which shows the high water mark of the total number of open sessions in this server
- Most Popular Web Applications bar chart, which shows the servlet exception and request rates (in events per second)
- Web Applications report, which displays aggregated performance data for each web application about all servlets and JSPs deployed to that web application, including response and error rates and response times

Accessing the Web Applications workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the J2EE application server of your choice.
5. Within the list of available J2EE application server workspaces on that server, click the **Web Applications** entry.

For additional information, see:

- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

Web Container workspace

This workspace displays summary statistics about servlets/JSPs invocations in SAP NetWeaver Web Container.

This workspace displays data provided by the "Web Container - NetWeaver attributes" on page 734.

The predefined workspace contains the following items:

- Security Sessions bar chart, which displays the number of current valid, invalid, and also timed out security sessions
- Http Sessions bar chart, which displays the number of current valid, invalid and also timed out http sessions
- Servlets/JSPs report, which displays performance information about the servlets and JSPs invoked by the application

Accessing the Web Container workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, expand the SAP NetWeaver application server.
5. Within the list of available J2EE application server workspaces on the SAP NetWeaver server, click the **Web Container** entry.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

WebSphere App Server CE workspace

This workspace displays overall statistics for each application server being monitored by the Tivoli Enterprise Monitoring Agent.

The predefined workspace contains the following items:

- Heap Usage - History bar chart, which displays free memory size and used memory size (in kilo bytes) within the J2EE Application Server heap over time. The hover help associated with the chart displays the exact values

  This view displays data provided by the Garbage Collection Analysis attributes .
- Response Time - History graph, which shows the server response time to requests over time

  This view displays data provided by the Request Times and Rates attributes .
- Request Rate - History graph, which shows the rate at which requests have been received by this server over time

  This view displays data provided by the Request Times and Rates attributes .
- Percent CPU Used - History graph, which shows the percentage of the CPU that this server consumed over time

  This view displays data provided by the Application Server attributes .
- Application Server Summary report, which displays overall information about this WebSphere App Server CE application server, including JVM statistics and CPU usage statistics

  This view displays data provided by the Application Server attributes .

Accessing WebSphere App Server CE workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications associated with that node, expand the list of J2EE agents.
4. Within the list of available agents, click the **WebSphere App Server CE** entry of your choice.

For additional information, see:
- "Organization of the predefined workspaces" on page 625
- "Attribute groups used by the predefined workspaces" on page 666

Attributes for ITCAM Agent for J2EE

IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE is a Tivoli Enterprise Management Agent that is located within your distributed system. This agent gathers data about running J2EE Application Server processes that have been collected and stored by the ITCAM for J2EE data collector, and stores this data in elements called attributes. Each attribute is a characteristic of an object. For example, the Receive Count attribute in the JMS Summary attribute group counts the number of messages your applications have retrieved from JMS messages queues.

Attribute groups

The ITCAM Agent for J2EE attributes are organized into groups of related items. These attribute groups comprise the attribute tables for this agent. For example, the Garbage Collection Analysis attribute group provides information about the frequency with which the Java Virtual Machine (JVM) starts its garbage collector.

Attributes and workspaces

Various attributes are referenced by the predefined situations of the product. You can also use the IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE attributes to create your own situations to monitor the performance of your J2EE application servers and their applications. These situations can monitor your J2EE Application Server resources or correlate multiple conditions to alert you to problems that might have occurred when attribute values exceed defined thresholds.

Attribute groups used by the predefined workspaces

A workspace contains graphical data or report columns that correspond directly to particular attributes in an attribute group. The table shows the correlations between the predefined workspaces and the attribute groups. The primary and secondary workspaces are listed alphabetically, not in the order in which they display in the Navigator.

Table 93. Workspaces and the attribute groups they reference

Workspace	Related Attribute Groups
Application Health Summary	Application Health Status
Application Registry	Application Monitoring Configuration
Allocation Failures	Allocation Failure - J2EE
BEA WebLogic App Server	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE
Datasources Selected Datasource - History	Datasources - J2EE
DB Connection Pools Selected DB Connection Pool - History	DB Connection Pools - NetWeaver
EJB Components	Enterprise Java Bean Components - WebLogic
EJB Modules	Enterprise Java Bean Modules - J2EE
EJBs - Selected Enterprise Application	Enterprise Java Bean - WebLogic
Enterprise Java Beans	Enterprise Java Bean Service - NetWeaver Enterprise Java Bean Modules - J2EE

Table 93. Workspaces and the attribute groups they reference (continued)

Workspace	Related Attribute Groups
Garbage Collection Analysis	Garbage Collection Analysis - J2EE
Garbage Collections - Selected Allocation Failure	Garbage Collection Cycle - J2EE
J2EE Agent	J2EE Agent Events Application Server Status - J2EE
J2SE Application	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE
JBoss App Server	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE
JCA Connection Pools	JCA Connection Pools - J2EE J2EE Connector Connection Pools - WebLogic
JDBC Connection Pools Selected JDBC Connection Pool - History	JDBC Connection Pools - WebLogic
JMS Session	JMS Sessions - WebLogic
JMS Summary	JMS Summary
JTA Resources	JTA Resources - J2EE Java Transaction Service - WebLogic
JTA Summary	JTA Summary - NetWeaver
JVM Statistics	JDK - Operating System JDK - Memory JDK - JVM JDK - Threading
Log Analysis DC Message Events	Log Analysis - J2EE DC Messages - J2EE
Oracle App Server	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE
Request Analysis Selected Request - History	Request Analysis - J2EE
Selected Request - Baseline	Baseline attributes
Selected Application - Application Tier Analysis Selected Application - Backend Tier Analysis Selected Application - Health History Selected Application - Client Tier Analysis	Application Health Status
Selected Application - Configuration	Application Monitoring Configuration
Selected Request - Datasources	Selected Request - J2EE
Selected Request - JMS Queues	Selected Request - J2EE
SAP NetWeaver Server workspace	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE
Selected Request - Resource Adapters	Selected Request - J2EE
Servlets/JSPs - Selected Enterprise Application	Servlets and JSPs - WebLogic
Servlets/JSPs - Selected Web Application	Servlets JSPs - J2EE
Tomcat Server	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE

Table 93. Workspaces and the attribute groups they reference (continued)

Workspace	Related Attribute Groups
Web Applications	Web Applications - J2EE Web Applications - WebLogic
Web Containers	Web Container - NetWeaver
WebSphere App Server CE	Application Server - J2EE Request Times and Rates - J2EE Garbage Collection Analysis - J2EE

Allocation Failure - J2EE attributes

The **Allocation Failure - J2EE** attribute group provides information about the heap-allocation failure that caused the Java Virtual Machine hosting the application server to start its garbage-collection routine. Use the Allocation Failure attributes to determine the events that caused the JVM to start garbage collection.

The attributes within this group are used to build the "Allocation Failures workspace" on page 634.

Allocation Failure Number The identifier assigned to the current allocation-failure block, which is associated with a bar in the Heap Usage - History bar chart (supported IBM Developer Kit for Java). The valid format is a positive integer.

Bytes Needed The number of bytes needed on the heap when this allocation failure occurred. The valid format is a positive integer.

GC Cycle Count The number of Garbage Collection cycles ran for this allocation. The valid format is a positive integer.

Heap Expanded The total number of kilobytes by which the heap expanded or contracted as a result of garbage collection. The valid format is a positive integer.

Heap Free (%) after GC The percentage of heap that is free after allocation failure. The valid format is a positive integer.

Heap Status Indicates whether the out-of-heap-space alert has been raised. Valid values are Normal, Out_of_heap_space, Heap_space_is_low, and Insufficient_space.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Kbytes Free at Start of GC The number of kilobytes that were available in the heap before garbage collection began in response to this allocation failure. The valid format is a positive integer.

Kbytes Freed by GC The number of kilobytes freed for this allocation failure. The valid format is a positive integer.

Kbytes Used The number of kilobytes in the heap that were in use when this allocation failure occurred. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum length of 256 characters.

Objects Moved The total objects moved during compaction available. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process ID of the Java Virtual Machine (JVM). The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 94. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data were collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Time since Last Failure (ms) The time elapsed since the last allocation failure. The valid format is a positive integer.

Time to Complete (ms) The time (in milliseconds) taken to complete the action that resulted from this allocation failure. The valid format is a positive integer.

Total Kbytes Freed by GC The total number of kilobytes freed by the garbage collector in response to this allocation failure. The valid format is a positive integer.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Health Status attributes

The **Application Health Status** attributes provide information for real-time and historical application health status.

The attributes within this group are used to build the Application Health Summary workspace.

Application Health The combined application health level. Valid values are Unknown, Good, Fair, and Bad.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Application Status The status of the monitored application (only supported for Tomcat, and Oracle). Valid values are Standby, Discovered, Unknown, Starting, Running, Stopping, Stopped, and Failed.

Application Tier Health The health level of the application tier. Valid values are Unknown, Good, Fair, and Bad. Application tier health indicator is determined from EJB or custom request delays collected on the interval and compared against thresholds configured for application requests.

Backend Tier Health The health level of the backend tier. Backend tier health indicator is determined from JDBC, JCA, JNDI, JMS delays collected on the interval and compared against thresholds configured for application requests. Valid values are Unknown, Good, Fair, and Bad.

Client Tier Health The health level of the client tier. Valid values are Unknown, Good, Fair, and Bad. Client tier health indicator is determined from servlet/JSP or portal delays collected on the interval and compared against thresholds configured for application requests.

Completion Level The completion level of the requests during the interval. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from request data as the percentage of number of failed requests to the total number of application requests on the interval.

Custom Requests The availability indicator of custom requests. Valid values are Unknown, Good, Fair, and Bad.

EJB Container The health level of the EJB container. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from EJB delay types collected during the interval and compared against application thresholds.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA The overall health status of J2EE Connector Architecture (JCA) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JCA delay types collected during the interval and compared against application thresholds.

JDBC The overall health status of Java DataBase Connectivity (JDBC) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JDBC delay types collected during the interval and compared against application thresholds.

JNDI The overall health status of Java Naming and Directory Interface (JNDI) resources used by applications. Valid values are Unknown, Good, Fair, and Bad.

This attribute is determined from JNDI delay types collected during the interval and compared against application thresholds.

JMS The overall health status of Java Message Service (JMS) resources used by applications. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from JMS delay types collected during the interval and compared against application thresholds.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Portal Container The health level of the portal container. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from portal delay types collected during the interval and compared against application thresholds.

Response Level The health level of the response time for the requests. Valid values are Unknown, Good, Fair, and Bad. This attribute is determined from application requests response times collected during the interval and compared against application thresholds.

Sample Date and Time The date and time the Tivoli Enterprise Monitoring Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 95. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Web Container The health level of the web container. Valid values are Unknown, Good, Fair, and Bad.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Monitoring Configuration attributes

The **Application Monitoring Configuration** attributes provide information for the Application Monitoring Configuration. Use these attributes to monitor different J2EE applications running within an application server.

The attributes within this group are used to build the Selected Application - Configuration workspace.

Application Alias The alias name that you can optionally assign for the application. In practice, this attribute enables you to combine multiple applications under the same common alias and report their request in the Tivoli Enterprise Portal as it would come from same application. This attribute is blank by default. You can assign the value to it from Take Actions at any time in the application monitoring life cycle. The valid format is an alphanumeric string, with a maximum of 256 characters.

App ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Application Name The name of the application to which the request belongs. You can define the pattern of this name in the Application Registry workspace. The valid format is an alphanumeric string, with a maximum of 256 characters.

Bad Completion Rate (%) The bad completion rate threshold for the requests. The valid format is an alphanumeric string, with a maximum of 256 characters.

Baselining Elapsed Time The number of seconds during which the application baselining has been running. The valid format is a positive integer.

Baselining Status The status of the application baselining process. Valid values are Idle, Running, and Standby.

Baselining Scheduled Stop Time The date and time baselining is scheduled to finish. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 96. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Baselining Start Time The date and time when the application baselining was started. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 97. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Baselining Update Interval The number of seconds that defines how often active baselining data is incrementally updated to the monitoring agent. The valid format is a positive integer.

Fair Completion Rate (%) The fair completion rate threshold for the requests. The valid format is an alphanumeric string, with a maximum of 256 characters.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Monitoring Status The current application monitoring status. Valid values are Discovered, Enabled, Disabled, and Standby.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Data Monitoring Level The custom request aggregation level for all application requests. Valid values are Default, Disabled, Level1, and Level2. This attribute is set to Default when the application is first discovered.

Request Data Sampling Rate The custom request aggregation rate for all application requests. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 98. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Server Status - J2EE attributes

The **Application Server Status - J2EE** attributes provide status information for all J2EE application servers as well as the J2EE administrative server being monitored by the OMEGAMON XE agent.

The attributes within this group are used to build the "J2EE Agent workspace" on page 645.

Cluster Name The name of the server group (cluster) that the application server belongs to. The valid format is an alphanumeric string, with a maximum of 128 characters. This is supported for WebLogic application server only.

J2EE Configuration Repository Directory Name The name of the J2EE configuration repository directory, which normally resides in the config subdirectory of the product installation root directory. The valid format is an alphanumeric string, with a maximum of 128 characters.

J2EE Node Name The name of the J2EE node group that the application server belongs to. The valid format is an alphanumeric string, with a maximum of 64 characters.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process identifier of the Java virtual machine. The valid format is a positive integer.

Sample Date and Time The date and time that the OMEGAMON XE for J2EE Application Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 99. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year

Table 99. Format of the 12-character timestamp (continued)

Character String	Meaning
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Start Date and Time The date and time when the J2EE application server started. The valid format is a timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the J2EE Application Server. The valid values are Starting, Running, Stopping, Stopped, and Failed.

Server Type The type of J2EE server. The valid values are Unknown, AppServer, AdminServer.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Application Server - J2EE attributes

The **Application Server - J2EE** attributes gather status and summary data for a specific J2EE application server. They provide performance data for the J2EE Application Server runtime (JVM memory), HTTP sessions, and configuration parameters. They also provide some information from other attribute groups to provide an overall view of the J2EE application server. Use the Application Server attributes to monitor the health and performance of a J2EE application server.

The attributes within this group are used to build the J2EE application servers workspaces.

CPU Used (ms) The number of milliseconds used during the interval. The valid format is a positive integer.

CPU Used (%) The percentage of the CPU used during the interval. The valid format is a decimal (formatted to one decimal place).

Garbage Collection Monitoring This attribute indicates whether Garbage Collection is being monitored. Valid values are Disabled and Enabled.

Instrumentation Level The JVM instrumentation level. Valid values are None, Low, Medium, High, Basic, Extended, All, Custom and Maximum. This field is blank if no instrumentation level is set.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

JVM Memory Free (bytes) The amount of JVM memory that is free (in bytes). Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

JVM Memory Total (bytes) The total amount of JVM memory (in bytes). Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

JVM Memory Used (bytes) The amount of JVM memory that has been used (in bytes). Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, maximum 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, maximum 128 characters.

Process CPU Utilization (ms) The process CPU utilization. The valid format is a positive integer.

Process ID The process identifier of the Java virtual machine. The valid format is a positive integer.

Resource Data Monitoring This attribute indicates whether resource data is being monitored. Valid values are Disabled and Enabled.

Request Data Monitoring Level The monitoring level for request data stored by the data collector. Valid values are Disabled, Level1 (in other words, only edge request data-such as servlets and JSPs- are displayed), and Level2 (nested request data, such as JDBC and JMS requests, are also displayed).

Request Data Sampling Rate (%) The percentage of requests being sampled. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 100. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute

Table 100. Format of the 12-character timestamp (continued)

Character String	Meaning
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, maximum 256 characters.

Start Date and Time The date and time when the J2EE application server started. The valid format is a timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the J2EE application server. The valid values are Starting, Running, Stopping, Stopped, and Failed.

System Paging Rate (Kbytes/sec) The system paging rate during the interval. The valid format is a positive integer.

Version The version of the J2EE Application Server. The valid format is an alphanumeric string maximum 8 characters.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Baseline attributes

The **Baseline** attributes provide information for baseline extract data for the given application. The baselining collects statistical information about an application requests completion times and uses this information to assign fair and bad thresholds on the application requests. The product divides the whole request response times into buckets and collects individual hits into each bucket. Use these attributes to get statistics from individual requests collected during the baselining interval.

The attributes within this group are used to build the Selected Request - Baseline workspace.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Bad Hits (%) The percentage of bucket hits in the metric bad value zone. The valid format is a positive integer.

Bucket Number The bucket number of the baselining data. The valid format is a positive integer.

EJB (%) The average percent of time that bucket requests were executed inside EJB container. The valid format is a positive integer.

Fair Hits (%) The percentage of bucket hits in the metric fair value zone. The valid format is a positive integer.

Good Hits (%) The percentage of bucket hits in the metric good value zone. The valid format is a positive integer.

Hits (%) The percentage of hits for the bucket during the baselining. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA (%) The average percent of time that bucket requests spent for JCA access. The valid format is a positive integer.

JDBC (%) The average percent of time that bucket requests spent for JDBC access. The valid format is a positive integer.

JMS (%) The average percent of time that bucket requests spent for JMS access. The valid format is a positive integer.

JNDI (%) The average percent of time that bucket requests spent for JNDI access. The valid format is a positive integer.

Lower Boundary (msec) The lower boundary of bucket response times in milliseconds. The valid format is a positive integer.

Metric ID The metric identifier of the baselining data. The valid format is a positive integer.

Metric Type The metric type of the baselining data. Valid formats are Request, Error, and Resource.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Portal (%) The average percent of time that bucket requests were executed inside portal container. The valid format is a positive integer.

Response Time Mean (msec) The mean time of bucket response times. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 101. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Selection Hits (%) The percentage of bucket hits in the metric selection value zone. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet JSP (%) The average percent of time that bucket requests were executed inside the servlet container. The valid format is a positive integer.

Total Hits The total hits number for the bucket during the baselining. The valid format is a positive integer.

Upper Boundary (msec) The upper boundary of bucket response times. The valid format is a positive integer.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Data sources - J2EE attributes

The **Data sources - J2EE** attributes provide database usage information. For example, these attributes provide traffic information such as response times for database requests, the frequencies at which database connections are created and destroyed, and how often databases are accessed.

The attributes within this group are used to build the Data sources workspace .

Important: The attributes within this attribute group contain meaningful values only if your site has set the request data monitoring level to Level2 to collect data on data source requests.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Average Processing Time (ms) The total average processing time (in milliseconds) that the data source is used by an application. The valid format is a decimal (formatted to three decimal places).

Average Query Processing Time (ms) The average time (in milliseconds) per request used by queries to the data source. The valid format is a decimal (formatted to three decimal places).

Average Update Processing Time (ms) The average time (in milliseconds) per request used by updates to the data source. The valid format is a decimal (formatted to three decimal places).

Connection Average Wait Time (ms) The average time (in milliseconds) that applications had to wait for a connection. The valid format is a decimal (formatted to three decimal places).

Connection Count The number of connections to the data source. The valid format is a positive integer.

Connection Max Wait Time (ms) The maximum amount of time (in milliseconds) that applications had to wait for a connection to the data source. The valid format is a positive integer.

Connection Rate (per sec) The number of connection requests (per second) created for the data source. The valid format is a decimal (formatted to three decimal places).

Connection Total Wait Time (ms) The total time (in milliseconds) that applications had to wait for a connection to the data source. The valid format is a positive integer.

Database Product The name of the database product. The valid format is an alphanumeric string, with a maximum of 128 characters.

Database Product Version The version of the database product. The valid format is an alphanumeric string, with a maximum of 128 characters.

Datasource Label A shortened version of Data source Name, used to display the data source name in the chart view. The valid format is an alphanumeric string, with a maximum of 12 characters.

Datasource Name The name of the data source. The valid format is an alphanumeric string, with a maximum of 256 characters.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the process running the Java Virtual Machine (JVM). The valid format is a positive integer.

Query Count The number of queries performed against the data source. The valid format is a positive integer.

Query Rate (per sec) The number of queries (per second) being made to the data source. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 102. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate % The percentage of edge requests-such as servlets and JSPs-that were sampled for data source requests during the interval. The valid format is a positive integer.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Query Processing Time (ms) The total time (in milliseconds) used to process queries made to the data source. The valid format is a positive integer.

Total Update Processing Time (ms) The total time (in milliseconds) used to update the data source. The valid format is a positive integer.

Total Wait Time (ms) The time (in milliseconds) that applications had to wait for connections to the data source. The valid format is a positive integer.

Update Count The number of updates performed against the data source. The valid format is a positive integer.

Update Rate (per sec) The number of updates (per second) made to the data source. The valid format is a decimal (formatted to three decimal places).

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DB Connection Pools - NetWeaver attributes

The **DB Connection Pools - NetWeaver** attributes collect information about the database connection pool for each defined data source.

The attributes within this group are used to build the "DB Connection Pools workspace" on page 638.

Current Size The current size of data source pool. The valid format is a positive integer.

Datasource Name The name of the data source. The valid format is an alphanumeric sting, with a maximum of 256 characters.

Increment Step The increment step of data source pool. The valid format is a positive integer.

Init Size The initial size of data source pool. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Max Size The maximum size of data source pool. The valid format is a positive integer.

Min Size The minimum size of data source pool. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 103. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data were collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

UsedConnectionsCount The number of used connections. The valid format is a positive integer.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

DC Messages - J2EE attributes

The **DC Messages - J2EE** attribute group provides information about the data collector Messages.

The attributes within this group are used to build both the "Log Analysis workspace" on page 652 and the "DC Message Events workspace" on page 639.

Component The name of the component that caused the error. The valid format is an alphanumeric string, maximum 32 characters.

Event Date and Time The date and time that the event occurred. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 104. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

File Name The name of the file. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message Description The description of the message. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message ID The ID of the message. The valid format is an alphanumeric string, with a maximum of eight characters.

Method Name The name of the method. The valid format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Severity The severity of the message. Valid values are Info, Warning, Error, and Severe.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Sequence Number The sequence number in JMX notification stream. The valid format is a positive integer.

Thread ID The ID of the thread where the event occurred. The valid format is an alphanumeric string, with a maximum of 16 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Enterprise Java Bean Components - WebLogic attributes

The **WebLogic EJB Components - WebLogic** attributes provide the runtime information for an EJB component.

The attributes within this group are used to build the “EJB Components workspace” on page 640.

Current Entity EJBs The number of current entity EJBs. The valid format is a positive integer.

Current Message Driven EJBs The number of current message driven EJBs. The valid format is a positive integer.

Current Stateful EJBs The number of current stateful EJBs. The valid format is a positive integer.

Current Stateless EJBs The number of current stateless EJBs. The valid format is a positive integer.

Deployment State The current deployment state of the component. Valid values are Unprepared, Prepared, Activated, and New.

Enterprise Application Name The J2EE application name. The valid format is an alphanumeric string, with a maximum of 128 characters.

EJB Component Name The EJB Component name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 105. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Enterprise Java Bean Modules - J2EE attributes

The **Enterprise Java Bean Modules - J2EE** attributes collect performance information about each Enterprise Java Bean (EJB) deployed to the application server.

The attributes within this group are used to build the “EJB Modules workspace” on page 642 and “Enterprise Java Beans workspace” on page 641.

Bean Name The name of the Enterprise Java Bean (EJB); this name prefixes the application name and the EJB JAR name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Bean Type The type of bean. Valid values are Stateless, Stateful, Entity, and Message_Driven.

Create Count The number of times that beans were created during the interval. Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

EJB Module The name of the EJB Module. The valid format is an alphanumeric string, with a maximum of 128 characters.

EJB Count The number of EJBs in the EJB module. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Message Count The number of messages delivered to the bean on Message method. The valid format is a positive integer.

Method Ready Count The number of bean instances in ready state. The valid format is a positive integer.

Method Ready Count High The high water mark of the number of bean instances in ready state. The valid format is a positive integer.

Method Ready Count Low The low water mark of the number of bean instances in ready state. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Passive Count The number of beans that are in a passivated state (entity and stateful). Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

Passive Count High The high water mark of the beans that are in a passivated state (entity and stateful). The valid format is a positive integer.

Passive Count Low The low water mark of the beans that are in a passivated state. The valid format is a positive integer.

Pooled Count The average number of objects in the pool. The valid format is a positive integer.

Pooled Count High The high water mark of the average number of objects in the pool. The valid format is a positive integer.

Pooled Count Low The low water mark of the average number of objects in the pool. The valid format is a positive integer.

Ready Count The number of bean instances in ready state. The valid format is a positive integer.

Ready Count High The high water mark of bean instances in ready state. The valid format is a positive integer.

Ready Count Low The low water mark of bean instances in ready state. The valid format is a positive integer.

Remove Count The number of times that beans were removed. Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 106. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Summary of All EJB Modules The summary of statistical totals for all EJB modules. Valid values are EJB, No, and Yes.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Enterprise Java Bean Service - NetWeaver attributes

The **Enterprise Java Bean Service - NetWeaver** attributes collect performance information about each Enterprise Java Bean deployed to SAP NetWeaver application server.

The attributes within this group are used to build the “Enterprise Java Beans workspace” on page 641 in SAP NetWeaver Server.

Activations Number The number of activations. The valid format is a positive integer.

Active Sessions Count The count of not passivated sessions. The valid format is a positive integer.

Active Sessions Timeout The timeout for the active sessions. If a session stays idle and not passivated for this long, it is removed. The valid format is a positive integer.

Application Name The name of application using EJB. The valid format is an alphanumeric string, with a maximum of 128 characters.

Bean Name The bean class name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Bean Type The type of Bean. Valid formats are Stateless, Stateful, Entity, Message_Driven, and [Summary].

Create Count The count of times of a "create" method was invoked on the bean. The valid format is a positive integer.

Current Pool Size The current size of pool. The valid format is a positive integer.

Completed Sessions The count of already completed sessions. The valid format is a positive integer.

EJB Count The number of EJB in the EJB module. The valid format is a positive integer.

EJB Module The name of the EJB Module. The valid format is an alphanumeric sting, with a maximum of 128 characters.

Initial Pool Size The initial size of pool. The valid format is a positive integer.

Interval Time The length of the interval in seconds. The valid format is a positive integer.

Loads Number The number of loads. The valid format is a positive integer.

Max Pool Size The maximum size of pool. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Passive Sessions Count The count of passivated sessions. The valid format is a positive integer.

Passive Sessions Timeout The timeout for the passive sessions. If a session stays idle and passivated for this long, it is removed. The valid format is a positive integer.

Passivations Number The number of passivations. The valid format is a positive integer.

PoolCurrUsed0bj The number of currently used pool objects. The valid format is a positive integer.

Pool Increment Size The size of pool increment. The valid format is a positive integer.

Remove Count The count of times of a "remove" method was invoked on the bean. The valid format is a positive integer.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 107. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Stores Number The number of the stores. The valid format is a positive integer.

Summary of All EJB Modules The summary row of statistical totals for all EJB modules. Valid values are EJB and Yes.
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Enterprise Java Beans - WebLogic attributes

The **Enterprise Java Beans - WebLogic** attributes collect performance information about each Enterprise Java Bean deployed to the WebLogic application server.

The attributes within this group are used to build the “EJBs - Selected Enterprise Application workspace” on page 643.

Activation Rate The number of beans from this EJB Home that have been activated per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Beans Destroyed Percent The percent of the number of beans destroyed from the number of requests for a bean for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Cache Accesses The number of attempts to access a bean from the cache for the interval since the previous sample. The valid format is a positive integer.

Cache Access Rate The number of attempts per second to access a bean from the cache for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Cache Miss Percent The percent of the number of times a container cannot find a bean in the cache to the number of times it attempts to find a bean in the cache for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Current Cached Beans The total number of beans from this EJB Home currently in the EJB cache. The valid format is a positive integer.

Current Lock Entries The current number of beans currently locked. The valid format is a positive integer.

Current Lock Waiters The current number of threads that wait for a lock on a bean. The valid format is a positive integer.

Current Pool Beans The summary number of free and in-use beans in the pool. The valid format is a positive integer.

Current Pool Free Percent The percent of the free beans available in the pool. The valid format is a decimal (formatted to one decimal place).

Current Pool Waiters The number of threads currently waiting for an available bean instance from the free pool. The valid format is a positive integer.

Enterprise Application Name The J2EE application name. The valid format is an alphanumeric string, with a maximum of 128 characters.

EJB Component Name The EJB Component name. The valid format is an alphanumeric string, with a maximum of 128 characters.

EJB Name The EJB-name for this EJB. It is as defined in the ejb-jar.xml deployment descriptor. The valid format is an alphanumeric string, with a maximum of 128 characters.

EJB Type The type of EJB. Valid values are Stateless, Stateful, Entity and Message Driven.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Lock Manager Accesses The number of attempts to obtain a lock on a bean since the previous sample. The valid format is a positive integer.

Lock Manager Access Rate The number of attempts to obtain a lock on a bean per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Lock Timeout Percent The percent of timeouts to accesses for the lock manager for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Lock Waits Percent The percent of times a thread had to wait to obtain a lock on a bean comparing to the total amount of lock requests issued for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Passivation Rate The number of beans from this EJB Home that have been passivated per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Pool Accesses The number of times an attempt was made to get an instance from the free pool since the previous sample. The valid format is a positive integer.

Pool Access Rate The number of times per second an attempt was made to get an instance from the free pool for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Pool Miss Percent The percent of times a request was made to get a bean from the pool when no beans were available. The valid format is a decimal (formatted to one decimal place).

Pool Timeout Percent The percent of requests that have timed out waiting for a bean from the pool from the total number of requests made for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 108. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Transactions Processed The number of transactions processes. The valid format is a positive integer.

Transaction Process Rate The number of transactions processed per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Transaction Rolled Back Percent The percent transactions that have rolled back to the number of total transactions involving the EJB for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Transactions Timed Out Percent The percent of transactions that have timed out to the number of total transactions involving the EJB for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collection Analysis - J2EE attributes

The **Garbage Collection Analysis - J2EE** attribute group provides information from the profiler about the garbage collector in the Java Virtual Machines that are hosting an application server. For example, these attributes report the number of times the collector ran during the interval and the resulting number of objects that the collector freed. Use the Garbage Collection Analysis attributes in situations to monitor garbage-collection performance and possible problems.

The attributes within this group are used to build the “Garbage Collection Analysis workspace” on page 643 and the J2EE application servers workspaces.

GC Rate (per min) The rate (per minute) at which the Java Virtual Machine is invoking its garbage-collection routine. The valid format is a decimal (formatted to three decimal places).

Heap Used (%) The percentage of heap used at the end of interval. The valid format is a decimal (formatted to one decimal places).

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Kbytes Free The total number of free kilobytes in the heap at the end of the last garbage-collection cycle during the interval. The valid format is a positive integer.

Kbytes Used The number of kilobytes in the heap that were in use at the end of the last garbage collection cycle during the interval. The valid format is a positive integer.

Kbytes Used Delta The delta value between the "Kbytes in Use" value for this interval and the "Kbytes in Use" value for the prior interval. A positive value indicates that the number of kbytes in use grew during the interval. The valid format is a positive or negative integer.

Kbytes Total Freed by GC The total number of kbytes freed by the garbage collector during the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Objects Freed The number of objects the garbage collector freed during the interval (supported for IBM Developer Kit for Java). The valid format is a positive integer.

Objects Moved The number of objects the garbage collector moved during the interval (supported for IBM Developer Kit for Java). The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process ID of the JVM. The valid format is a positive integer.

Real Time (ms) The total real time (in milliseconds) that the garbage collector required during the most recent cycle. The valid format is a positive integer.

Real Time % The percentage of real time that the garbage collector was active during the interval. The valid format is a decimal (formatted to one decimal place).

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 109. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Times Run The number of times the garbage collector ran during the interval. The valid format is a positive integer.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Garbage Collection Cycle - J2EE attributes

The **Garbage Collection Cycle - J2EE** attribute group provides information about a single garbage-collection cycle that the Java Virtual Machine hosting the application server performed. For example, these attributes report the free heap space before and after garbage collection, the heap space freed, and the number of objects moved during garbage collection. Use the Garbage Collection Cycle attributes in situations to examine the results of a particular garbage collection.

The attributes within this group are used to build the Garbage Collections - Selected Allocation Failure workspace .

Allocation Failure Number The allocation failure block number for which this cycle ran (supported for IBM Developer Kit for Java). The valid format is a positive integer.

Compact (ms) The time (in milliseconds) required for the compaction phase of the garbage-collection cycle. The valid format is a positive integer.

Compaction Reason The code describing the reason garbage collection was initiated. The valid format is a positive integer.

The compaction codes are shown in the following table:

Table 110. Reasons for initiating garbage collection

Compaction Code	Definition
1	Insufficient free space for the allocation request following the mark and sweep phases.
2	The heap is fragmented and benefits from a compaction.
3	Less than 15% free space available.
4	A call to System.gc requested garbage collection.
5	Less than 5% free space available.
6	Less than 128 K free space available.
7	Parameter Xcompactgc specified.
8	The transient heap has less than 5% free space available.
9	The heap is fragmented (this code marks additional reasons for compaction apart from compaction code 2).

Final sees The Final references that are collected. The valid format is a positive integer.

Garbage Collection Number The garbage collection cycle number. The valid format is a positive integer.

GC Date and Time The date and time the Java Virtual Machine invoked the garbage collector. The valid format is a 16-character timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Note to Solaris and HP-UX users: Because allocation-failure information is not recorded on these platforms, this column is always empty on these platforms.

Heap Capacity The total number of kilobytes allocated to the main heap after this garbage-collection cycle. The valid format is a positive integer.

Heap Free % after GC The percentage of heap space that is available after this garbage-collection cycle. The valid format is a decimal (formatted to one decimal place).

Heap Space Free (kbytes) The number of kilobytes available within the heap after this garbage-collection cycle. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Kbytes Free at Start of GC The number of kilobytes available in the heap before garbage collection began. The valid format is a positive integer.

Kbytes Freed The number of kilobytes freed by the garbage collector. The valid format is a positive integer.

Kbytes Moved The number of kilobytes moved on the heap during this compaction. The valid format is a positive integer.

Kbytes Used The number of kilobytes in the heap that were in use after this garbage-collection cycle. The valid format is a positive integer.

Mark (ms) The time (in milliseconds) required for the mark phase of the garbage-collection cycle. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Objects Moved The number of objects the garbage collector moved during this compaction. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Phantom sees The number of phantom reference objects collected during this garbage-collection cycle. "Phantom" refers to a specific Java class that defines object reachability. The valid format is a positive integer.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 111. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year

Table 111. Format of the 12-character timestamp (continued)

Character String	Meaning
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Soft sees The number of soft reference objects collected during this garbage-collection cycle. "Soft" refers to a specific Java class that defines object reachability. The valid format is a positive integer.

Sweep (ms) The time (in milliseconds) required for the sweep phase of the garbage-collection cycle. The valid format is a positive integer.

Time to Complete (ms) The time (in milliseconds) required to complete this garbage-collection cycle. The valid format is a positive integer.

Weak sees The number of weak reference objects collected during this garbage-collection cycle. "Weak" refers to a specific Java class that defines object reachability. The valid format is a positive integer.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

J2EE Agent Events attributes

The **J2EE Agent Events** attributes collect information about agent-level events that affect the ability of the OMEGAMON XE agent to collect data about J2EE Application Server. These attributes provide error messages, their IDs, and their severities.

The attributes within this group are used to build the “J2EE Agent workspace” on page 645.

Event Date and Time The date and time that the event occurred. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 112. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day

Table 112. Format of the 12-character timestamp (continued)

Character String	Meaning
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

File Name The name of the file. The valid format is an alphanumeric string, with a maximum of 64 characters.

Function Indicates the description of the message. The valid format is an alphanumeric sting, with a maximum of 32 characters.

Message Description The message description. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message ID The message ID. The valid format is an alphanumeric string, with a maximum of eight characters.

Node Name The system on which the server is running. The valid format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Severity The severity of the message. Valid values are Info, Warning, Error, and Severe.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

J2EE Connector Connection Pools - WebLogic attributes

The **J2EE Connector Connection Pools - WebLogic** attributes provide data and statistics for the Oracle/BEA WebLogic Connector Connection Pools.

The attributes within this group are used to build the "JCA Connection Pools workspace" on page 647.

Connection Creation Rate The number of Connector connections created in this Connection Pool per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connection Destroy Rate The number of Connector connections destroyed in this Connector Pool per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connections Matched The number of times a request for a Connector connection was satisfied via the use of an existing created connection since the previous sample. The valid format is a positive integer.

Connection Match Rate The number of times a request for a Connector connection was satisfied per second via the use of an existing created connection since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connections Recycled The number of Connector connections that have been recycled in this Connector Pool since the previous sample. The valid format is a positive integer.

Connection Recycled Rate The number of Connector connections that have been recycled in this Connector Pool per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connection Rejected The number of rejected requests for a Connector connection since the previous sample. The valid format is a positive integer.

Connection Rejection Rate The number of rejected requests for a Connector connection per second since the previous sample. The valid format is a decimal (formatted to three decimal places).

Cumulative Average Active Connections Usage The running average usage of created connections that are active in the Connector Pool since the pool was last shrunk. The valid format is a positive integer.

Current Active Connections The current number of active connections. The valid format is a positive integer.

Current Free Connections The current number of free connections. The valid format is a positive integer.

Interval (ms) The length of the interval in seconds. The valid format is a positive integer.

JNDI Name The configured JNDI Name for the Connection Factory using this Connector connection pool. The valid format is an alphanumeric string, with a maximum of 128 characters.

Highest Active Connections The high water mark of active connections in this Connector Pool since the pool was instantiated. The valid format is a positive integer.

Highest Free Connections The high water mark of free connections in this Connector Pool since the pool was instantiated. The valid format is a positive integer.

Idle Connections Detected The number of idle connections detected for the interval since the previous sample. The valid format is a positive integer.

Leaked Connections Detected The number of leaked connections detected for the interval since the previous sample. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Maximum Capacity The maximum capacity configured for this Connector connection pool. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time the Tivoli Enterprise Management Agent for WebLogic Server agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 113. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JCA Connection Pools - J2EE attributes

The **JCA Connection Pools - J2EE** attribute group collects information about the JCA connection pools for each connection factory.

The attributes within this group are used to build the JCA Connection Pools workspace .

Average Usage Time (ms) The average time in milliseconds that a connection was in use. The valid format is a decimal (formatted to three decimal places).

Average Pool Size The average number of Managed Connections for the interval. Minimum instrumentation level required to collect these data: High. The valid format is a decimal (formatted to three decimal places).

Average Wait Time (ms) The average time in milliseconds that a client waited to be granted a connection. The valid format is a decimal (formatted to three decimal places).

Connections Closed The number of connections released. The valid format is a positive integer.

Connections Created The total number of Managed Connections created during the sampling interval. Minimum instrumentation level required to collect these data: Low. The valid format is a positive integer.

Connection Factory The name of Connection Factory. The valid format is an alphanumeric string, with a maximum of 256 characters.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA Pool Label The short name of the JCA Pool. The valid format is an alphanumeric string, with a maximum of 32 characters.

JCA Pool Usage % The percentage of the pool that was used during the sampling interval. The valid format is a decimal (formatted to three decimal places).

Maximum Pool Size The maximum number of managed connections that can be created in this connection pool (the field is blank for each individual managed connection). The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 114. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Usage (ms) The total time used. The valid format is a decimal (formatted to three decimal places).

Total Wait (ms) The total time wait. The valid format is a decimal (formatted to three decimal places).

Waiting Threads The number of threads waiting for a connection. The valid format is a positive integer.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JDBC Connection Pools - WebLogic attributes

The **JDBC Connection Pools - WebLogic** attributes provide data and statistics for JDBC connection pools.

The attributes within this group are used to build the "JDBC Connection Pools workspace" on page 648.

Average Connection Delay The averaged time necessary to get a connection from the database for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connections Created The number of JDBC connections created during the interval. The valid format is a positive integer.

Connection Creation Rate The number of JDBC connections per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Cumulative Average Active Connections The running average of active connections in this pool. The valid format is a positive integer.

Current Active Connections The current number of active connections in this pool. The valid format is a positive integer.

Current Available Connections The current number of connections that are available to applications. The valid format is a positive integer.

Current Capacity The current number of database connections in this pool. The valid format is a positive integer.

Current Unavailable Connections The current number of connections in this pool that are being tested or refreshed and not available to the applications. The valid format is a positive integer.

Current Waiters The current number of waiters for a connection. The valid format is a positive integer.

Leaked Connections Detected The number of leaked connections for the interval since the previous sample. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Highest Active Connections The high water mark of active connections in this pool. The valid format is a positive integer.

Highest Available Connections The highest number of connections that were available to applications. The valid format is a positive integer.

Highest Unavailable Connections The highest number of connections in this pool that are being tested or refreshed and not available to the applications. The valid format is a positive integer.

Highest Waiters The highest number of waiters for a connection. The valid format is a positive integer.

Highest Wait Time (sec) The number of seconds the longest waiter for a connection waited. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

JDBC Pool Name The JDBC connection pool name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Maximum Capacity The maximum capacity of this connection pool. The valid format is a positive integer.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Pool State The current state of the connection pool. Valid values are Running, Suspended, Unhealthy, and Unknown.

Prepared Statement Cache Access The number of prepared statement cache accesses. The valid format is a positive integer.

Prepared Statement Cache Misses Percent The percent of the number of times a prepared statement was not found in the cache to the total number of requests for a prepared statement for the interval since the previous sample. The valid format is a decimal (formatted to one decimal place).

Reconnect Failures The number of cases when a connection pool attempted to refresh a connection to a database and failed. The valid format is a positive integer.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 115. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year

Table 115. Format of the 12-character timestamp (continued)

Character String	Meaning
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JDK - Operation System attributes

The **JDK - Operation System** attributes provide data and statistics of the operating system on which the J2SE application server is running. The information includes the operating system architecture, name, version, and the available memory information. The data for attributes is gathered from JVM MBeans, which are supported in JVM version 1.5 and higher. No data is available for JVM with versions lower than 1.5.

The attributes within this group are used to build the "JVM Statistics workspace" on page 651.

Architecture The architecture of the operating system on which the J2SE application server is running (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Available Processors The number of available processors of the operation system on which the J2SE application server is running (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Committed Virtual Memory Size The committed virtual memory size in kilobytes (supported for JVM version 1.5 and higher). The valid format is a positive integer or N/A if this attribute is not applicable.

Free Physical Memory The free physical memory (in kilobytes) of the operation system (supported for JVM version 1.5 and higher). The valid format is a positive integer or N/A if this attribute is not applicable.

Free Swap Space Size The free swap space size (in kilobytes) of the operating system (supported for JVM version 1.5 and higher). The valid format is a positive integer or N/A if this attribute is not applicable.

Interval (sec) The length of the interval in seconds (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Node Name The name of the system on which the server is running (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 128 characters.

OS Name The name of the operating system, such as Windows 2003 (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

OS Version The version of the operating system on which the J2SE application server is running (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Processing Capacity The processing capacity of operation system (supported for JVM version 1.5 and higher). The valid format is a positive integer or N/A if this attribute is not applicable.

Sample Date and Time The date and time that the monitoring agent collected data (supported for JVM version 1.5 and higher). The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 116. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Physical Memory The total physical memory (in kilobytes) of the operation system (supported for JVM version 1.5 and higher). The valid format is a positive integer or N/A if this attribute is not applicable.

Total Swap Space Size The total swap space size (in kilobytes) of the operating system (supported for JVM version 1.5 and higher). The valid format is a positive integer or N/A if this attribute is not applicable.

- Organization of the predefined workspaces

- Attribute groups used by the predefined workspaces

JDK - Memory attributes

The **JDK - Memory** attributes provide the memory usage information of the operating system on which the J2SE application server is running. This includes heap memory information, heap memory usage information, and pending objects. The data for attributes is gathered from JVM MBeans, which are supported in JVM version 1.5 and higher. No data is available for JVM with versions lower than 1.5.

The attributes within this group are used to build the "JVM Statistics workspace" on page 651.

Committed Heap Memory Size The amount of committed heap memory (in kilobytes) allocated to the JVM (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Committed Non Heap Memory Size The amount of committed non-heap memory (in kilobytes) allocated to the JVM (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Initial Heap Memory Size The initial amount of the heap memory (in kilobytes) for the JVM of the J2SE application server (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Initial Non Heap Memory Size The initial amount of the non-heap memory (in kilobytes) for the JVM of the J2SE application server (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Interval (sec) The length of the interval in seconds (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Maximum Heap Memory Size The maximum amount of the heap memory (in kilobytes) used by the JVM of the J2SE application server (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Maximum Non Heap Memory Size The maximum amount of the non-heap memory (in kilobytes) used by the JVM of the J2SE application server (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Node Name The name of the system on which the server is running (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Object Pending Finalization Count The number of objects that are not finalized (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Origin Node The name of the server subnode (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time that the monitoring agent collected data (supported for JVM version 1.5 and higher). The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 117. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Used Heap Memory Size The amount of heap memory (in kilobytes) used by the JVM of the J2SE application server (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Used Non Heap Memory Size The amount of non-heap memory (in kilobytes) used by the JVM of the J2SE application server (supported for JVM version 1.5 and higher). The valid format is a positive integer.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JDK - JVM attributes

The **JDK - JVM** attributes provide overall information about the Java Virtual Machine (JVM) that the J2SE application server is using. This information includes the information about JVM name, version, and uptime. The data for attributes is gathered from JVM MBeans, which are supported in JVM version 1.5 and higher. No data is available for JVM with versions lower than 1.5.

The attributes within this group are used to build the "JVM Statistics workspace" on page 651.

Formatted Uptime The time with a specific format during which the Java Virtual Machine is running (supported for JVM version 1.5 and higher). The format is DDd HHh MMm SSs; For example, 1d 2 h 44 m 23 s.

Interval (sec) The length of the interval in seconds (supported for JVM version 1.5 and higher). The valid format is a positive integer.

JVM Name The name of the Java Virtual Machine (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

JVM Vendor The producer of the Java Virtual Machine (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

JVM Version The version of the Java Virtual Machine (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time that the monitoring agent collected data (supported for JVM version 1.5 and higher). The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 118. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Start Time The date and time when the Java Virtual Machine was started (supported for JVM version 1.5 and higher). The valid format is a 12-character timestamp. This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Uptime The amount of time (in seconds) the JVM has been running (supported for JVM version 1.5 and higher). The valid format is a positive integer.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JDK - Threading attributes

The **JDK - Threading** attributes provide overall information about the threads, including peak thread and daemon thread. A thread is 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. The data for attributes is gathered from JVM MBeans, which are supported in JVM version 1.5 and higher. No data is available for JVM with versions lower than 1.5.

The attributes within this group are used to build the "JVM Statistics workspace" on page 651.

Current Thread CPU Time The CPU time (in seconds) used to process the current thread (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Current Thread User Time The user time (in seconds) used for the current thread (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Daemon Thread Count The number of threads which run unattended to perform continuous or periodic functions (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Interval (sec) The length of the interval in seconds (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Node Name The name of the system on which the server is running (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 128 characters.

Peak Thread Count The maximum number of threads executed in the Java Virtual Machine (supported for JVM version 1.5 and higher). The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent collected data (supported for JVM version 1.5 and higher). The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 119. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server (supported for JVM version 1.5 and higher). The valid format is an alphanumeric string, with a maximum of 256 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JMS Sessions - WebLogic attributes

The **JMS Sessions - WebLogic** attribute group provides statistic for JMS session in WebLogic application server.

The attributes within this group are used to build the "JMS Sessions workspace" on page 649.

Acknowledge Mode The acknowledge mode. The valid format is an alphanumeric string, with a maximum of 64 characters.

Bytes Received The number of bytes received by this session since the previous sample. The valid format is a positive integer.

Byte Receive Rate The number of bytes received by this session per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Bytes Sent The number of bytes sent by this session since the previous sample. The valid format is a positive integer.

Byte Send Rate The number of bytes sent by this session per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Connection Name The name of the connection. The valid format is an alphanumeric string, with a maximum of 128 characters.

Consumers Created The number of consumers instantiated by this session since the previous sample. The valid format is a positive integer.

Consumer Creation Rate The number of consumers instantiated by this session per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Current Bytes Pending The number of bytes pending (uncommitted and unacknowledged) for this session. The valid format is a positive integer.

Current Consumers The current number of consumers for this session. The valid format is a positive integer.

Current Messages Pending The number of messages pending (uncommitted and unacknowledged) for this session. The valid format is a positive integer.

Current Producers The current number of producers for this session. The valid format is a positive integer.

Highest Consumers The peak number of consumers for this session since the last reset. The valid format is a positive integer.

Highest Producers The peak number of producers for this session since the last reset. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Is Transacted Whether the session is transacted. Valid values are Yes and No.

Messages Received The number of messages received by this session since the previous sample. The valid format is a positive integer.

Message Receive Rate The number of messages received by this session per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Messages Sent The number of messages sent by this session since the previous sample. The valid format is a positive integer.

Message Send Rate The number of messages sent by this session per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Producers Created The number of producers for this session created since the previous sample. The valid format is a positive integer.

Producer Creation Rate The number of producers for this session created per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 120. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour

Table 120. Format of the 12-character timestamp (continued)

Character String	Meaning
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Session Name The name of the session. The valid format is an alphanumeric string, with a maximum of 128 characters.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JMS Summary - J2EE attributes

The **JMS Summary - J2EE** attributes provide information about how J2EE Application Server applications are interacting with messaging middleware (J2EE MQ) using the Java Messaging Service (JMS). It provides such information as which queue managers and queues are being used and how many messages are being read and written.

The attributes within this group are used to build the JMS Summary workspace.

Important: The attributes within this attribute group contain meaningful values only if your site has set the request data monitoring level to Level2 to collect data on JMS requests.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Average Processing Time (ms) The average time (in milliseconds) per request using the JMS. The valid format is a decimal (formatted to three decimal places).

Browse Average Time (ms) The average time (in milliseconds) that it takes for each browse request from the queue to be processed. The valid format is a decimal (formatted to three decimal places).

Browse Count The number of messages browsed from the queue. The valid format is a positive integer.

Browse Rate (per sec) The number of messages (per second) browsed from a JMS queue. The valid format is a decimal (formatted to three decimal places).

Browse Total Time (ms) The total time (in milliseconds) used by browse requests from the queue. The valid format is a positive integer.

Full Name The complete name of the message queue, which consists of the queue manager name concatenated to the queue name and separated by a slash. The valid format is an alphanumeric string, with a maximum of 100 characters.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JMS Connection Label A shortened version of the full name. The valid format is an alphanumeric string, with a maximum of 12 characters.

Manager Name The name of the J2EE MQ queue manager (not supported currently). This attribute is blank now. The valid format is an alphanumeric string, with a maximum of 48 characters.

Name The name of the J2EE MQ queue. The valid format is an alphanumeric string, with a maximum of 48 characters.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Publish Average Time (ms) The average time (in milliseconds) that it takes for each publish request to be sent to the queue. The valid format is a decimal (formatted to three decimal places).

Publish Count The number of publish requests sent to the queue. The valid format is a positive integer.

Publish Rate (per sec) The number of publish requests (per second) sent to a JMS queue. The valid format is a decimal (formatted to three decimal places).

Publish Total Time (ms) The total time (in milliseconds) used by all publish requests for the queue. The valid format is a positive integer.

Receive Average Time (ms) The average time (in milliseconds) for each get from the queue. The valid format is a decimal (formatted to three decimal places).

Receive Count The number of destructive gets from the queue. The valid format is a positive integer.

Receive Rate (per sec) The number of destructive gets (per second) made from the queue. The valid format is a decimal (formatted to three decimal places).

Receive Total Time (ms) The total time (in milliseconds) consumed by gets from the queue. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 121. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Send Average Time (ms) The average time (in milliseconds) for each put to the queue. The valid format is a decimal (formatted to three decimal places).

Send Count The number of messages put to the queue. The valid format is a positive integer.

Send Rate (per sec) The number of messages (per second) put to the queue. The valid format is a decimal (formatted to three decimal places).

Send Total Time (ms) The total time (in milliseconds) consumed by puts to the queue. The valid format is a positive integer.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Time (ms) The total time (in milliseconds) spent accessing the queue. The valid format is a positive integer.

Type The type of message manager. The valid values are Queue and Topic.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Java Transaction Service - WebLogic attributes

The **Java Transaction Service - WebLogic** attributes provides statistics within a WebLogic server.

The attributes within this group are used to build the "JTA Resources workspace" on page 650.

Committed Transactions Time (ms) The summary number of seconds for all committed transactions since the previous sample. The valid format is a positive integer.

Current Active Transactions The number of active transactions on the server. The valid format is a positive integer.

Health State The health state of the JTA subsystem. Valid values are Warning, Critical, and Failed.

Health Reason Code The reason code of the health of the JTA subsystem. The valid format is an alphanumeric string, with a maximum of 128 characters.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 122. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Transactions Abandoned The number of transactions that were abandoned since the previous sample. The valid format is a positive integer.

Transaction Abandon Rate The number of transactions that were abandoned per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Transactions Committed The number of committed transactions since the previous sample. The valid format is a positive integer.

Transaction Commit Rate The number of committed transactions per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Transaction Heuristic Completions The number of transactions that completed with a heuristic status since the previous sample. The valid format is a positive integer.

Transaction Heuristic Completion Rate The number of transactions that completed with a heuristic status per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Transactions Processed The number of transactions proceeded since the previous sample. The valid format is a positive integer.

Transaction Process Rate The number of transactions processed per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Transaction Rolled Back The number of transactions that were rolled back since the previous sample. The valid format is a positive integer.

Transaction Rolled Back by Application The number of transactions that were rolled back due to an application error since the previous sample. The valid format is a positive integer.

Transaction Rolled Back by Application Percent The percent of transactions that were rolled back due to an application error. The valid format is a decimal (formatted to one decimal place).

Transaction Rolled Back by Resource The number of transactions that were rolled back due to a resource error since the previous sample. The valid format is a positive integer.

Transactions Rolled Back by Resource Percent The percent of transactions that were rolled back due to a resource error since the previous sample. The valid format is a decimal (formatted to one decimal place).

Transactions Rolled Back by System The number of transactions that were rolled back due to an internal system error since the previous sample. The valid format is a positive integer.

Transactions Rolled Back by System Percent The percent of transactions that were rolled back due to an internal system error since the previous sample. The valid format is a decimal (formatted to one decimal place).

Transactions Rolled Back by Timeout The number of transactions that were rolled back since the previous sample due to a timeout expiration. The valid format is a positive integer.

Transaction Rolled Back by Timeout Percent The percent of transactions that were rolled back due to a timeout expiration since the previous sample. The valid format is a decimal (formatted to one decimal place).

Transaction Rollback Rate The number of transactions that were rolled back per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JTA Resources - J2EE attributes

The **JTA Resources - J2EE** attributes group collects information about the Java Transaction API (JTA) Resources.

The attributes within this group are used to build the "JTA Resources workspace" on page 650.

Active Count The number of active transactions. The valid format is a positive integer.

Committed Count The number of committed transactions. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JTA Resource The name of the JTA Resources. The valid format is an alphanumeric string, with a maximum of 256 characters.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Rollback Count The number of Rollback transactions. The valid format is a positive integer.

Row Number The row number. The valid format is a positive integer.

For additional information, see:

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 123. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

JTA Summary - NetWeaver attributes

The **JTA Summary - NetWeaver** attributes provide Transaction Service performance data.

The attributes within this group are used to build the "JTA Summary workspace" on page 651.

Active Transactions Count The number of active transactions. The valid format is a positive integer.

Committed Transactions Count The number of transactions that have been committed. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Rolled Back Transactions Count The number of transactions that have been rolled back. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 124. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Suspended Transactions Count The number of suspended transactions. The valid format is a positive integer.

Timeouted Transactions Count The number of transactions that have timed out.
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Log Analysis - J2EE attributes

The **Log Analysis - J2EE** attributes provide application server error and exception conditions as recorded in the application server log file. The log file is SystemOut.log. Use the Log Analysis attributes in situations to monitor errors and exception conditions and their severity.

The attributes within this group are used to build the Log Analysis workspace .

Component The name of the component that caused the error. The valid format is an alphanumeric string, with a maximum of 32 characters.

Event Date and Time The date and time when the event occurred. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 125. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Job ASID The identifier assigned to the address space running this servant region. The valid format is an alphanumeric string, with a maximum of four characters.

Job Name Where the message originates; that is, the log file name and line number. The valid format is an alphanumeric string, with a maximum of eight characters.

Message ID The identifier assigned to the message. The valid format is an alphanumeric string, with a maximum of 12 characters.

Message Origin Where the message originates; that is, the log file name and line number. The valid format is an alphanumeric string, with a maximum of 32 characters.

Message Text The text of the error message. The valid format is alphanumeric string, with a maximum of 256 characters. All error message text data that goes beyond 256 characters is truncated and not shown in the portal.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process identifier of the Java virtual machine. The valid format is an alphanumeric sting, with a maximum of eight characters.

Sequence Number The sequence number in JMX notifications stream. The valid format is a positive integer.

Server Instance Name The name of the application server instance. This is the name of a single address space that can run application code (called a "specific server" or simply a "server") . The valid format is an alphanumeric string, with a maximum of eight characters.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Severity The severity of the message. The valid values are shown in the following table:

Table 126. Message severities and their meanings

Severity	Definition
Informational	A message intended to convey only user information
Unconditional	A message of type Unconditional
Dump	A message of type Dump
SystemOut	A message written directly to System.out by the user application or internal components
SystemError	A message written directly to System.err by the user application or internal components
User	A message of type user
EntryMethod	A message written upon entry to a method
ExitMethod	A message written upon exit from a method
Event	A message of type Event
Debug	A message of type Debug
Audit	An audit message
Warning	A warning message
Error	An error message

Table 126. Message severities and their meanings (continued)

Severity	Definition
Terminate	A message of type Terminate (exit process)
Fatal	A fatal message
Unknown	A placeholder that indicates the message type was not recognized

Thread ID The unique identifier of the thread where the event occurred. The valid format is an alphanumeric string, maximum 16 characters.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces
- "Situations for ITCAM Agent for J2EE" on page 738

Request Analysis - J2EE attributes

The **Request Analysis - J2EE** attributes provide response times and functional decomposition information about requests (servlets, JSPs, and EJB methods) that ran on the application server.

The attributes within this group are used to build the Request Analysis workspace .

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Application Time (ms) The average time (in milliseconds) this request spent processing application requests other than JCA, JMS, JNDI, and JDBC requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

Application Time (% of Average Response) The percentage of time this request spent processing application requests other than JCA, JMS, JNDI, and JDBC requests. The valid format is a decimal (formatted to one decimal place).

Application Bad Delay (msec) The delay time (in milliseconds) in the application tier collected during the interval. This column is non-zero value when application delay exceeds the bad threshold configured for it. The valid format is a positive integer.

Application Fair Delay (msec) The delay time (in milliseconds) in the application tier collected during the interval. This column is non-zero value when application delay exceeds the fair threshold configured for it. The valid format is a positive integer.

Application Good Delay (msec) The delay time (in milliseconds) in the application tier collected during the interval. This column is non-zero when application delay is less than fair threshold configured for it. The valid format is a positive integer.

Application Tier Delay Type This attribute defines the request type based on its delay time in the application tier. Valid values are Unknown, Good, Fair, and Bad.

Application Tier Response (msec) The actual response time in milliseconds in the application tier collected during the interval. The valid format is a positive integer.

Average Response (ms) The average time (in milliseconds) required each time this request ran during the interval. The valid format is a positive integer.

Backend Bad Delay (msec) The delay time (in milliseconds) in the backend tier collected during the interval. This column is non-zero value when backend delay exceeds the bad threshold configured for it. The valid format is a positive integer.

Backend Fair Delay (msec) The delay time (in milliseconds) in the backend tier collected during the interval. This column is non-zero value when backend delay exceeds the fair threshold configured for it. The valid format is a positive integer.

Backend Good Delay (msec) The delay time (in milliseconds) in the backend tier collected during the interval. This column is non-zero when backend delay is less than fair threshold configured for it. The valid format is a positive integer.

Backend Tier Delay Type This attribute defines the request type based on its delay time in the backend tier. Valid values are Unknown, Good, Fair, and Bad.

Backend Tier Response (msec) The actual response time in milliseconds in the backend tier collected during the interval. The valid format is a positive integer.

Client Bad Delay (msec) The delay time (in milliseconds) in the client tier collected during the interval. This column is non-zero value when client delay exceeds the bad threshold configured for it. The valid format is a positive integer.

Client Fair Delay (msec) The delay time (in milliseconds) in the client tier collected during the interval. This column is non-zero value when client delay exceeds the fair threshold configured for it. The valid format is a positive integer.

Client Good Delay (msec) The delay time (in milliseconds) in the client tier collected during the interval. This column is non-zero when client delay is less than fair threshold configured for it. The valid format is a positive integer.

Client Tier Delay Type This attribute defines the request type based on its delay time in the client tier. Valid values are Unknown, Good, Fair, and Bad.

Client Tier Response (msec) The actual response time in milliseconds in the client tier collected during the interval. The valid format is a positive integer.

Completion Count The number of requests that successfully completed during the interval. The valid format is a positive integer.

Custom Request Count The number of custom requests. The valid format is a positive integer.

Custom Request Time (ms) The average time (in milliseconds) the custom requests spent. The valid format is a positive integer.

Custom Request Time (%) The percentage of time the custom requests spent. The valid format is a decimal (formatted to one decimal place).

EJB Count The number of times this request invoked an Enterprise Java Bean (EJB) request. The valid format is a positive integer.

EJB Time (ms) The average time (in milliseconds) this request spent processing Enterprise Java Bean (EJB) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

EJB Time (%) The percentage of time this request spent processing Enterprise Java Bean (EJB) requests. The valid format is a decimal (formatted to one decimal place).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

JCA Count The number of times this request invoked a J2EE Connector Architecture (JCA) request. The valid format is a positive integer.

JCA Time (ms) The average time (in milliseconds) this request spent processing J2EE Connector Architecture (JCA) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

JCA Time (%) The percentage of time this request spent processing J2EE Connector Architecture (JCA) requests. The valid format is a decimal (formatted to one decimal place).

JMS Count The number of times this request invoked a Java Message Service (JMS) request. The valid format is a positive integer.

JMS Time (ms) The average time (in milliseconds) this request spent processing Java Message Service (JMS) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

JMS Time (%) The percentage of time this request spent processing Java Message Service (JMS) requests. The valid format is a decimal (formatted to one decimal place).

JNDI Count The number of times this request invoked a Java Naming and Directory Interface (JNDI) request. The valid format is a positive integer.

JNDI Time (ms) The average time (in milliseconds) this request spent processing Java Naming and Directory Interface (JNDI) requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

JNDI Time (%) The percentage of time this request spent processing Java Naming and Directory Interface (JNDI) requests. The valid format is a decimal (formatted to one decimal place).

Longest Response (ms) The maximum time (in milliseconds) it took this request to run during the interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The process identifier of the Java virtual machine. The valid format is a positive integer.

Request Bad Response Threshold (msec) The threshold that defines the bad requests. A request that spends more time to complete than this threshold to complete is a bad request. The valid format is a positive integer.

Request Completion (%) The percentage of the requests that completed successfully during the interval. The valid format is a positive integer.

Request Completion Level The completion level of the requests during the interval. Valid values are Unknown, Good, Fair, and Bad.

Request Count The number of times this request ran during the interval. The valid format is a positive integer.

Request Detail The URI for servlet requests, or the method name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Delay Type The type of the request delay. Valid values are Unknown, Good, Fair, and Bad.

Request Fair Response Threshold (msec) The threshold that defines the fair requests. A request that spends more time than this threshold and less time than the *Request Bad Response Threshold (msec)* attribute to complete is a fair request. The valid format is a positive integer.

Request Bad Delay (msec) The delay time (in milliseconds) collected during the interval. This column is non-zero value when the whole request response time exceeds the bad threshold configured for it. The valid format is a positive integer.

Request Fair Delay (msec) The delay time (in milliseconds) collected during the interval. This column is non-zero value when the whole request response time exceeds the fair threshold configured for it. The valid format is a positive integer.

Request Good Delay (msec) The delay time (in milliseconds) collected during the interval. This column is non-zero value when the whole request response time is less than fair threshold configured for it. The valid format is a positive integer.

Request Label A shortened version of Request Name, used to display the request name in the chart view. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Name The URL for servlet requests, or the fully qualified class name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Rate (per sec) The number of requests completed per second during the interval. If the sampling rate is less than 100%, this number is extrapolated to estimate 100% of completed requests. The valid format is a decimal (formatted to three decimal places).

Request Type The type of request being run. Valid values are Servlet, EJB_Method, Custom, All_Workloads, Unknown, and Portlet.

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 127. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of requests being sampled. The valid format is a positive integer.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet Count The number of times this request invoked a Servlet request. The valid format is a positive integer.

Servlet Time (ms) The average time (in milliseconds) this request spent processing Servlet requests. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

Servlet Time (%) The percentage of time this request spent processing Servlet requests. The valid format is a decimal (formatted to one decimal place).

SQL Connect Count The number of times this request connected to a JDBC database. The valid format is a positive integer.

SQL Connect Time (ms) The average time (in milliseconds) this request spent connecting to a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Connect Time (%) The percentage of time this request spent connecting to a JDBC database. The valid format is a decimal (formatted to one decimal place).

SQL Execute Count The number of times this request executed a JDBC database. The valid format is a positive integer.

SQL Execute Time (ms) The average time (in milliseconds) this request spent executing a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Execute Time (%) The percentage of time this request spent executing a JDBC database. The valid format is a decimal (formatted to one decimal place).

SQL Query Count The number of times this request queried a JDBC database. The valid format is a positive integer.

SQL Query Time (ms) The average time (in milliseconds) this request spent querying a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Query Time (%) The percentage of time this request spent querying a JDBC database. The valid format is a decimal (formatted to one decimal place).

SQL Update Count The number of times this request updated a JDBC database. The valid format is a positive integer.

SQL Update Time (ms) The average time (in milliseconds) this request spent updating a JDBC database. This field can have a zero value if the total time is less than the number of requests, due to truncation. The valid format is a positive integer.

SQL Update Time (%) The percentage of time this request spent updating a JDBC database. The valid format is a decimal (formatted to one decimal place).

Total Time (ms) The total CPU time (in milliseconds) this request consumed during the interval. The valid format is a positive integer.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Requests Monitoring Configuration attributes

The **Requests Monitoring Configuration** attributes provide information for all requests monitored in application. Use these attributes to monitor application edge requests. The agent supports three types of edge requests, Servlet/JSP, EJB, and Portal.

The attributes within this group are used to build the Request Baseline workspace.

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Application ID The unique identifier that is assigned automatically when the application is first configured and is preserved during the whole application life cycle. The valid format is a positive integer.

Application Tier Threshold (msec) The response time threshold in the application tier in milliseconds. The valid format is a positive integer.

Auto Threshold Bad Projection (%) The bad response time projection used for auto threshold. The valid format is a positive integer.

Auto Threshold Fair Ratio The percentage to derive the fair response time threshold from the baseline selection. The valid format is a positive integer.

Auto Threshold Fair Projection (%) The fair response time projection used for auto thresholds. The valid format is a positive integer.

Auto Threshold Mode The request auto threshold mode. Valid values are Default, Custom, and Disabled.

Auto Threshold Deviation (%) The maximum allowed deviation of requests baseline data used for auto threshold. The valid format is a positive integer.

Auto Threshold Percent (%) The minimum percent of requests baseline data used for auto threshold. The valid format is a positive integer.

Backend Tier Threshold (msec) The response time threshold in the backend tier in milliseconds. The valid format is a positive integer.

Bad Response Threshold (msec) The time (in milliseconds) that defines the bad requests. A request that spends more time than this threshold to complete is a bad request. Use this attribute with `Fair Response Threshold (msec)` attribute and `Fair Response Zone (msec)` attribute. The valid format is a positive integer.

Bad Errors Rate Threshold The value of bad error rate percentage. The valid format is a positive integer.

Client Tier Threshold (msec) The response time threshold in the client tier in milliseconds. The valid format is a positive integer.

Fair Response Threshold (msec) The time (in milliseconds) that defines the fair requests. A request that spends less time than this threshold to complete is a good request. Use this attribute with `Fair Response Zone (msec)` attribute and `Bad Response Threshold (msec)` attribute. The valid format is a positive integer.

Fair Response Zone (msec) The time span (in milliseconds) that defines the fair requests. This time span is between the fair response time threshold and the bad time threshold. If the response time of a request falls into this time span, the request is a fair request. Use this attribute with `Fair Response Threshold (msec)` attribute and `Bad Response Threshold (msec)` attribute. The valid format is a positive integer.

Fair Errors Rate Threshold The value of fair error rate percentage. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Detail The request detail name. The valid format is an alphanumeric string, with a maximum of 256 characters.

Request ID The unique identifier of the request that belongs to the application. The valid format is a positive integer.

Request Label A shortened version of Request Name, used to display the request name in the chart view. The valid format is an alphanumeric string, with a maximum of 24 characters.

Request Name The URL for servlet requests, or the fully qualified class name for EJBs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Request Type The type of request being run. Valid values are All, Servlet/JSP, EJB, and Portal.

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 128. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Request Times and Rates - J2EE attributes

The **Request Times and Rates - J2EE** attribute group provides information about historical request throughput and average response time for a particular application server.

The attributes within this group are used to build the J2EE application servers workspaces.

Application Name The name of the application to which the request belongs. The valid format is an alphanumeric string, with a maximum of 256 characters.

Average Load The average number of concurrent requests during the interval. The valid format is a decimal (formatted to three decimal places).

Average Request Completion Rate The average request completion rate (that is, the request throughput). If the sampling rate is less than 100%, this number is extrapolated to estimate 100% of completed requests. The valid format is a positive integer.

Average Request Response Time (ms) The average request response time, in milliseconds. The valid format is a positive integer.

Error Rate (%) The error rate of the request during the interval. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Count The number of times this request ran during the interval. The valid format is a positive integer.

Request Type The type of request being run. Valid values are Servlet, EJB_Method, Custom, All_Workloads, Unknown, and Portlet.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 129. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of edge requests-such as servlets and JSPs that were sampled during the interval. The valid format is a positive integer.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total (ms) The total time. The valid format is a decimal (formatted to three decimal places).

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Selected Request - J2EE attributes

The **Selected Request - J2EE** attribute group provides detailed information about transaction requests for database (that is, JDBC), messaging (that is, JMS), or EIS (that is, JCA) services.

The attributes within this group are used to build these workspaces:
- "Selected Request - data sources workspace" on page 659
- "Selected Request - JMS Queues workspace" on page 660
- "Selected Request - Resource Adapters workspace" on page 660

Note: The attributes within this attribute group contain meaningful values only if your site has set the request data monitoring level to Level2 to collect data on nested requests.

Activity Category The type of request. Valid values are n/a (not applicable), JDBC, JMS, and JCA.

Activity Detail Detailed information about the activity performed by the selected request, for example, the SQL statement being processed. The valid format is an alphanumeric string, maximum 128 characters.

Activity Label An abbreviated version of Activity Name, used to display the activity name in the chart view. The valid format is an alphanumeric string, with a maximum of 128 characters.

Activity Name The resource that the request is accessing, for example, the data source name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Activity Type The type of the resource being requested. Valid values display in the following table:

Table 130. Activity types

Type	Definition
n/a	not applicable
Servlet	A call to the doGet or doPost methods of a server
EJB_Method_Call	A call to a business method for an EJB class
Obtain_SQL_Connection_from_Datasource	A call to obtain a connection from a JDBC data source
SQL_Query	A Query request to a JDBC data source

Table 130. Activity types (continued)

Type	Definition
SQL_Update	An Update request to a JDBC data source
SQL_Other	Any other request to a JDBC data source
JMS_Message_Browse	A call to browse a message from a JMS queue
JMS_Message_Get	A call to receive a message from a JMS queue (that is, a destructive get)
JMS_Message_Put	A call to put a message from a JMS queue
JMS_Publish_Message	A call to publish a publication to a JMS queue
JCA_CCI_Execute_interaction	A request by a J2EE application to execute a JCA interaction (a JDBC, JMS, or other JCA-supported operation) against a backend system
JNDI_Lookup	A call to JNDI to build an Initial Context or to perform a lookup
Unknown	The activity type cannot be determined

Average Response (ms) The average time (in milliseconds) executing this request, per occurrence. The valid format is a decimal (formatted to one decimal place).

Delay (%) The percentage of execution time this activity consumed on average when processing this request. The valid format is a decimal (formatted to one decimal place).

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Longest Response (ms) The worst-case response time (in milliseconds) experienced by this request. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Occurrences The number of occurrences. The valid format is a positive integer.

Origin Node The name of the application server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The unique identifier of the JVM process (the class ID of the JVM). The valid format is a positive integer.

Request Detail The URI for servlet requests, or the method name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Name The URL for servlet requests, or the fully qualified class name for EJBs. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Type The type of transaction being run. Valid values are Servlet and EJB_Method, Custom, All_Workloads, Unknown, Portlet.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the

STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 131. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Sampling Rate (%) The percentage of edge requests-such as servlets and JSPs-that were sampled for nested requests during the interval. The valid format is a positive integer.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Total Time (ms) The total CPU time (in milliseconds) consumed by this request. The valid format is a positive integer.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Servlets JSPs - J2EE attributes

The **Servlets JSPs - J2EE** attributes collect performance information for servlets and Java server pages (JSPs).

The attributes within this group are used to build the "Servlets/JSPs - Selected Web Application workspace" on page 662.

Average Response Time (ms) The average servlet service time in milliseconds. The valid format is a decimal (formatted to three decimal places). This attribute might be empty if servlet is not invoked yet.

Error Count The number of servlets that are in error. The valid format is a positive integer.

Interval (sec) The length (in seconds) of the interval. The valid format is a positive integer.

Invocation Count The number of invocations. The valid format is a positive integer. This attribute might be empty if servlet is not invoked yet.

Max Time (ms) The longest service time in milliseconds. The valid format is a decimal (formatted to three decimal places).

Min Time (ms) The shortest service time in milliseconds. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 132. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet/JSP Name The name of the web application. The valid format is an alphanumeric string, with a maximum of 128 characters.

Total Time (ms) The total service time in milliseconds. The valid format is a decimal (formatted to three decimal places).

Web Application Name The name of the web application. The valid format is an alphanumeric string, with a maximum of 128 characters.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Servlets and JSPs - WebLogic attributes

The **Servlets and JSPs - WebLogic** attributes provide performance information for servlets and JavaServer pages (JSPs).

The attributes within this group are used to build the "Servlets/JSPs - Selected Enterprise Application workspace" on page 661.

Average Execution Time (ms) The average amount of time (in milliseconds) the invocations of the servlet have executed for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Context Path The context path. The valid format is an alphanumeric string, with a maximum of 256 characters.

Cumulative Average Execution Time (ms) The average amount of time (in milliseconds) all invocations of the servlet have executed since created. The valid format is a positive integer.

Enterprise Application Name The J2EE application name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Execution Time (ms) The amount of time (in milliseconds) all invocations of the servlet has executed since the previous sample. The valid format is a positive integer.

Highest Execution Time (ms) The amount of time (in milliseconds) the single longest invocation of the servlets has executed since created. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Invocations The number of times the servlet has been invoked since the previous sample. The valid format is a positive integer.

Invocation Rate The number of times the servlet has been invoked per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Lowest Execution Time (ms) The amount of time (in milliseconds) the single shortest invocation of the servlet has executed since created. The valid format is a positive integer.

Maximum Pool Capacity The maximum capacity of this servlet for single thread model servlets. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Reloads The number of times the servlet has been reloaded for interval since the previous sample. The valid format is a positive integer.

Reload Rate The number of times the servlet has been reloaded per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 133. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet Name The servlet or JSP name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Servlet Path The servlet path. The valid format is an alphanumeric string, with a maximum of 256 characters.

Web Application Name The web application component name. The valid format is an alphanumeric string, with a maximum of 128 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Container - NetWeaver attributes

The **Web Container - NetWeaver** attributes collect performance information about servlets and Java Server pages (JSPs).

The attributes within this group are used to build the “Web Container workspace” on page 664.

All Requests Count The number of all requests since server startup. The valid format is a positive integer.

Current Http Sessions The number of the currently valid http sessions. The valid format is a positive integer.

Current Security Sessions The number of the currently valid security sessions created for http clients. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Invalidated Http Sessions The number of http sessions invalidated by application. The valid format is a positive integer.

Invalidated Security Sessions The number of security sessions which have been invalidated by application. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 134. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Timed Out Http Sessions The number of http sessions which have timed out. The valid format is a positive integer.

Timed Out Security Sessions The number of security sessions which have timed out. The valid format is a positive integer.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Applications - J2EE attributes

The **Web Applications - J2EE** attributes provide aggregated information for each Web application and for the application server running that application. This

performance data describes all servlets and JSPs deployed to that Web application as well as performance data for all servlets and JSPs running in the application server. Examples include the number of loaded servlets and JSPs and total requests. Use the Web Applications - J2EE attributes to create situations that monitor Web application performance and application server loads.

The attributes within this group are used to build the Web Applications workspace.

Average Response Time The average response time of the application, in milliseconds. The valid format is a decimal (formatted to three decimal places). This attribute might be empty if servlet is not invoked yet.

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Invocation Count The total invocation count for application. The valid format is a positive integer. This attribute might be empty if servlet is not invoked yet.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 135. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet Count The number of servlets in these web applications. The valid format is a positive integer.

Total Time The total time. The valid format is a decimal (formatted to three decimal places).

Web Application Name The name of the Web application. The valid format is an alphanumeric string, with a maximum of 128 characters.

For additional information, see:
- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Applications - WebLogic attributes

The **Web Applications - WebLogic** attributes provide data and aggregated statistics for Web application components.

The attributes within this group are used to build the "Web Applications workspace" on page 663.

Context Root The context root (context path) for the Web application. The valid format is an alphanumeric string, with a maximum of 256 characters.

Current Servlet Sessions The current number of open servlet sessions. The valid format is a positive integer.

Enterprise Application Name The J2EE application name. The valid format is an alphanumeric string, with a maximum of 128 characters.

Highest Servlet Sessions The high water mark of the total number of open sessions in this server. The valid format is a positive integer.

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Row Number The row number. The valid format is a positive integer.

Sample Date and Time The date and time that the monitoring agent for WebLogic Server agent collected data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 136. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data-collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Name The name of the J2EE application server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Servlet Invocation Rate The number of servlet requests per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Session Monitoring Enabled Whether servlet sessions monitoring is enabled. It can be enabled in weblogic.xml. Valid values are Enabled and Disabled.

Servlet Reload Rate The servlets reloads per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Servlet Sessions Opened The number of servlet sessions opened since the previous sample. The valid format is a positive integer.

Servlet Session Creation Rate The number of servlet sessions opened per second for the interval since the previous sample. The valid format is a decimal (formatted to three decimal places).

Status The component status. The valid format is an alphanumeric string, with a maximum of 64 characters.

Web Application Name The web application component name. The valid format is an alphanumeric string, with a maximum of 128 characters.

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Situations for ITCAM Agent for J2EE

ITCAM Agent for J2EE provides 11 predefined situations that you can use to:

- Immediately begin monitoring your J2EE application servers
- Monitor and manage widely dispersed J2EE Application Server resources through localized automation
- Use models for creating your own situations

These predefined situations have an alert status of Critical. When these situations trigger an alert, you can investigate the event by opening its workspace. For example, you can use these situations to monitor a J2EE application server for errors occurring within it or the web applications on your site.

How the situations work

Situations are tests expressed in an IF-TRUE format of system conditions that you want to monitor. The tested value is an ITCAM Agent for J2EE attribute expressed in the form *attribute-group.attribute-name*. Thus, if the specified condition occurs or exists, the situation is true, and an alert is issued.

Avoid using negative values

If you define situations that use a counter or a range of numbers, always provide a threshold or use values in a positive range of numbers. For example, use a greater-than-or-equal-to-zero expression as shown in some of the following predefined situations. Using this expression prevents a situation from falsely tripping. If the ITCAM Agent for J2EE Tivoli Enterprise Monitoring Agent encounters an undefined attribute value, it interprets this value as a negative number and erroneously fires a situation that specified a negative number.

Predefined situations-descriptions and formulas

J2EEServletsJSPsError Monitors the error count for servlets and JSPs invoked by a J2EE Application Server application and issues a Critical condition whenever the count becomes nonzero. The J2SE application server does not support this situation. Its formula is as follows:

If

Servlets_JSPs.Error_Count is greater than 0

then

the situation J2EEServletsJSPsError is true.

J2EEError Monitors the error severity for a single J2EE Application Server and issues a Critical condition whenever that severity is greater than 21. The J2SE application server does not support this situation. Its formula is as follows:

If

Log_Analysis.Severity is greater than 21

then

the situation J2EEError is true.

J2EENotConnected Monitors the connection between the ITCAM for J2EE data collector running in an application server and the ITCAM Agent for J2EE monitoring agent to ensure that the monitoring agent is connected and issues a Critical condition whenever it is not. Its formula is as follows:

If

Application_Server_Status.Status equals 0

then

the situation J2EENotConnected is true.

J2EEOutofHeapSpace Monitors the heap allocation status and issues a Critical condition whenever heap space is exhausted. The formula is as follows:

If

Allocation_Failure.Heap_Status equals 1

then

the situation J2EEOutofHeapSpace is true.

J2EEHighResponseTime Monitors the average request response time and issues a Critical condition whenever that time exceeds 2 seconds. The formula is as follows:

If

Request_Times_and_Rates.Average_Request_Response_Time is greater than 2000

then

the situation J2EEHighResponseTime is true.

J2EEHighCPUPercentUsed Monitors the percentage of the CPU being consumed and issues a Critical condition whenever that time exceeds 80%. The formula is as follows:

If

Application_Server.CPU_Used_Percent is greater than 80

then

the situation J2EEHighCPUPercentUsed is true.

J2EEHighGCTimePercent Monitors the percentage of time being spent by the garbage collector and issues a Critical condition whenever that time exceeds 80%. The formula is as follows:

If

Garbage_Collection_Analysis.Real_Time_Percent is greater than 80

then

the situation J2EEHighGCTimePercent is true.

J2EEAppDiscovered monitors J2EE applications deployed in the application server and issues an Informational alert when a new application is discovered. The monitoring agent checks for new applications each time when it connects to the data collector or when an application is deployed when the data collector is already active. The J2SE application server does not support this situation. The formula is:

If

Application_Monitoring_Configuration.Monitoring_Status equals 0

then

the situation J2EEAppDiscovered is true.

The predefined Take Action command **Start_Baselining** associated with the J2EEAppDiscovered situation enables you to automate the baselining of newly discovered applications.

J2EEAppHealthGood monitors the overall application health and issues an Informational alert when application health is good. The J2SE application server does not support this situation. The formula is:

If

Application_Health_Status.Application_Health equals 1

then

the situation J2EEAppHealthGood is true.

The predefined Take Action command **Set_Appliction_Monitoring** associated with the J2EEAppHealthGood situation lowers the request monitoring level for applications generated alert, and reduces the monitoring overhead.

J2EEAppHealthFair monitors the overall application health and issues a warning alert when application health is fair. The J2SE application server does not support this situation. The formula is:

If

Application_Health_Status.Application_Health equals 2

then

the situation J2EEAppHealthFair is true.

The predefined Take Action command **Set_Appliction_Monitoring** associated with J2EEAppHealthFair situation raises the request monitoring level for applications generated alert, and enables you to collect detailed performance data that helps to pinpoint a bottleneck down to particular application tiers.

J2EEAppHealthBad monitors the overall application health and issues a Critical alert when the application health is bad. The J2SE application server does not support this situation. The formula is:

If

Application_Health_Status.Web_Tier_Health equals 3

then

the situation J2EEAppHealthBad is true.

The predefined Take Action command **Set_Appliction_Monitoring** associated with J2EEAppHealthBad situation increases the request monitoring rate for applications generated alert, and enables you to collect more detailed performance data that helps to collect the most precise data about each application tier health level.

For additional information, see:

"Attributes for ITCAM Agent for J2EE" on page 666

Take Action commands for ITCAM Agent for J2EE

Using the Take Action feature, your interactive Tivoli Enterprise Portal users can enter a command that stops or starts a process at any system in your network where one or more Tivoli Enterprise Monitoring Agents are installed. With the ITCAM Agent for J2EE Take Action commands, you can use the portal interface to start, stop, or recycle a J2EE application server or to control the level of monitoring for the current server.

You can start a Take Action command from a workspace, from the Navigator, from a situation that you create, in an improvised mode, or by recalling a saved Take Action command. For details on using these general features, see the online help for Tivoli Enterprise Portal.

Enable_Auto_Threshold: set threshold parameters

Use the Enable_Auto_Threshold Take Action to set automatic threshold parameters and remove any overrides of the thresholds.

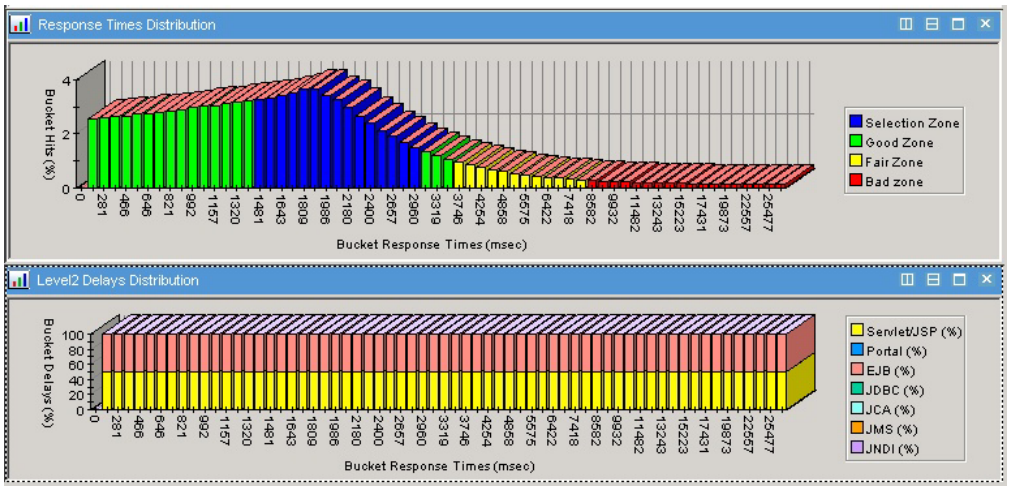
The baselining process supplies statistical information about request response times. ITCAM interprets this information to set automatic thresholds. Several parameters control this interpretation.

The default values for these parameters are sufficient for most cases. However, if the thresholds are not adequate and the baselining process was run recently, you might have to change these parameters. If there are many false alarms or warnings, you must raise the bad or fair threshold. If alarms or warnings are not triggered when needed, you must lower the bad or fair threshold.

Although you can change the parameters for the entire application or for all requests of a given type, usually you do this only for an individual request.

- To set threshold parameters for all requests in the application, select this application in the "Application Health Summary workspace" on page 635 or "Application Registry workspace" on page 636, and select the Enable_Auto_Threshold take action command.
- To set threshold parameters for all requests of a given type in the application, select this request type in the Application Request Configuration table of the "Selected Application - Configuration workspace" on page 656, and select the Enable_Auto_Threshold take action command.
- To set threshold parameters for an individual request, select this request in the "Selected Request - Baseline workspace" on page 655, and select the Enable_Auto_Threshold take action command.

In the "Selected Request - Baseline workspace" on page 655, when you select a line representing a request, you can see the bar charts representing statistical data for this request. This data was gathered during the baselining process. Colors on the bar charts show the way in which the parameters are applied. You can change the parameters using the Enable_Auto_Threshold take action command, and immediately see the effects on the bar charts.



The **Response Times Distribution** chart shows the statistical distribution of response times for this request. To the left are smaller (faster) response times; to the right, larger (slower) ones. The height of every bar shows the percentage of requests that had the indicated response time during the baselining period.

Some bars represent bigger time intervals than others; more bars are devoted to most common response times. For example, if the maximum encountered time is 1000 ms but most response times are 300 - 500 ms, then the first bar might be 0 - 50 ms, but there can also be bars like 305 - 310 ms and 400 - 402 ms.

The blue bars show the zone into which the "typical" response times for this application fall. The green bars show response times that are not "typical", but are below the fair threshold. Response times above the fair threshold but below the bad threshold are shown as yellow bars; for those above the bad threshold, the bars are red.

Use the Enable_Auto_Threshold take action command to set the parameters that affect both the position of the "typical" zone and the way the thresholds are derived from this zone.

For more information about how the bar chart and parameters work, see “Threshold calculation detail” on page 751.

The **Level2 Delays Distribution** chart shows the distribution of time spent in "nested requests" within the requests that had this response time range. Each bar represents a response time of the top-level request (the same as on the top chart). Within this bar, colored sections show how much time is spent within nested requests of different types; the color legend is shown on the bar. ITCAM uses this distribution within the selection zone (that is for typical overall request types) to work out the average share of time taken by each nested request type. When an error or warning arises, ITCAM checks which of the request types takes more than its usual share of time. Based on this, it displays whether the likely cause is the application, backend, or server.

Command syntax

YN:Enable_Auto_Threshold *App_Id Request_Id Auto_Threshold_Percent Auto_Threshold_Deviation Auto_Threshold_Fair_Projection Auto_Threshold_Bad_Projection Use_Default*

Where:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Request_Id
The request ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Auto_Threshold_Percent
Auto_Threshold_Deviation
ITCAM uses these two parameters to calculate the borders of the "typical zone". See “Threshold calculation detail” on page 751.

Auto_Threshold_Fair_Projection
This determines the position of the fair threshold. Increase this parameter to increase the fair threshold; decrease the parameter to decrease the fair threshold. If the parameter is set to 100, the fair threshold is at the right border of the selection zone. For details, see “Threshold calculation detail” on page 751. The bad threshold is not affected.

Auto_Threshold_Bad_Projection
This determines the position of the bad threshold. Increase this parameter to increase the bad threshold; decrease the parameter to decrease the bad threshold. If the parameter is set to 100, the bad threshold is at the right border of the selection zone. For details, see “Threshold calculation detail” on page 751. The fair threshold is not affected.

Use_Default
If set to 0, the auto threshold setting is modified according to the other parameters in this Take Action. If set to 1, the value of the auto threshold settings for this request is taken from the "parent": the values that have been set for the request type, for the entire application, or the ITCAM default values.

Example: YN:Enable_Auto_Threshold 1 12 50 200 150 300 0

Override_Auto_Threshold: override threshold values

Use the Override_Auto_Threshold Take Action to override fair and bad response time threshold values for any request in the application. In this case, while the baselining statistical data is still preserved, ITCAM does not use automatically calculated thresholds.

Do not override threshold values unless you have analyzed the application performance in detail (or were instructed to override threshold values by IBM Level 3 Support). To adjust threshold values without manually overriding them, see “Enable_Auto_Threshold: set threshold parameters” on page 742.

To remove an override, select a request in the “Selected Request - Baseline workspace” on page 655, and select the Enable_Auto_Threshold takes action command. Leave all parameters as they are, in order to use the same auto threshold parameters as were used before the override. If you need to change these parameters, see “Enable_Auto_Threshold: set threshold parameters” on page 742.

Command syntax

YN:Override_Auto_Threshold *App_Id Request_Id Fair_Response_Threshold Bad_Response_Threshold*

Parameters:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Request_Id
The request ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Fair_Response_Threshold
The fair response time threshold, in milliseconds.

Bad_Response_Threshold
The bad response time threshold, in milliseconds.

Recycle_Application_Server: Recycle a J2EE application

Use the Recycle_Application_Server command to stop and then restart an application server. The J2SE application server does not support this take action command.

Command syntax

If invoked from the J2EE Agent entry on the Navigator, the syntax is:

YJ:CycleAppSrv *server_name user password*

Where *server_name* is the J2EE server name, and the *user* and *password* are your own J2EE Application Server identifiers set by the J2EE administrative console (required only if J2EE global security is enabled.)

If, however, this command is invoked from a subnode of the J2EE Agent entry on the Navigator, the syntax is:

YJ: CycleAppSvr *user password*

Where *user* and *password* are your own J2EE Application Server identifiers set by the J2EE administrative console; these are required only if J2EE global security is enabled. (In this case, *server_name* is not required because the subnode name-that is, the server name is already known.)

Remove_J2EE_Application: Remove a J2EE Application

You can use the Remove_J2EE_Application command to remove an application that is in an undeployed state.

Command syntax

If invoked from the WebSphere Agent entry on the Navigator, the syntax is:

YN:Remove Application *&App_Id*

where *&App_Id* is the application ID.

If this command is invoked from a subnode of the J2EE Agent entry on the Navigator, the syntax is:

YN:Remove Application App_Id

In this case you must enter the value for the App_Id where *App_Id* is the application ID.

Remove_J2EE_SubNode: Remove an inactive J2EE application server

Use the Remove_SubNode command to remove a J2EE application server that is no longer active from the Navigator tree.

Command syntax

If invoked from the J2EE Agent entry on the Navigator, the syntax is:

YJ:RemSubNode *server_name*

where *server_name* is the J2EE server (the subnode name).

If, however, this command is invoked from a subnode of the J2EE Agent entry on the Navigator, the syntax is:

YJ:RemSubNode

In this case, the *server_name* value is not required because the subnode name (server name) is already known.

Set_Application_Monitoring: Set monitoring

Use the Set_Application_Monitoring command to set monitoring of the J2EE application. The J2SE application server does not support this take action command.

Command syntax

YJ:Set_Application_Monitoring *App_Id Monitoring_Enabled Request_Data_Monitoring_Level Request_Data_Sampling_Rate*

where *App_Id* is the application ID which is automatically assigned in the portal from the selection context when Take Action was invoked.

Monitoring_Enabled is a Boolean value and the valid values are 0 and 1. It defines whether monitoring agent application dashboard monitoring feature is enabled for the given application.

Request_Data_Monitoring_Level is an integer value that defines custom request monitoring level for the given application. Valid values are 0 (DISABLE), 1 (LEVEL1), and 2 (LEVEL2). This level overrides any level that the user might set for the entire monitoring agent.

Request_Data_Sampling_Rate is an integer value that defines custom request monitoring rate (in percentage) for the given application. Valid values range from 0 to 100.

Set_Completion_Thresholds: Set completion thresholds

Use the Set_Completion_Thresholds command to define the thresholds of the error rate for the J2EE application. The J2SE application server does not support this take action command.

Command syntax

YJ:Set_Completion_Thresholds *App_Id Fair_Completion_Rate Bad_Completion_Rate*

where *App_Id* is the application ID which is automatically assigned in the portal from the selection context when Take Action was invoked.

Fair_Completion_Rate and *Bad_Completion_Rate* are the values in percentage that define thresholds for fair and bad requests completion rates.

Set_Request_Sampling_Rate: Set the sampling rate for request data

Use the Set_Request_Sampling_Rate command to define the percentage of requests to monitor.

Command syntax

YJ:SetRequestSamplingRate *percent*

where *percent* is the percentage of requests you want sampled. Specify a value in the range 1 - 100.

Start_Application_Server: Start a J2EE application server

Use the Start_Application_Server command to start a J2EE application server. The J2SE application server does not support this take action command.

Command syntax

If invoked from the J2EE Agent entry on the Navigator, the syntax is:

YJ:StartAppSrv *server_name user password*

where *server_name* is the J2EE server name, and *user* and *password* are your own J2EE Application Server identifiers set by the J2EE administrative console (required only if J2EE global security is enabled).

If, however, this command is invoked from a subnode of the J2EE Agent entry on the Navigator, the syntax is:

YJ:StartAppSrv *user password*

where *user* and *password* are your own J2EE Application Server identifiers set by the J2EE administrative console; these are required only if J2EE global security is enabled. (In this case, *server_name* is not required because the subnode name-that is the server name-is already known.

Start_Baselining: Start the baselining process

ITCAM can run a *baselining process* for every application. During this process, which runs for a preset period, the data collector collects statistical data on metric

values for a given period. Based on this statistical data, the monitoring agent can automatically set the fair and bad thresholds, as well as the typical breakdown of response times for nested requests. Use the Start Baselining Take Action command to start the baselining process.

When ITCAM begins monitoring an application for the first time, it automatically starts this process for the application. However, with time, average response times can change because of configuration, load pattern, database size, and other issues. You can manually start the baselining process again to take these changes into account. You can also use IBM Tivoli Monitoring (ITM) policies and workflow management to run the baselining process every few months.

As soon as you take the Start Baselining action, the baselining process begins. The thresholds are updated when either the Period or the Update Interval passes.

While the baselining process is running, you can trigger a baseline update to immediately set the thresholds based on the information collected so far.

Command syntax

YN:Start_Baselining *App_Id Period Update_Interval Run_Clean*

Where:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Period
The time for which the baselining process runs. The data collector collects the necessary statistical information for this entire period. When ITCAM starts the baselining process automatically, it sets the period to seven days. The format is `ddd/hh:mm:ss`.

Update_Interval
If you set this parameter to a time interval, ITCAM updates the thresholds according to the information already collected every time this interval passes. For example, when ITCAM starts the baselining process automatically, it sets the update interval to 1 hour. During the seven days that the initial baselining runs, every hour the thresholds are updated according to the statistical data collected so far (for all request types where at least one request was received during the baselining process). The format is `ddd/hh:mm:ss`.

Run_Clean
Set to either 0 or 1. If set to 0, statistical data collected in any previous baselining for the same requests are kept and "amalgamated" with the new data; if set to 1, only the new data is used for setting the thresholds. Normally, you set this to 1.

Note: when this Take Action command is selected for a node representing a z/OS servant region, applies to all servant regions in the same managed system (IBM WebSphere Application Server instance).

Start_GC_Monitoring: Begin reporting garbage-collection data

Use the Start_GC_Monitoring command to activate the display of garbage-collection statistics. This setting is on top of the J2EE Application Server Verbose Garbage Collection value, which must also be active for garbage-collection data to be reported.

Command syntax

YJ:StartGCMonitor

Start_Request_Monitoring : Begin reporting request data

Use the Start_Request_Monitoring command to activate the display of request data.

Command syntax

YJ:StartRequestMonitor *level*

where *level* is the resource-data collection level, either Level1 or Level2. When the collection level is set to Level1, only edge request data-such as servlets and JSPs-are collected; when set to Level2, nested request data (such as JDBC and JMS requests) are also collected.

Important: You can set the monitoring level for any operation separately, using the "Set_Application_Monitoring: Set monitoring" on page 746 Take Action command. The monitoring level from the Start_Request_Monitoring Take Action command is applied by default to newly discovered applications.

Start_Resource_Monitoring: Begin reporting PMI data

Use the Start_Resource_Monitoring command to activate the display of resource (that is, PMI) data. This setting is on top of the J2EE Application Server PMI instrumentation levels, which must also be set for resource data to be reported.

Command syntax

YJ:StartResourceMonitor

Stop_Application_Server: Stop a J2EE application server

Use the Stop_Application_Sever command to stop an application server. The J2SE application server does not support this take action command.

Command syntax

If invoked from the J2EE Agent entry on the Navigator, the syntax is:

YJ:StopAppSvr *server_name user password*

where *server_name* is the J2EE server name, and *user* and *password* are your own J2EE Application Server identifiers set by the J2EE administrative console (required only if J2EE global security is enabled).

If, however, this command is invoked from a subnode of the J2EE Agent entry on the Navigator, the syntax is:

YJ:StopAppSvr *user password*

where *user* and *password* are your own J2EE Application Server identifiers set by the J2EE administrative console; these are required only if J2EE global security is enabled. (In this case, *server_name* is not required because the subnode name-that is, the server name-is already known.)

Stop_Baselining: stop the baselining process

Use the Stop_Baselining Take Action command to immediately stop the baselining process for an application, and recalculate the thresholds based on the request data available up to this point.

Normally you do not have to perform this action. To recalculate the thresholds based on the request data available up to this point, without stopping the baselining process, see "Update_Baseline: trigger a baseline update."

Command syntax

YN:Stop_Baselining *App_Id*

Where:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Note: when this Take Action is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere Application Server instance).

Stop_GC_Monitoring: Stop reporting garbage-collection data

Use the Stop_GC_Monitoring command to end the display of garbage-collection statistics.

Command syntax

YJ:StopGCMonitor

Stop_Request_Monitoring: Stop reporting request data

Use the Stop_Request_Monitoring command to end the display of request data.

Command syntax

YJ:StopRequestMonitor

Stop_Resource_Monitoring: Stop reporting PMI data

Use the Stop_Resource_Monitoring command to end the display of resource (that is, PMI) data.

Command syntax

YJ:StopResourceMonitor

Update_Baseline: trigger a baseline update

If the baselining process is running, the thresholds are set automatically when either the Period or the Update Interval passes. For the initial baselining process,

the first automatic update happens after one hour. With the Update_Baseline Take Action command, you can force ITCAM to update the thresholds immediately, based on the information collected so far. This can be useful if you do not want to wait for the periodic automatic update. Once the automatic update time comes, the threshold is updated again.

If a baselining process is not running for the application, an error is raised. If no requests of a given request type have been received since the baselining process has started, the update does not affect this request type.

Command syntax

YN:Update_Baseline *App_Id*

Where:

App_Id
The application ID, automatically assigned in the portal from the selection context when Take Action was invoked.

Note: When this Take Action is selected for a node representing a z/OS servant region, it applies to all servant regions in the same managed system (IBM WebSphere Application Server instance).

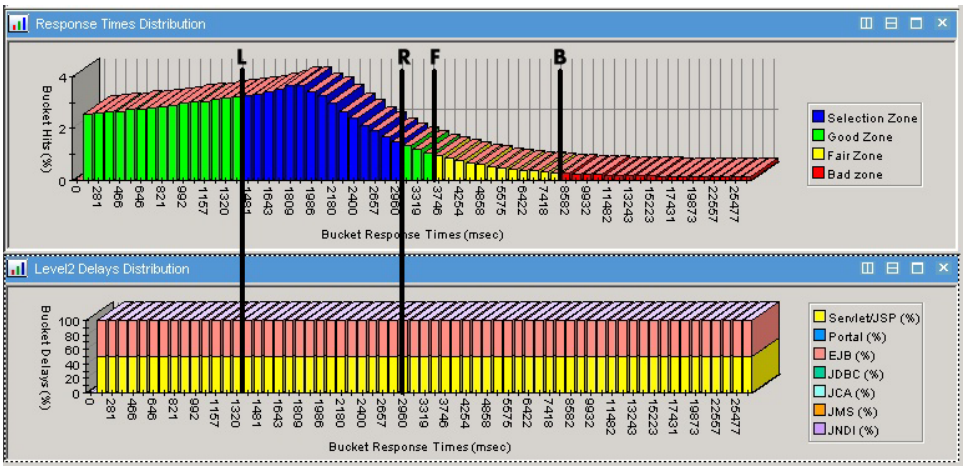
Threshold calculation detail

If you want to set parameters that affect the calculation of the automatic response time thresholds, you might need to know the details of this calculation.

ITCAM processes the baselining statistical data and applies the auto threshold parameters in the following way. The parameters are set in the Enable_Auto_Threshold take action command, see "Enable_Auto_Threshold: set threshold parameters" on page 742.

1. The response time results are sorted into up to 64 "buckets", from zero to the maximum response time encountered for this request. The buckets do not represent equal time intervals; for response time regions where most of the "hits" fall, the buckets represent smaller intervals. For example, if the maximum encountered time is 1000 ms but most response times are 300 - 500 ms, then the first bucket might be 0 - 50 ms, but there can also be buckets representing response times of 305 - 310 ms and 400 - 402 ms. ITCAM distributes the bucket borders so that the largest number of hits in any one bucket are not more than three times the number of hits in the smallest bucket.

   ITCAM calculates the percentage of the total amount of requests that fall into each response time bucket, and divides it by the time interval width represented by the bucket. This is shown on the **Response Times Distribution** bar chart in the "Selected Request - Baseline workspace" on page 655.



Each bar represents a bucket. The bar height shows the percentage of the requests in this bucket. All the subsequent calculations are rounded up to buckets.

2. ITCAM determines the selection zone, which contains the "typical" response time values. This zone is represented by the blue bars on the chart.

   ITCAM finds the response time interval (left border L to right border R) where the following factors are true:

   - The percentage of hits that fall into this interval is no less than the Auto_Threshold_Percent parameter.
   - The spread of the time interval, calculated as (R/L)*100 - 100, is not greater than the Auto_Threshold_Deviation parameter.

     Important: The Auto_Threshold_Deviation parameter does not denote the statistical definition of deviation.

   If several zones match these criteria, ITCAM chooses the one where the following value is the greatest: S/(R-L), where S is the total number of hits that fell into this zone.

   If a zone where both requirements are true cannot be found at all, ITCAM first determines the interval where the percentage of hits is not less than the Auto_Threshold_Percent parameter while the spread is as little as possible. Then, within this interval, it finds a zone where the spread is not greater than the Auto_Threshold_Deviation parameter and the percentage of hits is as large as possible.

   ITCAM determines the typical nested request times from the nested request times in this zone, shown on the **Level2 Delays Distribution** chart.

3. Finally, ITCAM calculates the thresholds.

   The fair threshold is determined using the left and right borders of the selection zone and the Auto_Threshold_Fair_Projection parameter:

   F = L + ((R-L) * Auto_Threshold_Fair_Projection / 100)

   The bad threshold is calculated in the same way, using the Auto_Threshold_Bad_Projection parameter:

   B = L + ((R-L) * Auto_Threshold_Bad_Projection / 100)

   Example: the left border of the selection zone can be L=1450 ms, and the right border R=3000 ms. By default, Auto_Threshold_Fair_Projection=150, and Auto_Threshold_Bad_Projection=300. In this case:

   - The fair response threshold is F = 1450 + ((3000-1450) * 150 / 100) = 3775 ms

- The bad response threshold is B = 1450 + ((3000-1450) * 300 / 100) = 6100 ms

Glossary for ITCAM Agent for J2EE

A

administrative console A graphical interface that guides the user through systems administration tasks such as deployment, configuration, monitoring, starting, and stopping applications, services, and resources.

C

connection pool A group of host connections that are maintained in an initialized state, ready to be used without having to create and initialize them.

CPU Central Processing Unit. The part of the computer that controls the interpretation and execution of instructions.

CTG CICS Transaction Gateway. CTG integrates the J2EE Application Server applications of your site with key mature business systems running on CICS servers. CTG integrates middle-tier application servers with CICS by providing a multiuser gateway that supports Web applications written in Java, C/C++, Visual Basic, and COBOL. CTG supports three programming interfaces: the External Call Interface (ECI), the External Presentation Interface (EPI), and the External Security Interface (ESI). The latest version of CTG implements the J2EE Connector Architecture (JCA), which allows enterprise Java technology to exploit CICS applications.

D

data sources Data pertaining to JDBC data streams. These are logical connections to database subsystems.

DCE Distributed Computing Environment. A specification from the Open Software Foundation (OSF) that supports remote function execution across a network. J2EE Application Server uses DCE to provide certain types of security, for example, authentication of tokens, tickets, or credentials in an untrusted network.

design patterns Written narratives that define a recurring problem, outline a solution, and describe the tradeoffs involved in implementing that solution. In object-oriented programming environments like Java, a design pattern also describes certain objects and object classes to be used when designing such a solution.

DHCP Dynamic Host Configuration Protocol. A communications protocol that lets network administrators centrally manage the assignment of Internet Protocol (IP) addresses across an organization network. Without DHCP, each computer IP address must be entered manually at each computer; if computers move to another location in another part of the network, a new IP address must be chosen. DHCP lets a network administrator supervise and distribute IP addresses from a central point and automatically sends a new IP address when a computer is plugged into a different location in the network.

dynamic cache A consolidation of several caching activities, including servlets, Web services, and J2EE commands into one service where these activities work together to improve performance and share configuration parameters.

E

EJB Enterprise Java Bean. A component architecture for the development and deployment of object-oriented, distributed, enterprise-level applications. Applications written using the Enterprise JavaBeans architecture are scalable, transactional, multi-user, and secure.

EJB module A software unit that consists of one or more enterprise beans and an EJB deployment descriptor.

G

garbage collection A facility of the JVM that reallocates and frees up memory lost when objects are created and later destroyed. Garbage collection usually occurs when insufficient heap space is available for allocating new objects, although you can trigger it manually by invoking the System.gc method.

Garbage collection can be expensive since it involves identifying all the unreferenced objects, removing them from the heap, and then possibly compacting the heap.

H

heap A memory storage area known formally as the System Resources area that stores system information about running applications. The main storage area used for allocating Java classes and object instances, as well as executable code. The allocation of objects on the heap creates a live or active object. The object exists as long as the JVM maintains a pointer to it; when all references to an object cease to exist, it is subject to garbage collection.

hover help A form of online help that can be viewed by moving a cursor over a GUI item such as an icon or field.

HTTP Hypertext Transfer Protocol. A suite of protocols for the Internet that transfer and display hypertext documents.

HTTP sessions Data related to sessions of specific World Wide Web browsers.

I

IIOP Internet Inter-ORB Protocol. A protocol that distributed programs written in different programming languages can use to communicate over the Internet. IIOP is a part of the Common Object Request Broker Architecture (CORBA). IIOP is based on the client/server computing model, in which a client program makes requests of a server program that waits to respond to client requests. With IIOP, you can write client programs that communicate with the existing server programs of your site wherever they are located. You can do this without having to understand anything about the server other than the service it performs and its address (called the Interoperable Object Reference, IOR, which comprises the server port number and IP address).

instrumentation The process of modifying program code so that the time it takes the code to run, along with other statistics, can be measured. These modifications insert hooks into these Java methods so that the real time and CPU time spent executing the methods can be measured. Instrumentation is key to certain

data-collection features (such as lock analysis) of IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE.

IBM Tivoli Monitoring platform The software architecture and foundation that support the development and operations of Tivoli Enterprise Portal and its Tivoli Enterprise Monitoring Agents.

J

J2EE Java 2 Platform, Enterprise Edition. An environment for developing and deploying enterprise applications using the Java language. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functions needed for developing multi-tiered, Web-based applications.

JAR file A Java archive file, which is a compressed file containing many class files along with their directory structure.

JavaBean The Java implementation of component-based software architecture, which defines how small, tested software entities called components can be integrated to build complete software solutions (called containers).

JCA J2EE Connector Architecture. A template for writing your own connectors. The template is a binding between a J2EE application server and a backend enterprise information system (EIS) or data repository such as CICS, IMS, or DB2.

JDBC Java Database Connectivity. An application programming interface (API) for connecting Java programs to the data in a relational database. With this API, you can encode data requests in Structured Query Language (SQL) that JDBC then passes to the database manager for interpretation and processing.

JMS Java Message Service. A programming interface that connects Java programs to middleware messaging applications, for example, J2EE MQ.

JMX Java Management Extensions. A standard technology for management and monitoring of Java applications and Java environments that defines architecture, application programming interfaces (APIs), and services for application and network management. JMX provides tools for building distributed, Web-based solutions for managing and monitoring Java devices and applications.

A JMX agent is an MBean container.

JNDI Java Naming and Directory Interface. A programming interface that connects Java applications to naming and directory servers such as LDAP.

JSP Java server page. A Web page that specifies one or more servlets whose execution on the Web server modifies the page content or appearance before it is presented to the user.

JTA Java Transaction API. An application programming interface developed by Sun Microsystems that specifies standard Java interfaces between a transaction manager (such as Tuxedo) and other players in a distributed transaction system: the resource manager, the application server, the transactional applications, and often the database manager as well.

JVM Java virtual machine. A software implementation of a central processing unit (CPU) that runs Java applets and applications. The JVM provides a software execution engine that safely and compatibly executes the byte codes in Java class files on various microprocessors, whether embedded in a computer or in another electronic device.

JVMPI Java Virtual Machine Profiling Interface. An experimental interface that provides profiling hooks into the Java virtual machine. This interface supports two-way function calls between the JVM and a profiler agent: the virtual machine notifies the agent of various events, and the agent issues requests for control information.

K

Kerberos A security system from the Massachusetts Institute of Technology that provides security services for networking.

L

LDAP Lightweight Directory Access Protocol. A protocol that uses TCP/IP to access directories. Applications can use these directories for storing common data and for retrieving data. For example, applications can use LDAP to access such information as email address, service configuration parameters, or public keys.

lock A semaphore created through the use of the Java synchronized keyword that prevents simultaneous access to a Java object or section of code.

M

MBeans Managed Beans. A set of standard instrumentation methods for use in Java programs and by Java management applications (such as IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE) that allows monitoring and management of Java-based applications. An MBean is a JavaBean that represents a JMX-manageable resource.

monitor 1) A transaction environment for maintaining large quantities of data in a consistent state and that controls which users and clients can access data through authorized servers. 2) A programming primitive created so multiple program threads can share the same resource (such as an object). A program creates a monitor for a given resource by requesting it from the system; the system returns a unique ID for that monitor. After that, any thread needing the resource must use the monitor to lock the resource while the thread is using it. If the monitor is already locked, a thread requesting the resource is queued by the system and then given control when the monitor becomes unlocked. Also called a mutex.

MVC The Model/View/Controller design pattern is based on the separation of the application into three objects: the user interface ("view"), the logical structure of the data the application requires ("model"), and the user commands that change the view or the model ("controller"). Design patterns themselves are written narratives that define a recurring problem, outline a solution, and describe the tradeoffs involved in implementing that solution (in object-oriented programming environments like Java, a design pattern also describes certain objects and object classes to be used when creating such a solution).

N

node Any managed system, such as a J2EE application server, that the Tivoli Enterprise Portal is monitoring. A node can also be a managed system of subnodes being managed under a primary node.

O

OLT Object Level Trace. A tool for testing distributed applications that include a graphical trace facility and a remote debugger.

ORB Object Request Broker. In object-oriented programming, software that serves as an intermediary by transparently enabling objects to exchange requests and responses.

P

PMI Performance Monitoring Infrastructure. The application programming interface (API) that IBM provides for extracting J2EE Application Server performance data. PMI must be enabled and the appropriate instrumentation levels set for the Tivoli Enterprise Monitoring Agent to extract PMI data and populate certain workspaces. PMI runs as an interface to JMX.

portal A Web-based application that commonly provides personalization, single signon, and content aggregation from different sources, and hosts the presentation layer of Information Systems (IS). Portals are becoming the next-generation desktop, delivering e-business applications over the Web to many types of client devices.

portlet A Java technology-based Web component, managed by a portlet container, that process requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to various heterogeneous Enterprise Information Systems (EIS) or Information Systems (IS).

R

request The entry point into an application whose processing or response time directly affects the response time the end user perceives. In IBM Tivoli Composite Application Manager for Application Diagnostics Agent for J2EE, a request is a servlet, JSP, or EJB method. Also called a transaction.

request analysis A configuration technique for a request (that is, a transaction) within J2EE Application Server that allows you to see how often the request runs and what factors about the request degrade its performance. Request analysis involves collecting and aggregating request data on each request that runs. Its primary purpose is to report on requests running in the application server, how often they run, their response time, and what is causing performance degradation.

Request analysis works by dynamic object code instrumentation, which modifies the Java object code for the application' s classes as they are being loaded into the JVM.

resource adapter An implementation of the J2EE Connector Architecture Specification that provides applications with access to resources outside of the server or provides server applications with access to an Enterprise Information System (EIS). A resource adapter can provide applications access to resources such

as DB2, CICS, SAP, and PeopleSoft. It can provide an EIS with the ability to communicate with message-driven beans configured on the server. A resource adapter is implemented in a resource adapter archive file, which has an extension of .rar. A resource adapter can be provided as a stand-alone adapter or as part of an application, in which case it is referred to as an embedded adapter.

RMI Remote Method Invocation. A Java standard from Sun Microsystems that performs a remote procedure call (RPC) to allow Java objects stored in the network to be accessed remotely in a client/server environment.

RPC Remote Procedure Call. A protocol based on the Distributed Computing Environment (DCE) of the Open Software Foundation. DCE allows one program to request services from a program running on another computer in a network. (A procedure call is also known as a function call or a subroutine call.) RPC uses the client/server model: the requesting program is the client, and the responding program is the server. As with a local procedure call, an RPC is a synchronous operation: the requesting program is suspended until the remote procedure returns its results.

S

servlet A Java application that runs in an application server or Web server, and that provides server-side processing.

session A series of requests to a servlet that originate from the same user at the same browser. Sessions allow applications running in a Web container to track individual users.

signature The name of an operation or method and its parameters.

SSL Secure Sockets Layer. A security protocol for communication privacy to provide secure client-server conversations.

struts A widely used framework for developing servlet- or JSP-based Web applications that is distributed as open source by the Apache Jakarta Project. Struts encourages application architectures based on the Model two approach, a variation of the Model-View-Controller (MVC) design paradigm.

T

thread A dispatchable unit of work.

thread pool The threads that are being used by or are available to a computer program.

transaction A single execution of a workload.

U

URI Uniform Resource Identifier. An identifier for a point of content on the Internet, be it a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form of URI called a Universal Resource Locator (URL). A URI typically describes the mechanism used to access the resource, the computer that houses the resource, and the name of the resource (such as a file name) on that computer.

URL Universal Resource Locator. The unique address for a file accessible via the Internet. Such a file might be a Web page (usually the home page), an image file, or a program such as a Java applet or servlet. The URL comprises the protocol used to access the file, a domain name that identifies a specific computer on the Internet, and a path name that specifies that file location on that computer.

W

Web applications The runtime environment for dynamic Web applications. A J2EE Web application server implements the services of the J2EE standard. It provides communications, resource management, security, transaction management, and persistence capabilities for Web applications. It also typically includes an administration interface for managing the server and deployed applications.

Web container A special J2EE container that manages Web applications in a J2EE architecture. A Web container specifies a runtime environment for Web components, including security, concurrency, life cycle management, transaction, deployment, and other services; it also enables a Web application to access external resources such as relational databases (via JDBC) and Enterprise Java Beans. A Web container provides the same services as a JSP container and provides support for managing Java Server Pages.

Every J2EE server contains at least one Web container.

Chapter 6. ITCAM Agent for HTTP Servers

IBM Tivoli Composite Application Manager Agent for HTTP Servers provides a Systems Management solution for the web servers for distributed platforms. Using ITCAM Agent for HTTP Servers, you can monitor multiple web servers running on the same physical node.

The Tivoli Enterprise Monitoring Agent collects performance data using the following methods:

- Apache servers parse the config file to get the server name and collect data by the module. The module handles all the HTTP requests.
- Sun web servers collect data by polling SNMP service for web server statistics and parsing web server configuration files to get information not provided by the SNMP subagent.

Attributes within ITCAM Agent for HTTP Servers collect data about the inner workings of a web server and performance information about user applications running under its control.

For additional usage information about this agent, see:

- Workspaces
- Attributes
- Situations
- Take Action commands

Workspaces for ITCAM Agent for HTTP Servers

As part of the integration of IBM Tivoli Composite Application Manager Agent for HTTP Servers with the Tivoli Enterprise Portal, the workspaces show views of monitoring data that provide detailed current data about the web servers running on the UNIX and Windows platforms of your site. In addition to reports and graphs, a workspace can contain other views (that is, windows), such as a Request Rate - History view, or a Take Action view from which you can issue commands.

Several views of high-level information

Several workspaces provide high-level information to help you meet the monitoring and administrative needs of your site. These workspaces report on status and availability for both the web server administrative server and its web server instances. You can use the workspaces to easily monitor the availability of your enterprise, the web servers, and web server instances.

Available Tivoli Enterprise Portal workspaces

For an overview of the organization of the available workspaces, see Organization of the predefined workspaces.

Organization of the predefined workspaces

The IBM Tivoli Composite Application Manager Agent for HTTP Servers workspaces for the Tivoli Enterprise Portal define data displays that display in the Navigator Physical view.

© Copyright IBM Corp. 2006, 2015

Workspace organization

The hierarchy levels shown in the Navigator depend upon the enterprise customization of the Tivoli Enterprise Portal. However, ITCAM Agent for HTTP Servers does provide a set of predefined workspaces, which do not require customization. The following list shows the order and hierarchy of the predefined workspaces provided by the ITCAM Agent for HTTP Servers Tivoli Enterprise Monitoring Agent. It is a representation of how the predefined workspaces are organized in the Navigator. For more detailed information about a workspace, click its name in this list.

operating system [for example, Windows]
- *system* [that is, nodename]
  - "web server Agent workspace" on page 765
    - "Apache web server workspace"
      - "Apache websites workspace" on page 763
    - "Sun Java System web server workspace" on page 764
      - "Sun websites workspace" on page 764

For additional information, see: Attribute groups used by the predefined workspaces

Apache web server workspace

This workspace shows the Apache web server information, including the summary rates over all websites (virtual hosts).

This workspace displays data provided by the "Apache web server attributes" on page 767.

The predefined workspace contains the following items:
- Request Rate - History graph, which shows the historical rate at which HTTP requests were made per second
- Server Failure Rate - History graph, which shows the historical rate at which server internal errors occurred per minute
- Transfer Rate - History graph, which shows the number of kilobytes received and sent by the web server per second
- Failed Request Rate - History graph, which shows the historical number of failed requests per minute
- Server Summary report, which displays summarized information about the Apache web server, including server status, request rate, kilobytes rate, and login rate

Accessing the Apache web server workspace

To access this workspace, complete the following steps:
1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the node list of monitored applications, expand the list of web server agents.
4. Within the list of available agents, click the **Apache Web Server** entry of your choice.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Apache websites workspace

This workspace shows the list of the websites (virtual hosts) that are configured for the Apache web server along with their status. Clicking the link in the Selected Web Site view shows data for the selected website in the same workspace. By default the website listed in first row in the websites table is presented in the Apache websites workspace.

There are five views in the Apache websites workspace:

- Selected Web Site
- Request Rate - History
- Transfer Rate - History
- Pages Failed Rate - History
- Failed Logins Rate - History

The Selected Web Site view displays all the web sites by default. Where as the four other views can only display the data for the first web site listed in the Selected Web Site view. This is a product limiataion.

This workspace displays data provided by the "Apache websites attributes" on page 768.

The predefined workspace contains the following items:

- Request Rate - History graph, which shows the historical rate at which HTTP requests were made, per second, for the selected Apache website
- Transfer Rate - History graph, which shows the number of kilobytes, per second, received, and sent by the selected web service
- Pages Failed Rate - History graph, which shows the number of requests per minute that were not satisfied by the server because the requested document was not found or is forbidden. This rate applies to the collection interval for a selected Apache website
- Failed Login Rate - History graph, which shows the historical number of failed logins per minute for the selected Apache website
- Websites report, which displays the status of each Apache website
- Selected website report, which displays aggregated information about all Apache websites

Accessing the Apache websites workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the node list of monitored applications, expand the list of web server agents.
4. Within the list of available agents, expand the web server of your choice.

5. Within the list of available web server workspaces on that server, click the **Apache Web Sites** entry. By default the website listed in first row in the websites table is presented in the Apache websites workspace.

Note:

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Sun Java System web server workspace

This workspace shows the Sun Java System web server information, including the summary rates over all websites (virtual servers).

This workspace displays data provided by the "Sun web server attributes" on page 770.

The predefined workspace contains the following items:

- Requests per second - History graph, which shows the rate, per second, at which requests have been processed by the virtual server
- Kilobytes per second - History graph, which shows the aggregated number of kilobytes received and sent per second
- Server Error per minute graph, which shows the rate at which the number of 500-level (Server Error) responses issued (per minute) by the web server during the collection interval
- Connection Queue Count - Which shows the number of connections currently in the web server connection queue
- Server Summary report, which displays the summarized information about the Sun Java System web server

Accessing the Sun Java System web server workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications on that node, expand the list of web server agents.
4. Within the list of available agents, click the **Sun Java System Web Server** entry of your choice.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Sun websites workspace

This workspace shows the list of the Sun websites (virtual servers) configured for the server along with their status. Clicking the link in the website table shows data for the selected website in the same workspace.

This workspace displays data provided by the "Sun websites attributes" on page 773.

The predefined workspace contains the following items:

- Request per second - History graph, which shows the rate, per second, at which requests have been processed by the server
- Pages not Found per minute - History graph, which shows the number, per minute, of 404-level responses (Pages Not Found) issued by the virtual server during the collection interval
- Kilobytes per second - History graph, which shows the aggregated number of kilobytes received and sent per second
- Failed Logins per minute - History graph, which shows the historical number of 401-level responses (Failed Login) issued, per minute, by the virtual server during the collection interval
- Websites report, which displays the status of each Sun website
- Selected website report, which displays aggregated information about all the Sun websites

Accessing the Sun websites workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications on that node, expand the list of web server agents.
4. Within the list of available agents, expand the web server of your choice.
5. Within the list of available web server workspaces on that server, click the **Sun websites** entry.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

web server Agent workspace

This workspace shows the status on all monitored web servers and the web server agent events.

This workspace displays data provided by both the "HTTP Servers Agent Events attributes" on page 776 and the "Web Servers Status attributes" on page 777.

The predefined workspace contains the following items:

- Web Servers Summary report, which displays the overall status of web servers, including server type, server name, server status, uptime, and process ID
- Agent Events report, which displays information about web server agent events, including severity, message ID, and message description

Accessing the web server Agent workspace

To access this workspace, perform the following steps:

1. Within the Navigator, expand Windows Systems, Linux Systems, or UNIX Systems, as appropriate for the node you are monitoring.
2. Within the node list, expand the entry that corresponds to your node name.
3. Within the list of monitored applications on that node, click **Web Server Agent**.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Attributes for ITCAM Agent for HTTP Servers

IBM Tivoli Composite Application Manager Agent for HTTP Servers is a Tivoli Enterprise Management Agent that is located within your distributed systems. This agent gathers data about web server processes that are running, and stores this data in elements called attributes. Each attribute is a characteristic of an object. For example, the Kilobytes Rate (per second) attribute in the Apache web server attribute group reports the rate at which kilobytes were sent and received, per second, during the collection interval.

Attribute groups

The ITCAM Agent for HTTP Servers attributes are organized into groups of related items. These attribute groups comprise the attribute tables for this agent.

Attributes and workspaces

Within the Tivoli Enterprise Portal workspaces, these attributes are displayed in, and correspond to, the columns in the reports and the items in the graphic displays for charts and graphs. You can use the collected data to analyze and monitor the performance of your web servers and the applications running within them. For an overview of the correlations between the predefined workspaces and the attribute groups, see Attribute groups used by the predefined workspaces.

Attributes and situations

Various attributes are referenced by the predefined situations of the product. You can also use the ITCAM Agent for HTTP Servers attributes to create your own situations to monitor the performance of your web servers and their applications. These situations can monitor your web server resources or correlate multiple conditions to alert you to problems that might occur when attribute values exceed thresholds defined by you.

Attribute groups used by the predefined workspaces

A workspace contains graphical data or report columns that correspond directly to particular attributes in an attribute group. The table shows the correlations between the predefined workspaces and the attribute groups. The workspaces are listed alphabetically, not in the order in which they are displayed in the Navigator.

Table 137. Workspaces and the attribute groups they reference

Workspace	Related Attribute Groups
"Apache web server workspace" on page 762	"Apache web server attributes" on page 767
"Apache websites workspace" on page 763	"Apache websites attributes" on page 768
"Sun Java System web server workspace" on page 764	"Sun web server attributes" on page 770
"Sun websites workspace" on page 764	"Sun websites attributes" on page 773
"web server Agent workspace" on page 765	"Web Servers Status attributes" on page 777 "HTTP Servers Agent Events attributes" on page 776

Apache web server attributes

The **Apache Web Server** attributes provide status information about the Apache web server.

The attributes within this group are used to build the "Apache web server workspace" on page 762.

Configuration File The fully qualified path of the Apache HTTP Server configuration file name. The valid format is an alphanumeric string, with a maximum of 256 characters.

Failed Login Rate (per min) The average number of failed logins that occurred, per minute. The valid format is a decimal (formatted to three decimal places).

Failed Requests Rate (per min) The average number (per minute) of failed requests. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length of the interval in seconds. The valid format is a positive integer.

Kilobytes Rate (per sec) The number of kilobytes that are sent and received per second. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum 256 of characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The identifier of the Apache Server process. The valid format is a positive integer.

Request Rate (per sec) The rate at which HTTP requests were made. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 138. Format of the 12–character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Failures Rate (per min) The average number of internal server errors. The valid format is a decimal (formatted to three decimal places).

Start Date and Time The date and time when the web server started. The valid format is a timestamp. This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the Apache HTTP server. Valid values are Not_Running, Running, and Error.

Successful Login Rate (per min) The average number of successful logins that occurred per minute. The valid format is a decimal (formatted to three decimal places).

Version The version of the Apache web server. The valid format is an alphanumeric string, with a maximum of 64 characters.

Web Server Name The name of the web server. The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Apache websites attributes

The **Apache Web Sites** attributes provide status and performance information about the Apache websites workspace.

The attributes within this group are used to build the "Apache websites workspace" on page 763.

Failed Login Rate (per min) The average number of failed logins that occurred per minute. The valid format is a decimal (formatted to three decimal places).

Failed Pages Rate (per min) The rate (per minute) of pages not found or forbidden. The valid format is a decimal (formatted to three decimal places).

Failed Request Rate (per min) The average number, per minute, of failed requests made to the server. The valid format is a decimal (formatted to three decimal places).

Interval (sec) The length (in seconds) of the sampling interval. The valid format is a positive integer.

Kilobytes Rate (per sec) The rate, per second, at which kilobytes sent and received. The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Request Rate (per sec) The rate at which HTTP requests were made. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 139. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Failures Rate (per min) The rate (per minute) at which Apache web server failures occurred during the collection interval. The valid format is a decimal (formatted to three decimal places).

Attention: This attribute shows incorrect results for some environments. For more reliable information, use the server failures rate attribute of the "Apache web server attributes" on page 767.

SSL Whether SSL is enabled for the virtual host. Valid values are Disabled and Enabled.

Successful Login Rate (per min) The average number of successful logins that were made per minute. The valid format is a decimal (formatted to three decimal places).

Successful Request Rate (per sec) The average number of requests that were fulfilled, per second, during the collection interval. The valid format is a decimal (formatted to three decimal places).

Web Server Name The name of the web server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Web Site Name The name of the website. The valid format is an alphanumeric string, with a maximum of 256 characters.

Web Site Status The status of the Apache website. Valid values are Not_Running, Running, and Error.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Sun web server attributes

The **Sun Web Server** attributes provide status information about the Sun web server.

The attributes within this group are used to build the "Sun Java System web server workspace" on page 764.

Client Errors Rate (per sec) The number of 400-level (Client Error) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Connection Queue Count The number of connections currently in the web server connection queue. The valid format is a positive integer.

Connection Queue Max The maximum number of connections allowed in the web server connection queue. The valid format is a positive integer.

Connection Queue Overflows The number of connections rejected as a result of connection queue overflow. The valid format is a positive integer.

Connection Queue Peak The largest number of connections that were queued simultaneously. The valid format is a positive integer.

Connection Queue Total The number of connections that were accepted. The valid format is a positive integer.

Failed Login Rate (per min) The number of 401-level (Failed Login) responses issued by the web server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Idle Threads Count The number of web server request processing threads currently idle. The valid format is a positive integer.

Interval (sec) The length of the sampling interval in seconds. The valid format is a positive integer.

Keepalive Queue Count The number of connections currently in the web server keepalive queue. The valid format is a positive integer.

Keepalive Queue Max The maximum number of connections allowed in the web server keepalive queue. The valid format is a positive integer.

Kilobytes Rate (per sec) The summary rate at which kilobytes are received and transmitted on the network (per second). The valid format is a decimal (formatted to three decimal places).

Kilobytes Received Rate (per sec) The rate at which kilobytes are received on the network (per second). The valid format is a decimal (formatted to three decimal places).

Kilobytes Sent Rate (per sec) The rate at which Kilobytes are transmitted on the network (per second). The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The valid format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Other Responses Rate (per sec) The number of responses at a level other than 2xx, 3xx, 4xx, or 5xx that were issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Pages Not Found Rate (per min) The number of 404 (Pages Not Found) responses issued by the web server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Process ID The process identifier of the Java virtual machine. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Size Fraction System Memory Usage The fraction of system memory that is being used by the web server instance process. The valid format is a positive integer.

Process Size Resident The web server instance process resident size in kilobytes. The valid format is a positive integer.

Process Size Virtual The web server instance process size in kilobytes. The valid format is a positive integer.

Redirected Request Rate (per sec) The number of 300-level (Redirection) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Request Rate (per sec) The number of requests processed, per second. The valid format is a decimal (formatted to three decimal places).

Response Bad Request Rate (per sec) The number of 400® (Bad Request) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Forbidden Rate (per min) The number of 403 (Forbidden) responses issued by the web server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Moved Temporarily Rate (per sec) The number of 302 (Moved Temporarily) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Not Modified Rate (per sec) The number of 304 (Not Modified) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Successful Rate (per sec) The number of 200 (OK) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Unavailable Rate (per min) The number of 503 (Unavailable) responses issued by the web server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 140. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Contact The contact information for people responsible for server instance. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Death Times The number of times that the server instance processes stopped during the collection interval. The valid format is a positive integer.

Server Description The description of the server instance. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Directory The directory of the server instance. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Failures Rate (per min) The number of 500-level (Server Error) responses issued by the web server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Server Instance The MIB index for the web server instance. The valid format is a positive integer.

Server Location The location of the server instance. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Organization The organization that is responsible for the server instance. The valid format is an alphanumeric string, with a maximum of 256 characters.

Start Date and Time The date and time when the web server started. The valid format is a timestamp. This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the Sun web server. Valid values are Non_Running and Running.

Successful Request Rate (per sec) The number of 200-level (Successful) responses issued by the web server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Thread Count The number of web server request processing threads. The valid format is a positive integer.

Thread Pool Count The number of threads in the pool. The valid format is a positive integer.

Thread Pool Instance The thread pool MIB index. The valid format is a positive integer.

Thread Pool Max The maximum number of threads allowed in pool. The valid format is a positive integer.

Thread Pool Name The thread pool identifier. The valid format is an alphanumeric string, with a maximum of 256 characters.

Thread Pool Peak The maximum number of threads in the pool. The valid format is a positive integer.

Version The software version of the server instance. The valid format is an alphanumeric string, with a maximum of 256 characters.

Web Server Name The name of the web server. The valid format is an alphanumeric string, with a maximum of 256 characters.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Sun websites attributes

The **Sun Web Sites** attributes provide status information about the Sun websites workspace of the Sun Java System web server.

The attributes within this group are used to build the “Sun websites workspace” on page 764.

Address Port SSL The Sun website IP address, port and whether security is enabled. The valid format is an alphanumeric string, with a maximum of 256 characters.

Client Errors Rate (per sec) The number of 400-level (Client Error) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Failed Login Rate (per min) The number of 401 (Failed Login) responses issued by the virtual server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Failed Responses Percent The percentage of failed responses of the total request count. The valid format is a positive integer.

Interval (sec) The length of the sampling interval in seconds. The valid format is a positive integer.

Kilobytes Rate (per sec) The summary rate at which kilobytes are received and transmitted on the network per second. The valid format is a decimal (formatted to three decimal places).

Kilobytes Received Rate (per sec) The rate at which kilobytes are received on the network (per second). The valid format is a decimal (formatted to three decimal places).

Kilobytes Sent Rate (per sec) The rate at which kilobytes are transmitted on the network (per second). The valid format is a decimal (formatted to three decimal places).

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Other Responses Rate (per sec) The number of responses at a level other than 2xx, 3xx, 4xx, or 5xx that were issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Pages Not Found Rate (per min) The number of 404 (Pages Not Found) responses issued by the virtual server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Redirected Request Rate (per sec) The number of 300-level (Redirection) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Request Rate (per sec) The number of requests processed per second by the virtual server during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Bad Request Rate (per sec) The number of 400 (Bad Request) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Forbidden Rate (per min) The number of 403 (Forbidden) responses issued by the virtual server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Moved Temporarily Rate (per sec) The number of 302 (Moved Temporarily) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Not Modified Rate (per sec) The number of 304 (Not Modified) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Successful Rate (per sec) The number of 200 (OK) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Response Unavailable Rate (per min) The number of 503 (Unavailable) responses issued by the virtual server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 141. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Failures Rate (per min) The number of 500-level (Server Error) responses issued by the virtual server (per minute) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Server Failures Percent The percentage of server failures of the total request count. The valid format is a positive integer.

Server Instance The web server MIB index for the web server instance. The valid format is a positive integer.

Successful Request Rate (per sec) The number of 200-level (Successful) responses issued by the virtual server (per second) during the collection interval. The valid format is a decimal (formatted to three decimal places).

Successful Requests Percent The percentage of successful requests of the total request count. The valid format is a positive integer.

Unauthorized Requests Percent The percentage of unauthorized requests of the total request count. The valid format is a positive integer.

Web Server Name The name of the web server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Web Site Instance The web server MIB index for the virtual server instance. The valid format is a positive integer.

Web Site Name The name of the website. The valid format is an alphanumeric string, with a maximum of 128 characters.

Web Site Status The status of the SUN website. Valid values are Running, Error, and AddressPortUnavailable.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

HTTP Servers Agent Events attributes

The **HTTP Servers Agent Events** attributes collect information about agent-level events that affect the ability of the IBM Tivoli Composite Application Manager for Web Servers agent to collect data for Web Servers, including Apache web server and Sun Java Web server.

The attributes within this group are used to build the "web server Agent workspace" on page 765.

Event Date and Time The date and time the event occurred. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 142. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Message Description The description of the message. The valid format is an alphanumeric string, with a maximum of 256 characters.

Message ID The unique identifier of the message. The valid format is an alphanumeric string, with a maximum of eight characters.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 128 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Sequence Number The sequence number of the message. The valid format is a positive integer.

Severity The severity of the message. Valid values are Info, Warning, Error, and Severe.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Web Servers Status attributes

The **Web Servers Status** attributes provide status about the monitored web servers.

The attributes within this group are used to build the "web server Agent workspace" on page 765.

Node Name The name of the system on which the server is running. The value format is an alphanumeric string, with a maximum of 256 characters.

Origin Node The name of the server subnode. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process ID The identifier of the web server process. The valid format is an alphanumeric string, with a maximum of 64 characters.

Sample Date and Time The date and time the Tivoli Enterprise Management Agent collected the data. The valid format is a 12-character timestamp. For the STR and SCAN functions, the format is MM/DD/YY HH:MM:SS; the following table shows the values contained in this character string:

Table 143. Format of the 12-character timestamp

Character String	Meaning
MM	Month
DD	Day
YY	Year
HH	Hour
MM	Minute

Table 143. Format of the 12-character timestamp (continued)

Character String	Meaning
SS	Second

Example: 09/13/06 18:32:03 indicates the data was collected on September 13 2006, at 18:32:03.

This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Server Directory The directory of the web server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Name The name of the web server. The valid format is an alphanumeric string, with a maximum of 256 characters.

Server Type The type of the web Server. Valid values are Apache and Sun_Web_Server

Start Date and Time The date and time when the web server started. The valid format is a timestamp. This attribute was designed for logging and reporting data collection times rather than for creating situations. To specify a time and date for comparison and testing, use attributes from the Universal Time or Local Time groups.

Status The status of the web server. Valid values are Error, Stopped, Start_Pending, Stop_Pending, Running, Continue_Pending, Pause_Pending, and Paused.

For additional information, see:

- Organization of the predefined workspaces
- Attribute groups used by the predefined workspaces

Situations for ITCAM Agent for HTTP Servers

IBM Tivoli Composite Application Manager Agent for HTTP Servers provides a number of predefined situations that you can use to complete the following tasks:

- Monitor your web servers
- Monitor and manage widely dispersed web server resources through localized automation
- Create your own situations using the predefined situations as examples

These predefined situations have alert status of Critical and Warning. When these situations trigger an alert, you can investigate the event by opening its workspace.

How the situations work

Situations are tests expressed in IF-TRUE statements of system conditions that you want to monitor; the tested value is an ITCAM Agent for HTTP Servers attribute expressed in the form *attribute-group.attribute-name*. Thus, if the specified condition occurs or exists, the situation is true, and an alert is issued.

Avoid using negative values

If you define situations that use a counter or a range of numbers, always provide a threshold or use values in a positive range of numbers. For example, use a greater-than-or-equal-to-zero expression as shown in some of the following predefined situations. This practice prevents a situation from falsely tripping. If the ITCAM Agent for HTTP Servers Tivoli Enterprise Monitoring Agent encounters an undefined attribute value, it interprets this as a negative number and erroneously triggers a situation that specified a negative number.

Predefined situations, descriptions, and formulas

Apache_HTTP_Stopped Monitors the status of the Apache web server and issues a Critical condition when the Apache HTTP server is not active. Its formula is as follows:

If

Apache_Web_Server.Server_Status does NOT equal 1

then

the situation Apache_HTTP_Stopped is true.

Apache_Site_Down Monitors the status of the Apache website and issues a Critical condition when one of the virtual hosts run by the Apache HTTP Server is unavailable. Its formula is as follows:

If

Apache_Web_Sites.Web_Site_Status does NOT equal 1

then

the situation Apache_Site_Down is true.

Apache_Site_failed Monitors the server failure rate of Apache web server and issues a Critical condition whenever the server failures rate is greater than 1. Its formula is as follows:

If

Apache_Web_Sites.Server_Failures_Rate is greater than 1

then

the situation Apache_Site_failed is true.

Apache_Site_trafic Monitors the count of kilobytes sent and received and issues a Warning condition whenever the kilobytes rate is greater than 10,000,000. Its formula is as follows:

If

Apache_Web_Sites.Kilobytes_Rate is greater than 10,000,000

then

the situation Apache_Site_trafic is true.

Apache_Site_requests Monitors the rate at which the Apache HTTP requests were made and issues a Warning condition whenever the request rate is greater than 100,000. Its formula is as follows:

If

Apache_Web_Sites.Request_Rate is greater than 100,000

then

the situation Apache_Site_requests is true.

Apache_Site_fail_logins Monitors the count of failed logins and issues a Warning condition whenever the failed login rate is greater than 100. Its formula is as follows:

If

Apache_Web_Sites.Failed_Login_Rate is greater than 100

then

the situation Apache_Site_fail_logins is true.

Apache_Site_fail_pages Monitors the rate of failed pages and issues a Warning condition whenever the failed pages rate is greater than 1,000. Its formula is as follows:

If

Apache_Web_Sites.Failed_Pages_Rate is greater than 1,000

then

the situation Apache_Site_fail_pages is true.

SWebSrvStoped Monitors the status of the Sun web server and issues a Critical condition when the server stops. Its formula is as follows:

If

Sun_Web_Server.Server_Status equals 0

then

the situation SWebSrvStoped is true.

SWebSrvHFrMEM Monitors the fraction of system memory that is being used by the web server instance process and issues a Warning condition whenever the system memory usage is greater than 80%. Its formula is as follows:

If

Sun_Web_Server.Process_Size_Fraction_System_Memory_Usage is greater than 80

then

the situation SWebSrvHFrMEM is true.

SWebSrvCONQLIM Monitors the numbers of connections in the connection queue and issues a Warning condition whenever the count is greater than 100. Its formula is as follows:

If

Sun_Web_Server.Connection_Queue_Count is greater than 100

then

the situation SWebSrvCONQLIM is true.

SWebSrvKPALQLIM Monitors the number of connections in the keepalive queue and issues a Warning condition whenever the number exceeds 100. Its formula is as follows:

If

Sun_Web_Server.Keepalive_Queue_Count is greater than 100

then

the situation SWebSrvKPALQLIM is true.

SWebSrvHNetSent Monitors the kilobytes transmitted on the network of the Sun web server and issues a Warning condition whenever the number exceeds 10,000,000 kilobytes per second. Its formula is as follows:

If

Sun_Web_Server.Kilobytes_Out_Rate is greater than 10,000,000

then

the situation SWebSrvHNetSent is true.

SWebSrvHNetRecv Monitors the kilobytes received on the network of the Sun web server and issues a Warning condition whenever the number exceeds 10,000,000 kilobytes per second. Its formula is as follows:

If

Sun_Web_Server.Kilobytes_In_Rate is greater than 10,000,000

then

the situation SWebSrvHNetRecv is true.

SVWebStHNetRecv Monitors the kilobytes received on the network of the Sun website and issues a Warning condition whenever the number exceeds 10,000,000 kilobytes per second. Its formula is as follows:

If

Sun_Web_Sites.Kilobytes_In_Rate is greater than 10,000,000

then

the situation SWebStHNetRecv is true.

SVWebStHNetSent Monitors the kilobytes transmitted on the network of the Sun website and issues a Warning condition whenever the number exceeds 10,000,000 kilobytes per second. Its formula is as follows:

If

Sun_Web_Sites.Kilobytes_Out_Rate is greater than 10,000,000

then

the situation SWebStHNetSent is true.

SVWebStFailed Monitors the percentage of failed responses violation and issues a Critical condition whenever the percentage exceeds 50%. Its formula is as follows:

If

Sun_Web_Sites.Percentage_of_failed_responses_violation is greater than 50

then

the situation SVWebStFailed is true.

SVWebStServErr Monitors the percentage of server error violations and issues a Critical condition whenever the percentage exceeds 50%. Its formula is as follows:

If

Sun_Web_Sites.Percentage_of_server_errors_violation is greater than 50

then

the situation SVWebStServErr is true.

SVWebStUnAuthErr Monitors the percentage of unauthorized response violations and issues a Warning condition whenever the percentage exceeds 50%. Its formula is as follows:

If

Sun_Web_Sites.Percentage_of_unauthorized_responses_violation is greater than 50

then

the situation SVWebStUnAuthErr is true.

SVWebStSucssfResp Monitors the percentage of successful response violations and issues a Critical condition whenever the percentage exceeds 50%. Its formula is as follows:

If

Sun_Web_Sites.Percentage_of_successful_responses_violation is greater than 50

then

the situation SVWebStSucssfResp is true.

Take Action commands for ITCAM Agent for HTTP Servers

The Take Action commands allow your interactive Tivoli Enterprise Portal users enter a command or stop or start a process at any system in your network where one or more Tivoli Enterprise Monitoring Agents are installed. The ITCAM Agent for HTTP Servers Take Action commands allow you to use the Tivoli Enterprise Portal interface to start, stop, or restart a web server or a website.

Users can run a Take Action command from a workspace, from the Navigator, from a situation that you create, in an improvised mode, or by recalling a saved Take Action command. For details about using these general commands, see the online help for Tivoli Enterprise Portal.

Predefined Take Action commands for Apache web servers

StartServer: Start an Apache web server

Use the StartServer command to start an Apache web server instance.

Command syntax

HT:startServer

StopServer: Stop an Apache web server

Use the StopServer command to stop an Apache web server instance.

Command syntax

HT:stopServer

RestartServer: Restart an Apache web server

Use the RestartServer command to restart an Apache web server instance.

Command syntax

HT:restartServer

Predefined Take Action commands for Sun web servers

StartServer: Start a Sun web server

Use the StartServer command to start a Sun web server.

Command syntax

HT:startServer

StopServer: Stop a Sun web server

Use the StopServer command to stop a Sun web server.

Command syntax

HT:stopServer

RestartServer: Restart a Sun web server

Use the RestartServer command to restart a Sun web server.

Command syntax

HT:restartServer

Glossary for ITCAM Agent for HTTP Servers

A

Apache HTTP server An open source web server. IBM offers a web server, called the IBM HTTP Server, which is based on Apache.

application server A program in a distributed network that provides the execution environment for an application program.

attribute Data associated with a component. For example, a web server component might have attributes such as web server name, WWW server status, node name, origin node, interval time, and so on.

attribute group A set of related attributes that can be combined in a view or a situation. For example, the Apache web server attribute group and the Sun websites attribute group in the ITCAM Agent for HTTP Servers.

H

HTTP Hypertext Transfer Protocol. A suite of protocols for the Internet that transfer and display hypertext documents.

HTTP request A transaction initiated by a web browser and adhering to HTTP. The server usually responds with HTML data, but can send other kinds of objects as well.

I

IBM Tivoli Monitoring platform The software architecture and foundation that support the development and operations of Tivoli Enterprise Portal and its Tivoli Enterprise Monitoring Agents.

J

J2EE Java 2 Platform, Enterprise Edition. An environment for developing and deploying enterprise applications using the Java language. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functions needed for developing multi-tiered, Web-based applications.

Java An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

JSP Java server page. A web page that specifies one or more servlets whose execution on the web server modifies the page content or appearance before it is presented to the user.

M

monitor 1) A transaction environment for maintaining large quantities of data in a consistent state and that controls which users and clients can access data through authorized servers. 2) A programming primitive created so multiple program threads can share the same resource (such as an object). A program creates a monitor for a given resource by requesting it from the system; the system returns a unique ID for that monitor. After that, any thread needing the resource must use the monitor to lock the resource while the thread is using it. If the monitor is already locked, a thread requesting the resource is queued by the system and then given control when the monitor becomes unlocked. Also called a mutex.

monitored application An application that interfaces with a Tivoli Privacy Manager monitor to enable access to monitored items that flow between the application and the monitored system.

N

node Any managed system, such as a web server, that the Tivoli Enterprise Portal is monitoring. A node can also be a managed system of subnodes being managed under a primary node.

R

request The entry point into an application whose processing or response time directly affects the response time the user perceives.

S

server In a network, hardware, or software that provides facilities to clients. Examples of a server are a file server, a printer server, or a mail server.

session A series of requests to a servlet that originate from the same user at the same browser. Using sessions, applications running in a web container can track individual users.

T

thread pool The threads that are being used by or are available to a computer program.

U

URI Uniform Resource Identifier. An identifier for a point of content on the Internet, be it a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the web page address, which is a particular form of URI called a Universal Resource Locator (URL). A URI typically describes the mechanism used to access the resource, the computer that houses the resource, and the name of the resource (such as a file name) on that computer.

URL Universal Resource Locator. The unique address for a file accessible via the Internet. Such a file might be a web page (usually the home page), an image file, or a program such as a Java applet or servlet. The URL comprises the protocol used to access the file, a domain name that identifies a specific computer on the Internet, and a path name that specifies that file location on that computer.

V

virtual server A server that shares its resources with other servers to support applications.

W

Web Server A software program that can service Hypertext Transfer Protocol (HTTP) requests.

Web service A self-contained, self-describing modular application that can be published, discovered, and invoked over a network using standard network protocols. Typically, XML is used to tag the data, SOAP is used to transfer the data, WSDL is used to describe the services available, and UDDI is used for listing what services are available.

Website A related collection of files available on the web that is managed by a single entity (an organization or an individual) and contains information in hypertext for its users. A website often includes hypertext links to other websites.

workspace A window composed of one or more views.

Part 3. Appendixes

© Copyright IBM Corp. 2006, 2015

Appendix. Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.

The accessibility features in the product enable users to:

- Use assistive technologies, such as screen reader software and digital speech synthesizers, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using the technology with this product.
- Perform tasks with the software using only the keyboard.

General Navigation

Each page has four main sections:

- Headerbar
- Toolbar
- Main tabs
- Content

Each page has navigation points for screen readers. The following navigation points are all H1:

- Title bar
- Main tabs
- Main form
- Section labels
- Table labels

Menu Navigation

You use the Go To menu at the top of the screen to navigate to any of the applications that you have access to. The Go To menu is a cascading menu that is three levels deep at its deepest point. The following instructions describe how to get started with JAWS:

1. To get to the Go To menu press Alt+G.
2. When you open the menu, JAWS reads the first application in the menu. If JAWS does not begin to read the entry, restart the screen reader.
3. Navigate the list of applications in the menus by using the arrow keys.
4. JAWS indicates if a menu item has submenus. To get to a submenu, press the right arrow or enter.
5. Press the left arrow to move up a level in the hierarchy. If you press the left arrow at the highest level of the Go To menu, you leave the menu completely.
6. Press the Enter key to enter an application.

Accessibility help

The Accessibility Help panels provide details on general navigation, menu navigation, and hot keys. Click **Accessibility Help** from the toolbar of the product to access the help panels.

© Copyright IBM Corp. 2006, 2015

Screen reader setting

The product contains a screen reader flag. When you turn on the screen reader flag, the user interface is optimized to work with JAWS for Windows®. You use the **User** tab in the Users application to turn on the screen reader flag.

Keyboard shortcuts

You can navigate within the applications by using a combination of keys.

Accessible reports

To use the accessibility tools to read reports, you must access the reports in Microsoft Excel. In the reports applications, select the **Run Reports** option in the **Select Action** menu. With this option, you can email an .xls file version of a report to yourself at a scheduled time.

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: http://www.ibm.com/able

Index

A

account
  creating a user account 47
  deleting a user account 49
  modifying a user account 48
action
  alert 154
  data 154
action history, trap 154
active request
  canceling 124, 141
  lock contention 121
  priority 124, 140
  searching 115
  server activity display 116, 117
administration
  account management 46, 47
    role configuration 49, 50, 51, 52
    user profiles 47, 48, 49
  managing server 83
    system properties 83, 85, 86
  Monitoring on Demand (TM) 77, 79, 80, 81, 82
  server management 53
    data collector configuration 56, 58, 59, 60, 61, 62, 63
    self-diagnosis 86, 87, 88, 89
    server group 53, 54, 55
  user profiles 47
alert action 154
alerts 108, 109
  escalating 109
analysis
  capacity 169
  heap 135
  lock 164
  memory 134
  method 161
  MQI 163
  program 161
  request 160
  server availability 169
  SQL 162
  system resource 168
  transaction 160
application trap
  creating 144
  defining 144
  setting 144
archive agent
  self-diagnosis 87
audit trail
  user 200
audit-ms.log 200
authoritative server 157, 159
authorization
  component request 194
availability
  recent activity 100
    creating a report 100
availability *(continued)*
  server statistics overview
    configuring 98
  SMF data 107
  system resources 101, 103
    browser 102
  systems overview 89
    alerts 108
    alerts, escalating 109
    portal overview 93
    portlet summary 94
    Problem Center 110
    Problem Center, closing 114
    Problem Center, details 111
    Problem Center, manually adding 113
    server, WLM associated service class period detail 95
    server, WLM associated service class summary 95
    server, WLM enclave 96

B

baselining 751
browser
  system resources 102
    metrics 103
    WebSphere, PMI 103
    WebSphere, z/OS 103

C

calculating baseline 751
calculating threshold 751
capacity
  analysis 169
CICS
  composite requests 187, 191
  data collectors 190, 191
CICS transaction 103, 190
comparison
  installed binary 157
    results 157
  runtime environment 159
comparison server 157, 159
component trace 125, 194
  searching 126
composite
  indicator 196, 197
  request 187, 189, 190, 191, 192, 196, 197
    detail 196, 197
    indicator 192, 193, 194, 197, 198
    method trace 189, 190
    space 189, 190, 191
    stack trace 189, 190
    transaction 189, 190, 191
  transaction 187, 190, 191, 192
composite request indicator 192, 193, 194
configuration 56
  applying 62
  data collector 56
  deleting 63
  duplicating 63
  modifying 62
  role 49
contention
  lock 121

D

daily statistics
  deleting 183
  overview
    viewing 182
data
  PMI 103
  SMFviewing 107
data action 154
data collection settings
  configuring 83
data collector 56
  configuring 56, 58
    MQ 191
  controller
    self-diagnosis 88
  disabling 60
  enabling 60
  removing 59
  unconfiguring 59
data collector profiles 65
data collector profiles, adding 66
data collector profiles, configuring 68
data collector profiles, editing 65
data collector profiles, importing 67
data collector profiles, installing 75
data collector profiles, removing 66
data collector profiles, uninstalling 76
data collector profiles, viewing 65
database connection pools 103
date range settings 171
default
  system properties 83
Delta Normal CP time 99
Delta zAAP time 99
Delta zAAP-eligible time 99
Delta zAAP-eligible time on CP 99
detail
  request 122, 195
diagnosis
  memory 130
directory
  logs 200
display
  JVM thread 139, 140
  recent activity 100
    creating a report 100
  server activity 116, 119

© Copyright IBM Corp. 2006, 2015

display *(continued)*
activating a thread 123
active requests 117
canceling a request 124
changing a thread's priority 124
e-mailing a PDF 127
exporting to a file 128
lock contention 121
request detail 122
searching a method trace 126
suspending a thread 123
viewing a method trace 125
viewing a PDF 128
viewing a stack trace 125
viewing the request object and session object 126

E

e-mailing
PDF 127
edata collector profiles, exporting 68
EJB 103
enterprise overview
configuring 85
environment
runtime
comparison 159

F

file name match 157
file name/path/size match 157
filter list
exclude 62
exclude override 62
flow view 125
full match 157

G

global publish server
self-diagnosis 88
Glossary 207
group
access rights 48
create 53
creating 53
deleting 55
duplicating 55
modifying 54
server 53

H

heap analysis 135
heap dump
scheduling 133
heap dump management 130
history
trap action 154
HTTP
session
Web Session Browser 129

I

IBM Support Assistant
downloading
Memory Dump Diagnostic for Java 131
in-flight request
searching 192
in-flight request search 114, 115, 187, 188, 189, 190, 191, 192, 193, 194
composite
method trace 197
stack trace 196
request 196
sorting 116
installed binary
comparison
setting 157
viewing, results 157
interface
Request Mapper 204

J

JCA connection pools 103
JTA transactions 103
JVM thread display 139
canceling a thread 141
changing a thread's priority 140
stack trace 140
thread dump 142
JVM/system 103

K

kernel
self-diagnosis 86

L

L1, monitoring level 77
L2, monitoring level 77
L3, monitoring level 77
level
monitoring 77, 79, 81
lock
analysis
decomposition 164
detail 164
contention 121
logs
directory 200

M

managed space 188, 189, 190, 191
management
account 46
schedule
applying 80
creating 79
deleting 82
duplicating 82
modifying 81
server 52
traps 142
managing server 83
mapping behavior 202
match
file name 157
file name/path/size 157
full 157
maximum method records 83
MD5 checksum 157
memory
analysis 134
diagnosis 130
leak
candidate finder 137, 138
confirmation 136
diagnosis 136
overview 136, 137, 138
leak, diagnosis
references to live objects on the heap 138
Memory Dump Diagnostic for Java
downloading 131
message dispatcher
self-diagnosis 89
method profiling 180
activating 181
management 181
viewing 181
method trace 125, 154, 194
composite 197, 198
searching 126
metrics
CICS transaction 103
database connection pools 103
EJB 103
JCA connection pools 103
JTA transactions 103
JVM/system 103
Object Request Brokers (ORB) detail/interceptor 103
queue 103
queue manager 103
server 103
server regions 103
session manager 103
SQL 103
thread pools 103
Web applications 103
misbehaving transactions 146
misbehaving traps 146
mode
problem determination 77, 79
production 77, 79
tracing 77, 79
monitoring level 81
problem determination 77, 79
production 77, 79
schedule management 79
tracing 77, 79
Monitoring on Demand (TM), MOD 77
multiple hops 191

N

network
configuring the SNMP 86

O

object
  request 126
  session 126
Object Request Brokers (ORB)
  detail/interceptor 103
ORB detail/interceptor 103
original request string (ORS) 201, 202
ORS, original request string 201, 202
overview 99
  enterprise
    configuring 85
  memory leak
    confirmation 136
    diagnosis 136
  memory leak candidate finder
    creating 137
    viewing 138
  portal page summary 93
  portlet summary 94
  server
    WLM associated service class
      period detail 95
    WLM associated service class
      summary 95
    WLM enclave 96
  server statistics
    configuring 98
  systems 89

P

PDF
  e-mailing 127
  viewing 128
performance analysis 160, 193, 197, 198
  create application reports 160
    capacity analysis 169
    lock analysis 164
    MQI 163
    portal 165, 166
    request/transaction 161
    schedule 170
    server availability 169
    SQL 162
    system resource 168
    top reports 167
  method profiling 180, 181
  view saved reports 175
    deleting a report 178
    duplicating a report 177
    e-mailing a report 179
    exporting to a file 180
    modifying a report 175
    modifying a top report 176
    running a report 176
    understanding the date range
      settings 171
    viewing a PDF 180
    viewing the detail report 173
    viewing the reports 177
performance analysis and reporting 160
PMI data 103
portal 165
portal page summary 93
portlet 166
portlet summary
  overview 94
Problem Center 110
  closing 114
  details 111
  manually adding 113
problem determination 77
  in-flight request
    composite requests 187, 188, 189,
      190, 191, 192, 193, 194
    server activity display 192
  in-flight request search 114, 192
    application request 115, 116
    composite method trace 197
    composite stack trace 196
    request detail 196
  JVM thread display 139
    canceling a thread 141
    changing a thread's priority 140
    stack trace 140
    thread dump 142
  memory diagnosis 130
    heap analysis 135
    heap dump management 130, 133
    memory analysis 134
    memory leak 136, 137, 138
  server activity display 116, 117, 119
    activating a thread 123
    canceling a request 124
    changing a thread's priority 124
    component trace 125, 126, 194
    exporting to file 128
    lock contention 121
    method trace 125, 126, 194
    PDF 127, 128
    request detail 122, 195
    request object 126
    session object 126
    stack trace 125, 195
    suspending a thread 123
  software consistency check 156
    comparison 157
    installed binary comparison 157
    runtime environment
      comparison 159
  trap and alert management
    activating 150
    alert actions 154
    application trap 144
    data actions 154
    deactivating 151
    deleting 153
    duplicating 153
    managing 142
    modifying 152
    server resource trap 148
    trap action history 154
  Web Session Browser 129
production 77
  overriding 81
  problem determination 81
  tracing 81
profiles
  user 47
properties, system 83
publications, online xi
publish server
  self-diagnosis 87

Q

queue 103
queue manager 103

R

recent activity display 100
  creating a report 100
recent request
  server activity display 116, 119
references to live objects on the
  heap 138
reports 175
  capacity analysis 169
  create 160
  creating 160
  date range settings 171
  decomposition 160, 161, 162, 163,
    164, 168
  defining 160
  deleting 178
  detail 164, 173
  duplicating 177
  e-mailing 179
  exporting 180
  lock analysis 164
  memory analysis 134
  memory leak
    candidate finder 137, 138
    confirmation 136
    diagnosis 136
  method 161
  method analysis 161
  modifying 175
    top 176
  MQI analysis 163
  portal page 165
  portlet 166
  program analysis 161
  references to live objects on the
    heap 138
  request
    analysis 160
    detail 160
  running 176
  scheduled 170
  server availability analysis 169
  setting 160
  SQL analysis 162
  system resource analysis 168
  top 167
  transaction
    analysis 160
  trend 161, 162, 163, 168
  view 160
  viewing 177
    PDF 180
request
  active
    application 115
  authorization 194

request *(continued)*
composite 187, 188, 189, 190, 191, 192, 193, 194
detail 196
detail, PAR 197
detail, SAD 195
method trace 197
method trace, PAR 198
stack trace 196
detail
viewing 122
mapping behavior 202
name 200, 201
sampling rate 83
string 200, 201
Request Mapper 200
data 200, 201
deploying 203
interface 204
sample 205
writing 203
resident time - misbehaving transactions 146
resource
installed binary 157
role
assigning 50
configuring 49
creating 50
deleting 52
duplicating 51
modifying 51
runtime
environment comparison 159
running 159

S

sample
Request Mapper 205
schedule
applying 80
creating 79
deleting 82
duplicating 82
modifying 81
reports 170
viewing, management 79
self-diagnosis 86
archive agent 87
data collector controller 88
global publish server 88
kernel 86
message dispatcher 89
publish server 87
server 103
authoritative 157, 159
availability
analysis 169
comparison 157, 159
groups 53
management 52
managing 83
regions 103
server activity 99
server activity display 116, 117, 119, 192
activating a thread 123
server activity display *(continued)*
canceling a request 124
changing a thread's priority 124
component trace 125, 194
searching 126
lock contention 121
method trace 125, 194
searching 126
PDF 127, 128
exporting to file 128
request detail 122, 195
request object 126
session object 126
stack trace 125, 195
suspending a thread 123
server resource trap
creating 148
defining 148
setting 148
server statistics overview
configuring 98
session
Web Session Browser 129
viewing 129
session manager 103
settings
configuring the data collection 83
data collection 83
date range 171
SMF data 107
SNMP network
configuring 86
software consistency check 156
installed binary
comparison 157
runtime environment
comparison 159
SQL 103
stack trace 195
composite 196
string
original request 201, 202
request 200, 201
system
properties 83
resources
polling frequency 83
system resource analysis 168
system resources 101
browser 102
metrics 103
WebSphere, PMI 103
WebSphere, z/OS 103
systems overview 89
alerts 108
escalating 109
portal page summary 93
portlet summary 94
Problem Center 110
closing 114
details 111
manually adding 113
server
WLM associated service class summary 95
WLM associated service period detail 95
systems overview *(continued)*
server *(continued)*
WLM enclave 96

T

thread
activating 123
canceling 124, 141
dump 142
pools 103
priority 124, 125, 140
suspending 123
threshold 751
Tivoli Enterprise Portal 42
top reports 167
trace
component 125, 194
searching 126
composite
method 197, 198
stack 196
exporting 128
method 125, 154, 194
searching 126
stack 125, 140, 195
tracing 77
trap
action history 154
activating 150
alert action 154
application 144
creating 61
data action 154
deactivating 151
deleting 153
duplicating 153
modifying 152
server resource 148
trap action history 154
trap and alert management 142

U

user account 47
creating 47
deleting 49
modifying 48
user audit trail
accessing 200
user ids 202
user profiles 47

V

view
flow 125

W

Web applications 103
Web Session Browser 129

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

© Copyright IBM Corp. 2006, 2015

Notices

This information was developed for products and services offered in the U.S.A. IBM might 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 might 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might 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 or changes in the product(s) 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.

© Copyright IBM Corp. 2006, 2015

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 want 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
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information might 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 might vary significantly. Some measurements might 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 might have been estimated through extrapolation. Actual results might 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, 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, 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 for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.



Printed in USA