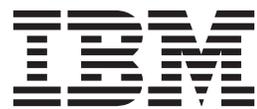


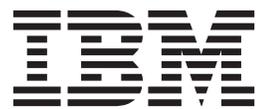
IBM Tivoli Composite Application Manager for Applications
Version 7.1

*WebSphere Message Broker
Monitoring Agent User's Guide*



IBM Tivoli Composite Application Manager for Applications
Version 7.1

*WebSphere Message Broker
Monitoring Agent User's Guide*



Note

Before using this information and the product it supports, read the information in "Notices" on page 339.

This edition applies to version 7.1 of WebSphere Message Broker Monitoring agent (product number 5724-V09 on Windows, UNIX, and Linux systems; product number 5698-B23 on z/OS systems) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2005, 2012.**

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

Contents

Figures vii

Tables ix

About this publication xi

Intended audience	xi
Publications	xi
Library for WebSphere Message Broker	
Monitoring agent	xi
Prerequisite publications	xii
Related publications	xii
Accessing terminology online	xiii
Accessing publications online	xiii
Ordering publications.	xiii
Accessibility	xiii
Tivoli technical training	xiii
Tivoli user groups	xiv
Support information	xiv
Conventions used in this publication	xiv
Typeface conventions	xiv
Revision bars.	xv

Chapter 1. Getting started 1

New in version 7.1	1
WebSphere Message Broker Monitoring agent	2
Workspaces and views	2
Monitoring situations	3
Attributes	3
Take Action commands	4
Broker data collection	4
Historical data collection	6
IBM Tivoli Monitoring	6
Tivoli Enterprise Monitoring Server.	7
Tivoli Enterprise Portal	7
Tivoli Enterprise Monitoring agents.	7
IBM Tivoli OMEGAMON DE product overview	8
Policy management	8

Chapter 2. Customizing the monitoring agent 9

Agent parameter file.	9
Names and locations of agent parameter files	9
Default agent parameter file	10
Agent parameter file syntax	11
Setting agent parameters	11
Modifying an existing parameter	12
Adding new parameters	12
Remotely configuring the WebSphere Message Broker Monitoring agent	14
Prerequisites	14
Remotely configuring through Tivoli Enterprise Portal	15
Remotely configuring through the command line	16

Creating multiple instances of the WebSphere Message Broker Monitoring agent	17
Windows systems: Creating multiple instances of the monitoring agent	17
UNIX and Linux systems: Creating multiple instances of the monitoring agent	18
Agent parameter descriptions	18
KqiAgent	19
MonitorBroker	28
ConnectQueueManager	33
Examples of agent configuration	34
Monitoring a large number of brokers	34
Disabling broker data collection	37
Enabling persistent broker data collection	37

Chapter 3. Monitoring with the CandleMonitor node 39

Prerequisites	40
Making the CandleMonitor node available in broker environments	40
Making the CandleMonitor node available in the Message Brokers Toolkit	44
Placing the CandleMonitor node in message flows	48
Monitoring the input or output of a message flow	49
Monitoring subflows	50
Monitoring other aspects of a message flow	52
Producing event messages	52
Guidelines for monitoring with the CandleMonitor node	53
Customizing a CandleMonitor node	53
Windows systems: Changing the values of configuration variables	54
UNIX or Linux systems: Changing the values of configuration variables	54
Deleting the CandleMonitor node from Message Brokers Toolkit	55
Message Brokers Toolkit V6.0: Deleting the CandleMonitor node	55
Message Brokers Toolkit V6.1: Deleting the CandleMonitor node	55
Message Brokers Toolkit V7.0 or later: Deleting the CandleMonitor node	56
WebSphere Message Broker V8 support: Known limitations and problems	56

Chapter 4. Using situations and Take Action commands 59

Predefined situations	59
Take Action commands	60
Authorizing Take Action users	61
Sending a Take Action command	61
Using Take Action commands in situations	61
Scenarios of using Situations and Take Action commands.	62

Preventing inadvertent use of trace active	62	Data model of WebSphere Message Broker	103
Determining when a message flow has failed	63	Monitoring agent	103
Collecting requested system trace data for a broker on a remote system	63	Sample reports	104
Stopping a message flow that has a full output queue	64	Known problems and workarounds	110
Automatically starting a broker that is stopped	64		
Starting and stopping message flows at periodic intervals	64		
Chapter 5. Monitoring with workspaces 67		Chapter 8. Configuring in a cluster environment on Windows systems 113	
Before you begin	67	MSCS clusters	113
Data availability	67	Active/active clustering	115
Comparison between broker accounting statistics data and CandleMonitor node statistics data	69	Prerequisites	115
Workspace summary	75	Configuring the WebSphere Message Broker	
Agent and application status workspaces	75	Monitoring agent	117
Broker and message flow information workspaces	76	Active/passive clustering	120
Event workspaces	76	Prerequisites	120
Statistics workspaces	77	Configuring the WebSphere Message Broker	
Accounting workspaces	78	Monitoring agent	122
Publish and Subscribe workspaces (WebSphere Message Broker V6 all releases only)	79	Known limitations	125
Resource Statistics workspaces (WebSphere Message Broker V7.0 or later only)	80		
Creating a workspace using a predefined workspace as a template	80	Chapter 9. Configuring in a cluster environment on AIX systems. 127	
Creating the user statistics workspace	81	Prerequisites	127
Scenarios of monitoring with workspaces	82	Active/active clustering	127
Monitoring application message flow performance	82	Active/passive clustering	129
Determining application delivery failure of messages	83	Configuring the WebSphere Message Broker	
Debugging a message flow	84	Monitoring agent	130
Verifying the broker configuration	84	Creating new instances of the WebSphere Message Broker Monitoring agent for each message broker	131
Planning broker capacity	85	Setting local variables in the agent configuration file	132
Ensuring reasonable message flow response times	85	Creating the directories for historical and situation data files	132
Verifying subscriptions without accessing the Message Brokers Toolkit	86	Configuring the Tivoli Enterprise Portal to list agents running in the cluster groups	133
Determining if an application has subscription access to a topic	86	Creating a file that is used to start the agent	133
Displaying a graphical view of your environment	86	Creating a file that is used to stop the agent	134
		Setting the scripts that are used to start and stop the agent in an HACMP environment	134
Chapter 6. Collecting historical data 91		Chapter 10. Reference 137	
Initial settings for historical collection	91	Attributes	137
Starting historical data collection	94	For WebSphere Message Broker V7.0 or later	137
Stopping historical data collection	95	For WebSphere Message Broker V6 (all releases)	192
Viewing historical data for a selected time frame	95	CandleMonitor node	258
Offline collection of historical data	97	Attributes	258
		Configuration variables	261
Chapter 7. Running reports with Tivoli Common Reporting 99		Take Action commands	262
Prerequisites	99	Commands for agent managed systems	264
Installing Cognos reports for WebSphere Message Broker Monitoring agent	101	Commands for broker-managed systems	265
Working with reports	102	Command restrictions on z/OS systems	270
Creating or editing Web-based reports	102	Workspaces	271
Creating ad hoc reports	102	For WebSphere Message Broker V7.0 or later	271
		For WebSphere Message Broker V6 (all releases)	290
		Appendix A. Accessibility 313	
		Magnifying what is displayed on the screen	313
		Navigating the interface using the keyboard	313

**Appendix B. Disk space requirements
for historical data tables 315**

Historical data tables 315
Historical table record sizes 317
Historical space requirement worksheets 321
Historical disk space summary worksheet 331

Appendix C. Language codes 335

Appendix D. Architecture codes 337

Notices 339

Trademarks 340

Glossary 343

A 343
B 343
C 344

D 344
E 344
F 345
H 345
I 345
L 345
M 345
O 346
P 346
Q 346
R 346
S 346
T 347
U 348
V 348
W 348

Index 349

Figures

1.	Broker data collection procedures performed at agent startup	5
2.	Agent–Server–Client architecture	7
3.	Default kqi.xml file on Windows, UNIX, and Linux systems.	10
4.	Default KQIXML file on z/OS systems	11
5.	Managed System Configuration window	15
6.	Example of a monitored flow	39
7.	Type I subflow	51
8.	Type II subflow	52
9.	Example of using the QI Start Component command in a situation	62
10.	The broker topology view.	87
11.	Started and stopped broker icons in the broker topology view.	88
12.	The message flow topology view	89
13.	The execution group topology view	90
14.	A two-computer MSCS cluster	114
15.	An example active/active cluster environment architecture with one cluster group active on each cluster node	116
16.	An example cluster environment architecture with a cluster group active on one cluster node and inactive on the other node.	121
17.	An example active-active cluster environment architecture with one cluster group active on each cluster node	128
18.	An example active-passive cluster environment architecture.	129
19.	QI Agent in the Navigator physical view	290
20.	Broker managed system	296
21.	Broker managed system merged with the QI Broker	296
22.	QI broker in the Navigator physical view	297

Tables

1. Comparison between broker accounting statistics and CandleMonitor node statistics	72	33. JDBC Connection Pools Resource Statistics (kqitrsjd) worksheet	323
2. Workspaces with user statistics links (for WebSphere Message Broker V7 or later)	81	34. JVM Resource Statistics (kqitrsjv) worksheet	323
3. Workspaces with user statistics links (for WebSphere Message Broker V6 all releases)	81	35. Message Flow Status (kqitmfls) worksheet	323
4. A selection of message flow topology nodes	89	36. Message Processing Nodes (kqitmpns) worksheet.	324
5. Historical attribute groups for sample reports	104	37. Monitor Node Base Statistics (kqitmfn) worksheet.	324
6. Parameters of Broker Daily Availability report	105	38. Monitor Node Broker Statistics (kqitmnr) worksheet.	324
7. Parameters of Broker Execution Group Daily Availability report	105	39. Monitor Node Events (kqitmnev) worksheet	324
8. Parameters of Broker Execution Group Weekly Availability report	106	40. Monitor Node Execution Group Statistics (kqitmneg) worksheet.	324
9. Parameters of Broker Message Flow Daily Availability report	106	41. Monitor Node Message Flow Statistics (kqitmnmf) worksheet	325
10. Parameters of Broker Message Flow Detail report	107	42. Monitor Node Sub-Flow Statistics (kqitmnsf) worksheet.	325
11. Parameters of Broker Message Flow Weekly Availability report	107	43. ODBC Resource Statistics (kqitrsod) worksheet.	325
12. Parameters of Broker Top n Elapsed Microseconds report	108	44. Parsers Resource Statistics (kqitrsps) worksheet.	325
13. Parameters of Broker Weekly Availability report	109	45. SOAPInput Resource Statistics (kqitrssp) worksheet.	325
14. From Object Name values	148	46. ACL Entries (kqitdsea) worksheet.	326
15. Object Status values	149	47. Broker Events (kqitbrev) worksheet	326
16. From Object Name values	202	48. Broker Information (kqitbrkr) worksheet	326
17. Object Status values	202	49. Broker Statistics (kqitstbr) worksheet	326
18. The effect of providing a subFlowName attribute for each node type.	259	50. CandleMonitor Node Statistics (kqitstfn) worksheet.	326
19. Historical data tables of the WebSphere Messaging Broker Monitoring agent (WebSphere Message Broker V7.0 or later)	315	51. Execution Group Information (kqitdfeg) worksheet.	327
20. Historical data tables of the WebSphere Messaging Broker Monitoring agent (WebSphere Message Broker V6 all releases)	316	52. Execution Group Statistics (kqitsteg) worksheet.	327
21. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V7.0 or later)	317	53. Message Flow Accounting (kqitasmf) worksheet.	327
22. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V6 all releases)	319	54. Message Flow Events (kqitflev) worksheet	327
23. Components (kqitcomp) worksheet	321	55. Message Flow Information (kqitdfmf) worksheet.	327
24. Product Events (kqitprev) worksheet	321	56. Message Flow Statistics (kqitstmf) worksheet	328
25. Accounting Message Flow Statistics (kqitacmf) worksheet	321	57. Message Processing Node Information (kqitdffn) worksheet	328
26. Accounting Node Statistics (kqitacnd) worksheet.	321	58. Multicast Group Statistics (kqitpsms) worksheet.	328
27. Accounting Terminal Statistics (kqitactr) worksheet.	322	59. Multicast Summary Statistics (kqitpsmg) worksheet.	328
28. Accounting Thread Statistics (kqitacth) worksheet.	322	60. Multicast Topic Statistics (kqitpsmt) worksheet.	328
29. Broker Status (kqitbrks) worksheet	322	61. Neighbors (kqitdsen) worksheet	329
30. Broker Status Events (kqitbsev) worksheet	322	62. Node Accounting (kqitasnd) worksheet	329
31. Execution Group Status (kqitegrs) worksheet	323	63. Publish-subscribe statistics (kqitpsst) worksheet.	329
32. File Resource Statistics (kqitrsfl) worksheet	323	64. Retained Publications (kqitdsr) worksheet	329
		65. Sub-Flow Statistics (kqitstsf) worksheet	329
		66. Subscriptions (kqitdses) worksheet	330
		67. Terminal Accounting (kqitastr) worksheet	330
		68. Thread Accounting (kqitasth) worksheet	330

69.	Disk space summary worksheet for historical tables (WebSphere Message Broker V7.0 or later)	331
70.	Disk space summary worksheet for historical tables (WebSphere Message Broker V6 all releases)	332
71.	Language codes for the supported languages	335
72.	Operating system architecture abbreviations	337

About this publication

This change applies to version 7.0.1 or later. The name of WebSphere® Message Broker Monitoring agent that runs on distributed systems is changed to IBM® Tivoli® Composite Application Manager (ITCAM) agent for WebSphere Message Broker. The agent name on z/OS® systems remains the same: WebSphere Message Broker Monitoring agent.

In this publication, the information is applicable to both these agents, but the agent is referred to as WebSphere Message Broker Monitoring agent. For users of distributed systems, this agent name is equivalent to ITCAM agent for WebSphere Message Broker.

WebSphere Message Broker Monitoring agent is an agent that is delivered as a part of ITCAM for Applications. You can use this agent to monitor, analyze, and manage IBM message broker products in your IBM business integration system. This book describes the features and capabilities of WebSphere Message Broker Monitoring agent, and explains how to use it to monitor your business integration systems.

Intended audience

This publication is intended for IT operations staff or administrators who monitor the reliability and availability of business integration systems and must react to any problems that occur within those systems. It is also useful for application developers who design, implement, and test business integration components and systems.

To perform any of the actions described in this document, you must have IBM Tivoli Monitoring and the ITCAM agent for WebSphere Message Broker installed and configured within your environment. For instructions, see the installation and configuration documents listed in “Prerequisite publications” on page xii.

Publications

This section lists publications in the library for WebSphere Message Broker Monitoring agent and related documents. The section also describes how to access Tivoli publications online and how to order Tivoli publications.

Library for WebSphere Message Broker Monitoring agent

The following documents are available in the library for WebSphere Message Broker Monitoring agent:

- *IBM Tivoli Composite Application Manager Agents for WebSphere Messaging: Installation and Setup Guide, SC14-7520*
Describes how to install WebSphere MQ Monitoring agent, WebSphere MQ Configuration agent, and WebSphere Message Broker Monitoring agent on Windows, UNIX, Linux, and i5/OS™ systems.
- *IBM Tivoli Composite Application Manager Agents for WebSphere Messaging: Upgrade and Migration Guide, SC14-7522*

Provides information about how to upgrade or migrate from previous versions of WebSphere MQ Monitoring agent, WebSphere MQ Configuration agent, and WebSphere Message Broker Monitoring agent to version 7.1.

- *IBM Tivoli Composite Application Manager Agent for WebSphere Message Broker User's Guide, SC14-7524*

Provides instructions for using the features of the WebSphere Message Broker Monitoring agent.

- *IBM Tivoli Composite Application Manager Agents for WebSphere Messaging: Troubleshooting Guide, GC14-7521*

Provides problem determination and resolution information for the issues most commonly encountered when using WebSphere MQ Monitoring agent, WebSphere MQ Configuration agent, and WebSphere Message Broker Monitoring agent.

Prerequisite publications

To use the information in this publication effectively, you must have some prerequisite knowledge, which you can obtain from the following publications:

- *IBM Tivoli Monitoring Installation and Setup Guide, SC32-9407*

Provides instructions for installing and configuring IBM Tivoli Monitoring components on Windows, Linux, and UNIX systems.

- *IBM Tivoli Composite Application Manager Agents for WebSphere Messaging: Installation and Setup Guide, SC14-7520*

Describes how to install WebSphere MQ Monitoring agent, WebSphere MQ Configuration agent, and WebSphere Message Broker Monitoring agent on Windows, UNIX, Linux, and i5/OS systems.

- *Configuring Tivoli Enterprise Monitoring Server on z/OS, SC32-9463*

Gives detailed instructions for using the Configuration Tool to configure Tivoli Enterprise Monitoring Server on z/OS systems. Includes scenarios for using batch mode to replicate monitoring environments across the z/OS enterprise. Also provides instructions for setting up security and for adding application support to a Tivoli Enterprise Monitoring Server on z/OS systems.

Related publications

The following documents also provide useful information:

- *IBM Tivoli Monitoring Administrator's Guide, SC32-9408*

Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.

- *Tivoli Enterprise Portal User's Guide, SC32-9409*

Provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.

- *IBM Tivoli Monitoring Troubleshooting Guide, GC32-9458*

Provides information and messages to help you troubleshoot problems with IBM Tivoli Monitoring.

- *IBM Tivoli Monitoring Command Reference, SC23-6045*

Provides detailed syntax and parameter information, as well as examples, for the commands you can use in IBM Tivoli Monitoring.

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. Refer to the readme file on the CD for instructions on how to access the documentation.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center Web site at <http://www.ibm.com/tivoli/documentation>.

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

Ordering publications

You can order many Tivoli publications online at <http://www.ibm.com/e-business/weblink/publications/servlet/pbi.wss>.

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 <http://www.ibm.com/e-business/weblink/publications/servlet/pbi.wss>.
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 all features of the graphical user interface.

For additional information, see Appendix A, "Accessibility," on page 313.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site at <http://www.ibm.com/software/tivoli/education>.

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users. Tivoli user groups include the following members and groups:

- 23,000+ members
- 144+ groups

Access the link for the Tivoli Users Group at www.tivoli-ug.org.

Support 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:

Online

Access the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to <http://www.ibm.com/software/support/isa>.

Troubleshooting Guide

For more information about resolving problems, see the *IBM Tivoli Composite Application Manager Agents for WebSphere Messaging: Troubleshooting Guide, GC14-7521*.

Conventions used in this publication

This publication uses several conventions for special terms and actions.

Typeface conventions

This publication 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, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")

- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents...

Monospace

- Examples and code examples
- Message text and prompts addressed to the user
- Text that the user must type

Revision bars

Revision bars are used in this book. Compared with the last version, changed and new information is indicated by a vertical bar (|) in the left margin.

Chapter 1. Getting started

The WebSphere Message Broker Monitoring agent is a monitoring and management tool that provides you with the means to verify, analyze, and tune message broker topologies associated with the following WebSphere products:

- IBM WebSphere Event Broker V6
- IBM WebSphere Message Broker with Rules and Formatter Extension V6
- IBM WebSphere Message Broker V6 (all releases)
- IBM WebSphere Message Broker V7 (all releases)
- IBM WebSphere Message Broker V8.0

Remember: WebSphere MQ version 7.0.1 is a prerequisite of WebSphere Message Broker V7.0. Before you upgrade WebSphere MQ to version 7.0.1 or WebSphere Message Broker to version 7.0, stop the WebSphere Message Broker Monitoring agent if it is running. Otherwise, the monitoring agent cannot connect to the queue manager after the upgrade. You have to manually restart the monitoring agent to fix the problem.

New in version 7.1

Version 7.1 of WebSphere Message Broker Monitoring agent has the following changes and enhancements:

- WebSphere Message Broker V7.0 is supported. Because the broker product and its architecture have been changed since version 7.0, the WebSphere Message Broker Monitoring agent provides a whole new set of workspaces, attributes, situations, and Take Action commands to monitor version 7.0 brokers or later.
- Resource Statistics data that is published by WebSphere Message Broker is now supported. Several resource statistics workspaces are provided to get the statistics of system resources collected by a broker. For an overview of these workspaces, see “Resource Statistics workspaces (WebSphere Message Broker V7.0 or later only)” on page 80.
- Support of WebSphere Message Broker V8 is provided. Applications and libraries, which are introduced by WebSphere Message Broker V8, can be monitored by the agent now.
- Self-describing agent application support installation is supported for WebSphere Message Broker Monitoring agent. With this self-describing capability, you do not have to manually install the agent application support on IBM Tivoli Monitoring components (monitoring server and portal server). For detailed instructions, see *IBM Tivoli Composite Application Manager Agents for WebSphere Messaging: Installation and Setup Guide, SC14-7520*.
- New configuration parameters are added in case the WebSphere Message Broker Monitoring agent cannot find the proper installation directory of WebSphere Message Broker or WebSphere MQ in a complex environment.
 - “defaultWMBInstallDirectory” on page 27
 - “defaultWMQInstallDirectory” on page 28
 - “WMBInstallDirectory” on page 33
 - “WMQInstallDirectory” on page 33
- The CandleMonitor node is supported in 64-bit broker environment on Windows systems. For instructions about how to install the CandleMonitor node available

in 64-bit broker environment, see “Making the CandleMonitor node available in 64-bit broker environments on Windows systems” on page 41.

- Cognos® data models and sample reports for WebSphere Message Broker Monitoring agent are provided to work with Tivoli Common Reporting V2.1.1. For more details, see Chapter 7, “Running reports with Tivoli Common Reporting,” on page 99.
- The PARMGEN configuration method is provided and now is the preferred method of the agent configuration on z/OS systems. For more information about configuring the WebSphere Message Broker Monitoring agent with the PARMGEN method, see *IBM Tivoli OMEGAMON XE for Messaging on z/OS: Planning and Configuration Guide, SC23-7951*.

WebSphere Message Broker Monitoring agent

You can use the WebSphere Message Broker Monitoring agent to ensure the reliability and performance of your broker environment by detecting and correcting broker and message flow problems before they have an impact on service speed and availability. The WebSphere Message Broker Monitoring agent also reduces the amount of time it takes to deploy broker applications by helping you to debug message flows and providing statistics that you can use to tune your environment.

You can use the WebSphere Message Broker Monitoring agent to do the following tasks:

- Monitor the status of your IBM broker product and its components
- View information and performance statistics for broker topologies at broker, execution group, message flow, node, terminal, and thread level in both tabular and chart forms
- Issue IBM broker product commands using the WebSphere Message Broker Monitoring agent interface to directly manage your environment, or create automatic responses to WebSphere Message Broker events
- Receive alerts when performance thresholds are exceeded or when message flow events occur
- Retain data samples in history files and save them to a historical database for reporting and analysis purposes

The WebSphere Message Broker Monitoring agent collects data from WebSphere Message Brokers. The data is presented in charts and tables that you can examine to monitor the performance of your WebSphere Business Integration systems. The agents also evaluate the data to determine when specified values meet the criteria that you have defined, and trigger alerts or programmed actions in response.

In addition, the WebSphere Message Broker Monitoring agent provides a CandleMonitor node. When inserted into a message flow, the CandleMonitor node collects message flow and subflow performance statistics and provides a mechanism for generating user-defined events. For the following workspaces to contain data, the broker must have at least one deployed CandleMonitor node (For information about how to deploy a CandleMonitor node in a broker, see Chapter 3, “Monitoring with the CandleMonitor node,” on page 39).

Workspaces and views

The *workspace* is the working area of the application window. The workspace provides you with status, definition, and statistical information in tabular and graphical form.

The workspaces predefined by the WebSphere Message Broker Monitoring agent provide the following information:

- Status information for brokers, components, and agents
- Information about brokers, execution groups, message flows, and message processing nodes
- Information about product, broker, and message flow events
- Information about the environment publish and subscribe configuration
- Performance statistics organized by broker, execution group, message flow, and subflow
- Accounting statistics organized by message flow, thread, node, and terminal

You can use the information that is provided by these workspaces to trace the causes of performance problems or the reasons why an alert is triggered.

You can also customize these workspaces and the views that they contain, or create your own workspaces and views to display information about a specific set of attributes.

For information about predefined workspaces included with the WebSphere Message Broker Monitoring agent, see Chapter 5, “Monitoring with workspaces,” on page 67, and the WebSphere Message Broker Monitoring agent section of the Tivoli Enterprise Portal online help. For information about how to create a workspace using a predefined workspace as a template, see “Creating a workspace using a predefined workspace as a template” on page 80. For detailed information about creating and customizing views and workspaces, see the Tivoli Enterprise Portal online help or *Tivoli Enterprise Portal User's Guide, SC32-9409*.

Monitoring situations

Situations are descriptions of conditions to which you want to be alerted. When situations are used on monitored systems they can, for example, alert you to a broker that has not been started, or to a message flow event. Situations can also be used to automate responses to problems, such as restarting a component or stopping a message flow that is consuming too much CPU processing time.

The WebSphere Message Broker Monitoring agent provides a set of predefined situations, which are designed to help you monitor critical activity and to serve as templates for creating customized situations of your own.

For information about the predefined situations included with the WebSphere Message Broker Monitoring agent, see “Predefined situations” on page 59. For information about creating and editing situations, see the Tivoli Enterprise Portal online help.

Attributes

Attributes are characteristics or properties of the objects monitored by the WebSphere Message Broker Monitoring agent; for example, the status of brokers, execution groups, and message flows, or the average time taken for a message flow to process a message. You specify attributes in query definitions, which are used to collect information presented in workspace views and to specify the conditions, or situations that trigger alerts and automated actions.

Attributes are organized into groups of related items. The WebSphere Message Broker Monitoring agent monitors 28 groups of attributes.

Attributes are used to define the queries which collect the information presented in workspace views. They are also used to specify the conditions, or *situations*, that trigger alerts and actions.

For information about the attribute groups available with WebSphere Message Broker Monitoring agent, see “Attributes” on page 137. For further information about attributes, see Tivoli Enterprise Portal online help.

Attributes and queries:

Chart and table views use queries to specify which attribute values to request from a Tivoli Enterprise Monitoring agent for display. You can modify those queries or design custom views by creating your own queries that collect data for just those attributes you specify. For more information about how to design custom views, see Tivoli Enterprise Portal online help.

Attributes and situations:

You can also use the attributes available with WebSphere Message Broker Monitoring agent to create your own situations to monitor the performance of your WebSphere Business Integration applications and message flows. These situations can monitor your WebSphere Business Integration broker resources or analyze message flows to alert you to problems when attribute values exceed the thresholds you specify.

For information about the predefined situations available with WebSphere Message Broker Monitoring agent, see “Predefined situations” on page 59.

Take Action commands

Using Tivoli Enterprise Portal, you can interact directly with your brokers and message flows with the Take Action feature. You enter your own commands, or choose from a list of predefined commands.

You can add a Take Action command to a monitoring situation and configure it to be run every time the situation becomes true. If you have IBM Tivoli OMEGAMON® DE, you can also create automation policies using Take Action commands.

You can use the Take Action commands included with WebSphere Message Broker Monitoring agent to issue IBM broker product commands from Tivoli Enterprise Portal. For example, you can start or stop brokers and their queue managers, and start and stop message flows, or change their trace characteristics by issuing commands from Tivoli Enterprise Portal. You can even change the type of accounting statistics collected or the interval at which they are collected.

For information about the commands provided with WebSphere Message Broker Monitoring agent, see “Take Action commands” on page 60 or the WebSphere Message Broker Monitoring agent section of the Tivoli Enterprise Portal online help.

Broker data collection

The WebSphere Message Broker Monitoring agent collects data about deployed broker objects for use in drawing topology views and display in workspace tables. The collection of this information can be slow because of the large volume of data

that must be retrieved from the broker. However, version 7.1 of WebSphere Message Broker Monitoring agent offers more efficient broker data collection to overcome these performance issues.

In versions of WebSphere Message Broker Monitoring agent prior to version 7.0, which did not include topology workspaces, broker data was requested from the broker every time the agent was started. This method has the advantage that data is always up-to-date, but data collection can be slow if complex message flows are involved. This method can cause long agent startup times and the large amounts of data that is requested from the broker can cause delays in the broker processing other messages.

The WebSphere Message Broker Monitoring agent version 7.1 includes persistent broker data and other new features to overcome these problems. Broker data is stored in a file even when the agent is stopped, so that there is no need for the agent to collect the data every time the agent is started. The process performed by the agent during startup is illustrated in Figure 1.

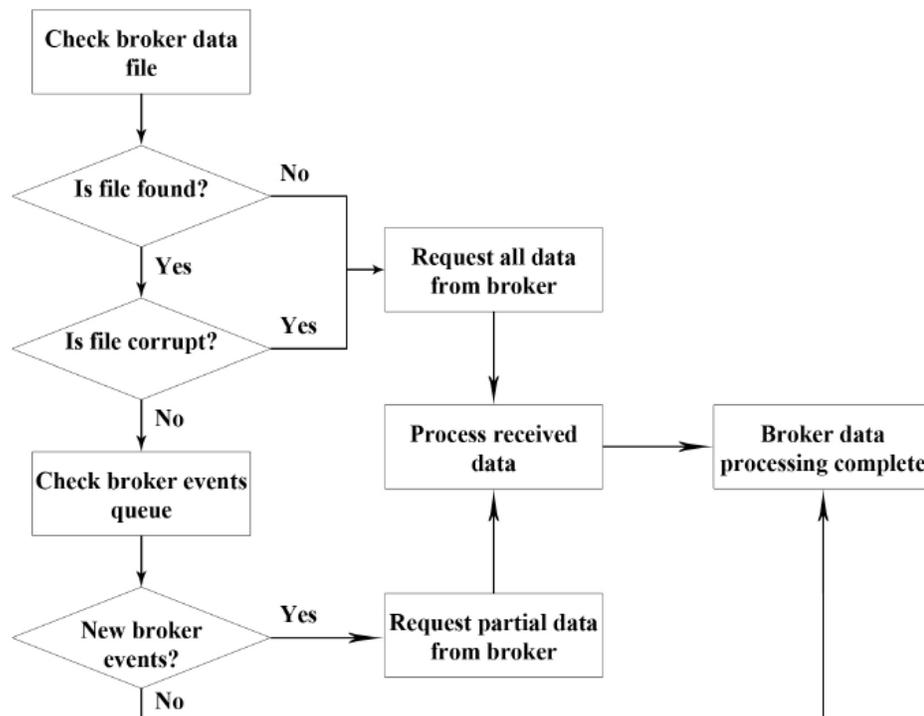


Figure 1. Broker data collection procedures performed at agent startup

Every time the agent is started, it loads the broker information from the broker data file. When it is the first time the agent has been started, the broker data file does not exist, so all broker data is requested from the broker, and one file is created with the received information. This information is also requested when the broker data file is corrupted.

If the broker data file is read successfully, the agent proceeds to check the reply queue for broker event messages. If the messages indicate that a broker object, such as a message flow, has changed, the agent requests information about only

the changed object from the broker. As only information about changed objects is requested, agent startup times and broker overhead are greatly reduced.

Remember: The trace level and accounting/statistics settings are still collected every time the agent is started. However, the overhead placed on both the agent and broker by collecting this data is small.

Requesting all data from the broker, which is done when the agent is started for the first time, can place a significant overhead on the broker because of the large volume of data requested. From WebSphere Message Broker Monitoring agent version 7.0 onwards, the agent no longer requests all data simultaneously. Instead, requests for data are sent to the broker in stages. After sending a request for information (for example, about a single broker component) to the broker, the agent waits for a response before sending the next request.

By using this broker data collection method, the broker processes only one message from the agent at a time, instead of receiving a large number of requests simultaneously. This method greatly reduces the performance overhead on the broker. However, the total time required for version 7.0 and later versions of the agent to collect all required information might be longer than with previous versions of the agent.

Historical data collection

You can use the historical data collection function of Tivoli Enterprise Portal to store data collected by the WebSphere Message Broker Monitoring agent. You can define the following properties of historical data collection:

- The interval at which data is collected
- The interval at which data is stored in a data warehouse (if you choose to do so)
- The location (either at the agent or at the Tivoli Enterprise Monitoring Server) at which the collected data is stored

For information about how to collection historical data for WebSphere Message Broker Monitoring agent, see Chapter 6, “Collecting historical data,” on page 91.

IBM Tivoli Monitoring

IBM Tivoli Monitoring manages system and network applications on several operating systems and keeps track of the availability and performance of all parts of your enterprise. It provides IBM Tivoli OMEGAMON XE products with a common agent-server-client architecture:

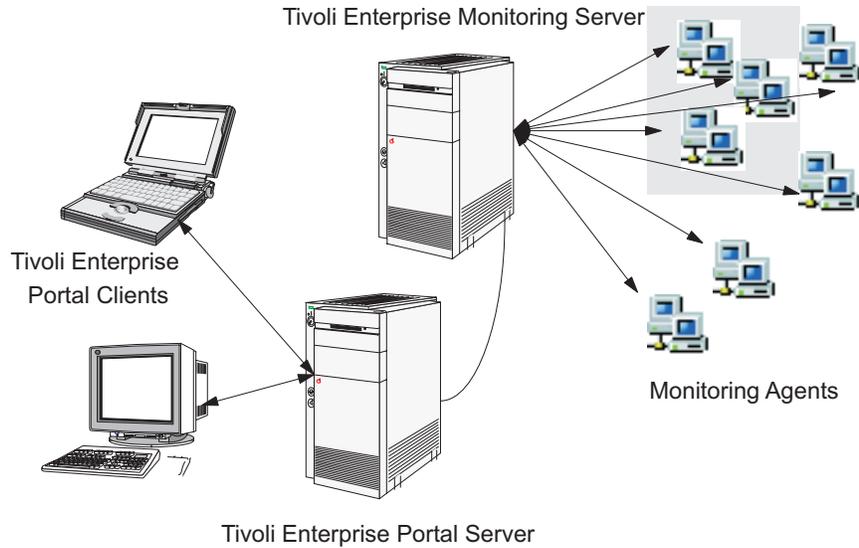


Figure 2. Agent-Server-Client architecture

Tivoli Enterprise Monitoring Server

Tivoli Enterprise Monitoring Server (monitoring server) gathers data from the Tivoli Enterprise Monitoring agent (monitoring agent), and acts as a collection and control point for alerts that are received from the agents. The Tivoli Enterprise Monitoring Server sends the data that it receives from the agents to Tivoli Enterprise Portal clients, where it is displayed in tabular or graphic views in a set of predefined or customized workspaces. The monitoring server also accepts requests for information or action from Tivoli Enterprise Portal clients and distributes them to the agents for processing.

Tivoli Enterprise Portal

Tivoli Enterprise Portal (portal) is the Java-based interface to the data monitoring and management resources of IBM Tivoli Monitoring. Depending on how it is installed, Tivoli Enterprise Portal can be used as either a desktop or browser-based client.

Tivoli Enterprise Portal has its own server, Tivoli Enterprise Portal Server (portal server). Tivoli Enterprise Portal Server performs common Tivoli Enterprise Portal functions, which reduces the processing performed by the Tivoli Enterprise Portal client.

Tivoli Enterprise Monitoring agents

Tivoli Enterprise Monitoring agents (monitoring agents) collect system or application data from *monitored*, or *managed* systems. For example, you can use the WebSphere MQ Monitoring agent to collect and analyze WebSphere MQ-specific data for all your remote and local queue managers from a single vantage point. The data is passed to Tivoli Enterprise Monitoring Server, and displayed in the Tivoli Enterprise Portal client.

The monitoring agents can also compare the current values of monitored properties against a set of defined conditions, and trigger alerts or actions when

those conditions occur. They can accept and perform requested actions that are relayed to them from Tivoli Enterprise Portal clients by the monitoring server.

Configuration agents can create and configure objects. The WebSphere MQ Configuration agent can configure objects such as WebSphere MQ queue managers and all their components (queues, channels, processes, and other objects).

IBM Tivoli OMEGAMON DE product overview

The IBM Tivoli OMEGAMON DE feature package for Tivoli Enterprise Portal offers a process-driven view of your enterprise. You can use this software to bring together information from disparate sources in one workspace, including a range of operating systems, servers, databases, mainframes, and network and Internet components. Tivoli OMEGAMON DE software provides a single point of control from which you can manage all the resources that your business-critical applications rely on.

Tivoli OMEGAMON DE extends the capabilities of Tivoli OMEGAMON XE to include the following views:

- Enterprise-specific Navigator views
The Navigator physical view presents the hierarchy of your managed enterprise by operating system and type of Tivoli Enterprise Monitoring agents. The Navigator business view offered by Tivoli OMEGAMON DE presents the hierarchy of managed objects. You can also define Navigator views for any logical groupings, such as business processes or by departmental hierarchy.
- Views of data from different types of monitoring agents in one workspace
In a single workspace, you can build a table or chart with data from one type of monitoring agent, and another table or chart with data from a different agent. Within that workspace, you can display views from as many different agent types as are included on that branch of the Navigator.
- Linking application workspaces
You can define links from a workspace associated with one type of monitoring agent to workspaces associated with other types of agents.

Policy management

The Tivoli Enterprise Portal policy management solution incorporates all the features of Tivoli OMEGAMON DE policy management and adds automation capabilities with the Workflow editor. You can use the Workflow editor to design sets of automated system processes, called policies, to resolve system problems. A policy performs actions, schedules work to be performed by users, or automates manual tasks.

Chapter 2. Customizing the monitoring agent

When you install the WebSphere Message Broker Monitoring agent, configuration parameters for the monitoring agents are set to their default values. This section describes the parameter file, and includes instructions for how to change parameter values if required.

Agent parameter file

The parameters that determine the operational and monitoring characteristics of an agent are stored in an XML file, which is created during the installation process of WebSphere Message Broker Monitoring agent.

After installation, you can view and edit the agent parameter settings file at any time. You can also add optional configuration parameters to the file.

Names and locations of agent parameter files

This section describes the name and location of the agent parameter file on different operating systems.

Agent parameter files on z/OS systems

The agent parameter file member name is KQIXML. By default, it is installed in the following partitioned data set:

```
&rhilev.RKANDATV
```

Agent parameter files on UNIX and Linux systems

The agent parameter file name is `kqi.xml`.

If you specify `broker` and `agentId` parameters when the agent is started, the name of the agent parameter file is modified. The agent parameter file name for an agent started with optional `broker` and `agentId` parameters has the following form:

```
<hostName>_qi_<brokerName>_##_<agentId>.xml
```

If you specify only the `agentId` parameter and not `broker` parameters when the agent is started, then the agent parameter file name has the following form:

```
<hostName>_qi_<agentId>.xml
```

The agent parameter file is located in the `<install_dir>/config` directory, where `<install_dir>` is the directory where IBM Tivoli Monitoring is installed. The default directory is `/opt/IBM/ITM`.

Agent parameter files on Windows systems

The agent parameter file name is `kqi.xml`.

If you use the **Create Instance** option in the Manage Tivoli Monitoring Services window to replicate the agent, the name of the agent parameter file is modified. The agent parameter file name for a file associated with an agent has the following form:

kqi_<instancename>.xml

The agent parameter file is located in the <install_dir>\TMAITM6 directory, where <install_dir> is the directory where IBM Tivoli Monitoring is installed. The default directory is C:\IBM\ITM.

Default agent parameter file

The default agent parameter file contains a base set of parameters that are configured to their default values. However, you can modify the values of these parameters to suit the needs of your environment and you can add more parameters.

The default file on UNIX, Linux, and Windows systems

When you first install the WebSphere Message Broker Monitoring agent, the default kqi.xml file resembles the file that is displayed in Figure 3.

This default file contains the core parameters that control reporting and monitoring by the agent on UNIX, Linux, and Windows systems. Because no individual brokers or queue managers are specified in the file, using the default parameters displayed in this example results in all brokers being monitored by the agent.

```
<KqiAgent version="710"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="60"
  defaultFlowEventInterval="15"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  defaultRetainRecentResourceSamples="1"
  holdTimeForQuery="180"
  defaultReplyQueueName="KQI.AGENT.REPLY.QUEUE"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="NO"
  maximumMessageLength="10240"
  defaultPersistentBrokerData="NO"
  defaultRefreshInterval="300"
  defaultTakeActionAuthUsers="*"
>

</KqiAgent>
```

Figure 3. Default kqi.xml file on Windows, UNIX, and Linux systems

The default file on z/OS systems

When you first install the WebSphere Message Broker Monitoring agent, the default KQIXML file resembles the file displayed in Figure 4 on page 11.

On z/OS systems, you must specify all monitored brokers in the parameter file using the <MonitorBroker> tag. Using the parameters displayed in Figure 4 on page 11 results in the M60ABRK broker being monitored by the agent. To monitor additional brokers, add additional <MonitorBroker> tags as necessary.

```

<KqiAgent version="710"
  agentid="BUT1"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="60"
  defaultFlowEventInterval="15"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  defaultRetainRecentResourceSamples="1"
  holdTimeForQuery="180"
  defaultReplyQueueName="KQI.AGENT.REPLY.QUEUE"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="NO"
  maximumMessageLength="10240"
  defaultPersistentBrokerData="NO"
  defaultRefreshInterval="300"
  defaultTakeActionAuthUsers="*"
  persistentDataPath="/cdl/usr/CandleHome/data" >
<MonitorBroker name="M60ABRK"
  alias="M60ABRK"
  envfileDirectory="/var/wmqi/M60ABRK">
</MonitorBroker>
</KqiAgent>

```

Figure 4. Default KQIXML file on z/OS systems

Agent parameter file syntax

The agent parameter file contains a single pair of top-level <KqiAgent></KqiAgent> tags. It might also contain one or more instances of the <MonitorBroker></MonitorBroker> and <ConnectQueueManager></ConnectQueueManager>. Agent parameters are defined using attributes of these elements.

The parameters in the kqi.xml file (or KQIXML member on z/OS systems) are stored in XML format. You must adhere to XML syntax conventions when modifying the file. All attribute values must be enclosed in double quotation marks. However, the coding is not column-specific, so the spacing and line separation displayed in the sample files are not critical.

Setting agent parameters

You can change the agent configuration by modifying the default parameter values listed in the agent parameter file (see “Modifying an existing parameter” on page 12), or adding new parameters to it (see “Adding new parameters” on page 12). For detailed description about the configuration parameters, see “Agent parameter descriptions” on page 18.

Modifying an existing parameter

To modify a parameter in the agent parameter file, do the following steps:

1. Open the agent parameter file in a standard text editor.
2. Substitute the original value with your new value for the parameter that you want to modify.

Remember: The parameter value must be enclosed in a set of double quotation marks (" ").

3. Save and close the parameter file.

Example:

Original parameter:

```
defaultStatisticInterval="original_value"
```

Modified parameter:

```
defaultStatisticInterval="new_value"
```

Adding new parameters

To add new parameters to the agent parameter file, you must first determine whether an appropriate XML tag exists for the parameter.

- If the XML tag that the attribute belongs to already exists, insert the new parameter string into the tag block. The following examples illustrate how to add an **agentId** parameter to a shortened version of the default `kqi.xml` file:

Original version:

```
<KqiAgent version="710"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="120"
  defaultFlowEventInterval="30"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  holdTimeForQuery="180"
  defaultReplyQueueName="SOME.NAME"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="NO"
  maximumMessageLength="10240"
  defaultPersistentBrokerData="NO"
  defaultRefreshInterval="300"
  defaultTakeActionAuthUsers="*">
</KqiAgent>
```

Modified version:

```
<KqiAgent version="710"
  agentId="new_value"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="120"
  defaultFlowEventInterval="30"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
```

```

    holdTimeForQuery="180"
    defaultReplyQueueName="SOME.NAME"
    defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
    defaultCollectNodeData="NO"
    maximumMessageLength="10240"
    defaultPersistentBrokerData="NO"
    defaultRefreshInterval="300"
    defaultTakeActionAuthUsers="*"
  </KqiAgent>

```

- If the appropriate XML tag does not already exist, insert the correct opening and closing XML tags before adding the parameter string.

For example, to set the **MonitorBroker** parameters, insert the `<MonitorBroker></MonitorBroker>` tags, and then add the parameters as follows:

Original version:

```

<KqiAgent version="710"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="120"
  defaultFlowEventInterval="30"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  holdTimeForQuery="180"
  defaultReplyQueueName="SOME.NAME"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="NO"
  maximumMessageLength="10240"
  defaultPersistentBrokerData="NO"
  defaultRefreshInterval="300"
  defaultTakeActionAuthUsers="*"
</KqiAgent>

```

Modified version:

```

<KqiAgent version="710"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="120"
  defaultFlowEventInterval="30"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  holdTimeForQuery="180"
  defaultReplyQueueName="SOME.NAME"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="NO"
  maximumMessageLength="10240"
  defaultPersistentBrokerData="NO"
  defaultRefreshInterval="300"
  defaultTakeActionAuthUsers="*"
<MonitorBroker name="MySpecialBrokerName"
  alias="BrokerAlias"
  statisticInterval="60"
  flowEventInterval="20"
  retainBrokerEvents="5"
  retainFlowEvents="5"
  collectNodeData="NO"
  takeActionAuthUsers="A?B,C*"
</MonitorBroker>

```

```

    <MonitorBroker name="AnotherBrokerName"
      collectNodeData="NO" >
    </MonitorBroker>
    <ConnectQueueManager name="MY.BROKER.QMGR"
      replyQueueName="SOME.OTHER.NAME"
      replyQueueModel="MY.SPECIAL.MODEL.QUEUE">
  </ConnectQueueManager>
</KqiAgent>

```

Tip: Each tag block encapsulates the parameters of a particular component. The `kqismp1.xml` sample file is included with UNIX, Linux, and Windows system versions of WebSphere Message Broker Monitoring agent; you can use this file as a template when you create your own `kqi.xml` file.

- On UNIX and Linux systems, the `kqismp1.xml` file is stored in the `<install_dir>/config` directory, where `<install_dir>` is the installation directory of IBM Tivoli Monitoring. The default installation directory is `/opt/IBM/ITM`.
- On Windows systems, the `kqismp1.xml` file is stored in the `<install_dir>\TMAITM6` directory, where `<install_dir>` is the installation directory of IBM Tivoli Monitoring. The default installation directory is `C:\IBM\ITM`.

Remotely configuring the WebSphere Message Broker Monitoring agent

On distributed systems such as Windows and UNIX operating systems, the agent parameter file is encoded using UTF-8 and can contain non-English language characters. On these systems, you can use the remote agent configuration feature instead of editing this file directly. On z/OS systems, however, the file is encoded using EBCDIC and must be edited manually.

You can remotely configure the WebSphere Message Broker Monitoring agent through Tivoli Enterprise Portal, or through the command line. See the following topics for detailed instructions:

- “Prerequisites”
- “Remotely configuring through Tivoli Enterprise Portal” on page 15
- “Remotely configuring through the command line” on page 16

Prerequisites

Before you can remotely deploy the WebSphere Message Broker Monitoring agent, identify the operating system where you want to remotely configure the agents and ensure that the following requirements are fulfilled in your monitoring environment:

- An OS agent is installed or deployed on the computer where you want to remotely configure the agents. For example, if you want to deploy the agents on the computer where the Windows operating system is running, you must have the monitoring agent for Windows OS installed on that computer. And the OS agent must be configured to connect to the same Tivoli Enterprise Monitoring Server as the WebSphere Message Broker Monitoring agent connects to. For information about how to install the OS agent, see *IBM Tivoli Monitoring Installation and Setup Guide, SC32-9407*.
- The agent depot is populated on the Tivoli Enterprise Monitoring Server from which you configure agents across your environment. For example, if you want to remotely configure the WebSphere Message Broker Monitoring agent, you must have the WebSphere Message Broker Monitoring agent depot populated on the Tivoli Enterprise Monitoring Server. For detailed information about how to

populate the agent depot, see "Populating your agent depot" in Chapter 12, Deploying monitoring across your environment from a central location in *Installation and Setup Guide, SC14-7520*.

Remotely configuring through Tivoli Enterprise Portal

To remotely configure the WebSphere Message Broker Monitoring agent through Tivoli Enterprise Portal, do the following steps:

1. Log on to the Tivoli Enterprise Portal.
2. In the Navigator view, navigate to the computer where you want to remotely configure the agent.
3. Right-click the Navigator item of the agent that you want to configure, and then click **Configure** from the menu. The Manage System Configuration window is opened as displayed in Figure 5.

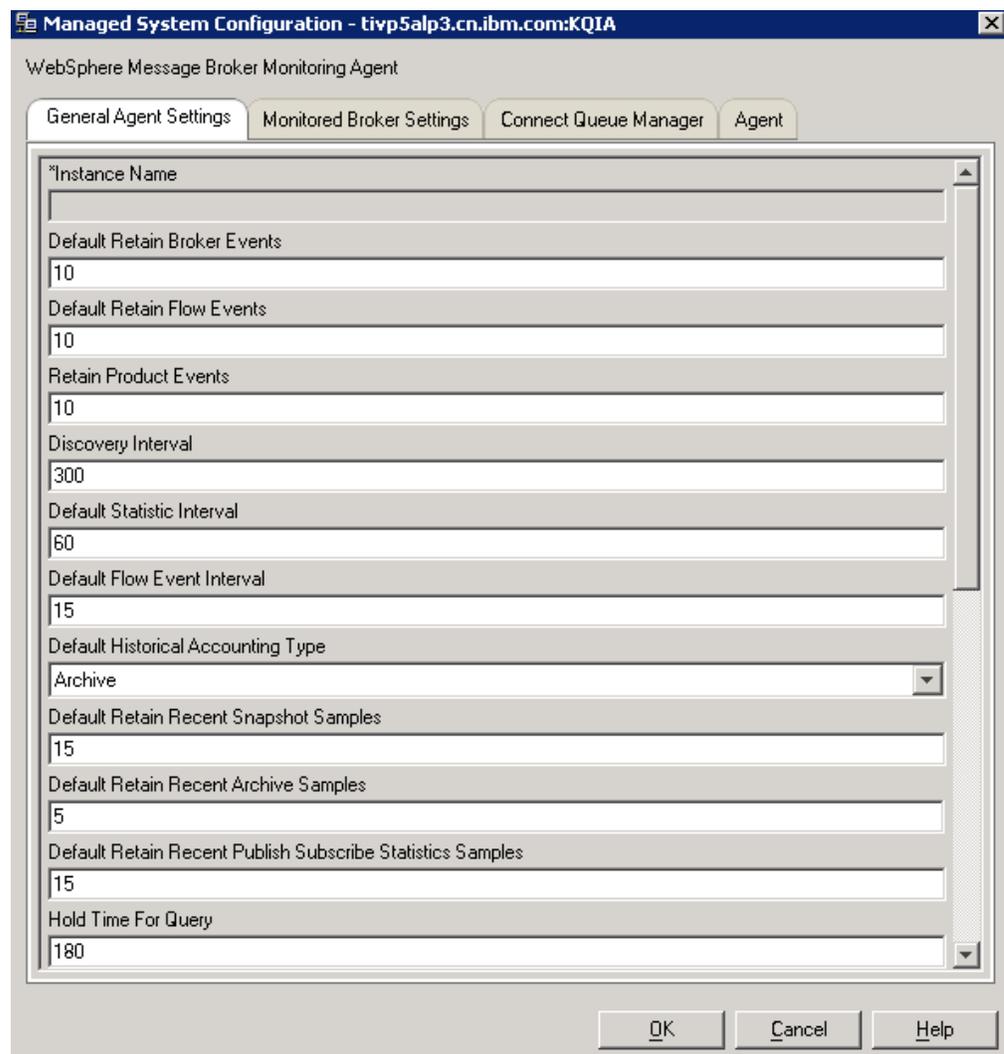


Figure 5. Managed System Configuration window

4. Provide the configuration information for agent in the required tabs.

Tips:

- The parameters for the **General Agent Settings** tab are the same as those parameters that are enclosed in the KqiAgent tag, except that there is an **Instance Name** parameter in the **General Agent Settings** tab, which indicates the name of the agent instance that you are configuring.
 - The parameters for the **Monitored Broker Settings** tab are the same as those enclosed in the MonitorBroker tag.
 - The parameters for the **Connect Queue Manager** tab are the same as those enclosed in the ConnectQueueManager tag.
 - To get detailed information about the parameters for the **Agent** tab, at the lower right corner in the window, click **Help**.
5. When you finish the configuration, click **OK** to close the window.

Remotely configuring through the command line

Use the **tacmd configureSystem** command to remotely edit the configuration options of an agent. The **tacmd configureSystem** command has the following syntax:

```
tacmd configureSystem
      {-m | --system} system
      {-p | --property} section.name=value ...
```

To remotely configure the WebSphere Message Broker Monitoring agent from the command line, do the following procedure on the Tivoli Enterprise Monitoring Server to which the WebSphere Message Broker Monitoring agent is connected:

1. From a command prompt, change to the appropriate directory:

- on Windows systems: *ITM_home\bin*
- on UNIX or Linux systems: *ITM_home/bin*

where *ITM_home* is the directory where IBM Tivoli Monitoring is installed.

2. Run the **tacmd login** command to log on to the Tivoli Enterprise Monitoring Server, to which the agent that you want to remotely configure reports.

Provide the **-m** option with the host name of the system where the monitoring server is installed, and use the **-u** and **-p** options to specify the authenticated user.

The following command logs on to the Tivoli Enterprise Monitoring Server on the *hostname* system with the *administrator* user ID, the *mypassword* password, and a login expiration time of 1440 minutes.

```
tacmd login -s hostname -u administrator -p mypassword -t 1440
```

3. Run the **tacmd configureSystem** command to remotely configure the WebSphere Message Broker Monitoring agent.

The following command configures a secondary agent instance whose **agentId** parameter is specified to *test* in the configuration file on the *AF66359F.cn.ibm.com* system.

```
tacmd configureSystem -m test:AF66359F.cn.ibm.com:KQIA -p
KqiAgent.defaultRetainBrokerEvents=5
```

The following command configures the primary agent instance on the *AF66359F.cn.ibm.com* system by setting the value of the **defaultRetainBrokerEvents** parameter in the KqiAgent tag to 5.

```
tacmd configureSystem -m AF66359F.cn.ibm.com:KQIA -p
KqiAgent.defaultRetainBrokerEvents=5
```

Important: To ensure that the agent can be configured successfully, you must provide the **-m** option with the correct managed system name. The appropriate managed system name for each agent is displayed at the bottom of the

workspace when you click the Navigator item of the agent instance in the Navigator view of Tivoli Enterprise Portal. You can also use the **tacmd listSystems** command to list the existing managed systems. The following command lists all the systems in your enterprise with the product code *QI* (WebSphere Message Broker Monitoring agent). Use the managed system name that ends in **KQIA**.

```
tacmd listSystems -t QI
```

For detailed descriptions about the configuration parameters, see “Agent parameter descriptions” on page 18.

Creating multiple instances of the WebSphere Message Broker Monitoring agent

By default, the WebSphere Message Broker Monitoring agent that is created during installation monitors all brokers on your system. If you want to monitor one broker other than all brokers on the system, you can create a secondary agent instance.

Windows systems: Creating multiple instances of the monitoring agent

Perform the following steps to create a secondary instance of the WebSphere Message Broker Monitoring agent on Windows systems:

1. From the **Start** menu, click **Programs > IBM Tivoli Monitoring > Manage Tivoli Monitoring Services**. The Manage Tivoli Enterprise Monitoring Services window is displayed.
2. In the Manage Tivoli Enterprise Monitoring Services window, right-click **WebSphere Message Broker Monitoring Agent**, and then click **Create Instance** from the menu.
3. Enter a name for the instance when you are prompted and click **OK**. The new agent instance is created and is displayed in the Manage Tivoli Enterprise Monitoring Services window.
4. In the Manage Tivoli Enterprise Monitoring Services window, right-click the newly created agent instance, and then click **Configure Using Defaults**.
5. When you are asked whether to update the configuration file of the agent instance before configuring the agent. Click **Yes**.
6. A message is displayed stating that configuration will wait for you to close your default text editor before continuing. Click **OK**. The configuration file is opened in your default text editor.
7. Edit the configuration file as follows:
 - a. Ensure that the value of the **agentId** parameter is different to all other instances of the agent running on the same system.
 - b. Use the **<MonitorBroker>** tag to specify which brokers the agent monitors. If this tag is not set, all brokers on the system are monitored. For further information about the **<MonitorBroker>** tag, see “MonitorBroker” on page 28.

For more information about the related parameters, see “agentId” on page 19 and “MonitorBroker” on page 28.

8. Save and close the file.
9. A message is displayed stating that the configuration file edit session is complete. Click **Yes** to configure the agent.

The secondary agent instance is now configured and ready to start.

UNIX and Linux systems: Creating multiple instances of the monitoring agent

Do the following steps to create a secondary instance of the WebSphere Message Broker Monitoring agent on UNIX and Linux systems:

1. Navigate to the *install_dir/bin* directory, where *install_dir* is the installation directory of IBM Tivoli Monitoring.
2. To create a secondary instance of the WebSphere Message Broker Monitoring agent, run the **itmcmd agent start** command with the **-o** and **-p** options.

```
./itmcmd agent -o agent_ID -p broker_name start qi
```

where *agent_ID* is a unique 4-digit alphanumeric agent ID, and *broker_name* is name of the monitored broker.

- **-o**
Specifies the monitoring agent ID (maximum of four characters).
- **-p**
Specifies the broker name (optional).

A secondary agent instance with the *agent_ID* name is created and it monitors the broker that you specified in the **itmcmd agent start** command.

Agent parameter descriptions

The parameters that you can set in the *kqi.xml* file (or in the *KQIXML* member on *z/OS* systems), and their associated XML tags, are described in the following sections. All the parameters are attributes of one of the three tags: *KqiAgent*, *MonitorBroker*, and *ConnectQueueManager*.

On distributed systems, the *KqiAgent* tag is required. The *MonitorBroker* and *ConnectQueueManager* tags are optional. If you only specify the *KqiAgent* tag in the agent parameter file, all brokers running on the same host system as the monitoring agent are monitored. However, if the monitored broker name contains non-alphanumeric ASCII characters, you must specify the **alias** attribute in the *MonitorBroker* tag. The alias is displayed as the managed system name of the broker on Tivoli Enterprise Portal. Broker names that contain non-alphanumeric ASCII characters might be displayed under wrong names on Tivoli Enterprise Portal.

On *z/OS* systems, the *KqiAgent* and *MonitorBroker* tags are required. Use the *KqiAgent* tag to specify parameters related to the monitoring agent itself. Use the *MonitorBroker* tag to specify monitoring attributes for each broker to be monitored.

Remember (non z/OS only): On a non *z/OS* system, if there are multiple versions of brokers on the same system including WebSphere Message Broker V6 (all releases) and WebSphere Message Broker V7 or later, using only one agent instance to monitor all versions of brokers on the system might cause the following problems:

- Situations does not work. Alerts cannot be issued when the situation conditions are met.
- Historical data cannot be collected for version 6 (all releases) brokers if they exist.

To avoid these problems, use one agent instance to monitor one specific version of brokers. For example, use one agent instance to monitor all version 7 or later brokers and use another instance to monitor all version 6 (all releases) brokers. In this case, the MonitorBroker tag is required to specify each broker name of the given version.

If there is no WebSphere Message Broker V7 or later in your environment, you can still use one agent instance to monitor all brokers on the same system.

KqiAgent

You can use the KqiAgent tag to specify parameters that are related to the agent itself. Any attributes that follow this tag apply to the agent as a whole and provide monitoring and connection defaults that can be overridden at the individual broker or queue manager level. All attributes are optional.

version

The **version** attribute specifies the version of WebSphere Message Broker Monitoring agent for which the parameters are set. This attribute is reserved for future product use. Do not alter this attribute from its default value unless instructed to do so by IBM Software Support.

agentId

The **agentId** attribute provides a short identifier (maximum length of 4 alphanumeric characters) for the WebSphere Message Broker Monitoring agent.

You must assign a unique agent ID to each agent in the following circumstances:

- You are running more than one WebSphere Message Broker Monitoring agent on the same system.
- You are running more than one monitored broker with the same name on different systems.

Unique agent IDs differentiate the broker-managed system names at the Tivoli Enterprise Monitoring Server, so that Tivoli Enterprise Portal presents them correctly.

This attribute is not set by default. To set it, supply an alphanumeric character string with no spaces. The value is used to create a unique managed system name.

defaultRetainBrokerEvents

The **defaultRetainBrokerEvents** attribute determines how many broker events to retain per broker. Report information is available only for retained broker events. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically.

The default number of events to retain is 10. This value must be greater than 0.

defaultRetainFlowEvents

The **defaultRetainFlowEvents** attribute determines how many message flow events to retain per broker. Report information is available only for retained message flow events. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically.

The default number of events to retain is 10. This value must be greater than 0.

retainProductEvents

The **retainProductEvents** attribute determines the number of product events to retain. Report information is available only for retained product events. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically.

The default number of events to retain is 10. This value must be greater than 0.

discoveryInterval

The **discoveryInterval** attribute determines the interval (in seconds) at which the agent attempts to discover new brokers that are created on the system. This attribute must be greater than 0; the default value is 300 seconds. This attribute can have a large value, for example, 86400 seconds (24 hours) or even longer when you are no longer creating new brokers. However, if you have a dynamic environment in which new brokers are created frequently, you must adjust this attribute accordingly.

defaultStatisticInterval

The **defaultStatisticInterval** parameter determines the minimum interval at which broker statistics are sampled, in seconds.

This parameter value controls the data collection interval of the following workspaces:

- WebSphere Message Broker V7.0 or later:
 - Monitor Node Base Statistics
 - Monitor Node Broker Statistics
 - Monitor Node Execution Group Statistics
 - Monitor Node Message Flow Statistics
 - Monitor Node Sub-Flow Statistics
- WebSphere Message Broker V6 (all releases):
 - Broker Statistics
 - CandleMonitor Node Statistics
 - Execution Group Statistics
 - Message Flow Statistics
 - Sub-Flow Statistics

If you are collecting historical data for one of the statistics attribute groups, the historical data collection interval must be a multiple of the **defaultStatisticInterval** value; otherwise, historical data might be inaccurate. For example, if you configure historical data collection for one of the statistics attribute groups to run every 5 minutes (300 seconds), the **defaultStatisticInterval** value must be a submultiple of 300, such as 60 (1 minute), 75 (1.25 minutes), 100 (1.67 minutes), 150 (2.5 minutes), or 300 (5 minutes). Additionally, for accurate results, the historical data collection interval of all statistics attribute groups for which historical collection is being collected must be the same.

This value must be greater than 0; the default value is 60 seconds.

Tip: Do not specify a value that is less than 60 to the **defaultStatisticInterval** parameter. Otherwise, the data amount might be enormous.

defaultFlowEventInterval

The **defaultFlowEventInterval** attribute determines the interval at which message flow events are sampled, in seconds. This interval does not apply to broker events or product events.

This value must be greater than 0; the default value is 15 seconds.

defaultHistoricalAccountingType

For WebSphere Message Broker V7.0 or later, the **defaultHistoricalAccountingType** attribute tells the agent whether to collect archive accounting data historically. For WebSphere Message Broker V6 (all releases), this attribute tells the agent which types of accounting data (snapshot and archive) to log historically. The value of this attribute has no effect unless historical data collection for any of the four accounting attribute groups has been enabled in Tivoli Enterprise Portal. This attribute can have the following values:

- None
No accounting data is logged historically, even if historical data collection is enabled in Tivoli Enterprise Portal.
- Archive
Only Archive accounting data is logged historically. This is the default setting.
- All (WebSphere Message Broker V6 all releases only)
All accounting data is logged historically.
- Snapshot (WebSphere Message Broker V6 all releases only)
Only Snapshot accounting data is logged historically

Important: Use All and Snapshot with caution. The amount of data can be enormous. If there is not enough disk space for the data to be stored on the system, there might be no snapshot accounting data displayed in the related workspaces.

Tip: When you enable historical data collection on Tivoli Enterprise Portal globally, you can set this attribute to None to disable historical data collection on some specific broker. To do this, you must have configured one agent instance to monitor the broker only.

defaultRetainRecentSnapshotSamples

The **defaultRetainRecentSnapshotSamples** attribute specifies the default number of recent snapshot records that are stored by the agent for any given message flow. The default value of this attribute is 15, which is equivalent to approximately 5 minutes of snapshot data. However, the agent always ensures that it retains enough historical samples to log the data historically if historical data collection is enabled.

You can set this attribute to 0. When the **defaultRetainRecentSnapshotSamples** value is 0, the WebSphere Message Broker Monitoring agent does not store any recent snapshot records. The following workspaces will be empty:

- WebSphere Message Broker V7.0 or later:
 - Snapshot Accounting Message Flow Statistics
 - Snapshot Accounting Node Statistics
 - Snapshot Accounting Terminal Statistics
 - Snapshot Accounting Thread Statistics
- WebSphere Message Broker V6 (all releases):
 - Snapshot Message Flow Accounting

- | – Snapshot Node Accounting
- | – Snapshot Terminal Accounting
- | – Snapshot Thread Accounting

defaultRetainRecentArchiveSamples

The **defaultRetainRecentArchiveSamples** attribute specifies the number of recent archive records that are stored by the agent for any given message flow. The greater this value, the more data is available in the Archive Accounting Message Flow Statistics workspace/Archive Message Flow Accounting workspace.

The default archive interval for a broker is 60 minutes. The default value of the **defaultRetainRecentArchiveSamples** attribute is 5, so 5 hours of recent archive data can be viewed in the Archive Accounting Message Flow Statistics workspace/Archive Message Flow Accounting workspace if the default archive interval has not been changed. The agent always ensures that it retains enough historical samples to log the data historically if historical data collection is enabled.

You can set this attribute to 0. When the **defaultRetainRecentArchiveSamples** value is 0, the WebSphere Message Broker Monitoring agent does not store any recent archive records. The following workspaces will be empty:

- WebSphere Message Broker V7.0 or later:
 - Archive Accounting Message Flow Statistics
 - Archive Accounting Node Statistics
 - Archive Accounting Terminal Statistics
 - Archive Accounting Thread Statistics
- WebSphere Message Broker V6 (all releases):
 - Archive Message Flow Accounting
 - Archive Node Accounting
 - Archive Terminal Accounting
 - Archive Thread Accounting

defaultRetainRecentPubSubSamples

The **defaultRetainRecentPubSubSamples** attribute specifies the minimum number of recent publish-subscribe data records stored by the agent. The higher this value is, the more data is available in the Publish-Subscribe Statistics workspace. The agent always ensures that it retains enough recent samples to log all data historically if historical data collection is enabled, so more than the number of records specified by this attribute might be retained.

The value of this attribute must be greater than 0. The default value is 15.

defaultRetainRecentResourceSamples

The **defaultRetainRecentResourceSamples** attribute specifies whether to store recent resource records for message flows. This attribute value can be either 1 or 0. The default resource interval for a broker is 20 seconds. So when this attribute is set to 1, 20 seconds of resource statistics data can be displayed in the Resource Statistics workspaces. To view more data about resource statistics, enable historical data collection on the concerned attribute group.

When this attribute is set to 0, the WebSphere Message Broker Monitoring agent does not subscribe the resource statistics events that are emitted by WebSphere Message Broker and the Resource Statistics workspaces are empty.

This attribute is valid for WebSphere Message Broker 7.0 or later only.

defaultRetainMonitoringEvents

The **defaultRetainMonitoringEvents** attribute determines how many monitoring events to retain per broker. Report information is available only for retained broker monitoring events. The retained monitoring events are displayed in the Monitoring and Audit Events workspace. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically.

The default number of events to retain is 10. This value must be greater than or equal to 0. You can set this attribute to 0 only for WebSphere Message Broker 7.0. When the **defaultRetainMonitoringEvents** value is 0, the WebSphere Message Broker Monitoring agent will not subscribe the monitoring events emitted by WebSphere Message Broker and the Monitoring and Audit Events workspace is empty.

holdTimeForQuery

The **holdTimeForQuery** attribute specifies the length of time, in seconds, that the agent must retain detailed accounting data. This data must be retained for viewing detailed information in accounting workspaces. The greater this value, the more data is available in the accounting workspaces.

The default value of this attribute is 180, which means that, regardless of other attributes related to data retention, you can view detailed information about a selected data sample for at least 3 minutes after the data sample was originally displayed. If you view the retained data in a workspace, every time you refresh the workspace the length of time for which the data is retained is extended by the **holdTimeForQuery** value.

This attribute must have a value greater than 0. If you do not want the agent to retain data, specify a value of 1 (1 second).

For example, the Snapshot Message Flow Accounting workspace presents a list of monitored message flows and their statistics. By default, for the next 3 minutes you can select a row in the table and link to a more detailed view of the displayed data (either Snapshot Thread Accounting or Snapshot Node Accounting). Each time that you view a data pertaining to a given message flow, you extend the time that you can view the sample by 3 minutes. If the sample is not retained for some other reason and if you do not access refresh the data for 3 minutes, the data is deleted. If the data is deleted, the data from the most recent sample is displayed instead.

defaultReplyQueueName

The **defaultReplyQueueName** attribute specifies the name of the queue that is used to receive publications and reply messages from the broker. A permanent dynamic queue is created using the model queue specified by “defaultReplyQueueModel” on page 24 if the name given does not pre-exist as a predefined or a permanent dynamic queue on the queue manager of the broker. This value can be overridden for individual queue managers using the **replyQueueName** attribute.

If this attribute is changed after starting a WebSphere Message Broker Monitoring agent, remove the subscription associated with the previous queue name and the previous queue.

The default value is KQL.AGENT.REPLY.QUEUE.

defaultReplyQueueModel

The **defaultReplyQueueModel** attribute specifies the name of the queue that is used as a model for creating the agent reply queue if the name given in the **defaultReplyQueueName** attribute does not pre-exist as a predefined or a permanent dynamic queue on the queue manager of the broker. The agent creates a queue with the name specified by the **defaultReplyQueueName** attribute for the queue manager of the broker, using the properties of the queue specified by the **defaultReplyQueueModel** attribute as a model. This value can be overridden for individual queue managers using the **replyQueueModel** attribute.

The model queue specified by this attribute must be a permanent dynamic model queue. If the **persistentBrokerData** attribute is set to YES, the model queue must enable persistent messages, otherwise, broker event messages that are received when the agent is stopped might be lost. If these event messages are lost, the workspace information is not up-to-date, because the broker uses these messages to detect changes in objects deployed in the broker environment while the agent is stopped. If you cannot use a permanent dynamic model queue with persistent messages enabled, use one of the following options:

- Set the **persistentBrokerData** attribute to No. This setting forces the agent to automatically detect all deployed objects at startup, ensuring that workspace information is always up-to-date. And this setting increases processing overhead on the message broker and in complex environments this might cause long delays in starting the WebSphere Message Broker Monitoring agent.
- When you know that the broker environment has been updated, issue the **QI Refresh Broker Data** take action command to update workspace information. Issuing this take action command causes all broker information to be updated instead of only information about updated components, and so might take a long time.

The default value is SYSTEM.BROKER.MODEL.QUEUE.

defaultTakeActionAuthUsers

The **defaultTakeActionAuthUsers** attribute specifies which Tivoli Enterprise Portal users are authorized to issue the Take Action commands that are associated with a WebSphere Message Broker Monitoring agent. You can specify multiple values for this attribute to authorize multiple users. You can also use the asterisk (*) and question mark (?) wildcard characters to authorize a group of users. Only Tivoli Enterprise Portal users who have an ID that matches the values of this attribute are authorized to issue Take Action commands handled by the WebSphere Message Broker Monitoring agent. The portal user IDs are defined at the Tivoli Enterprise Monitoring Server and do not necessarily exist on the system on which the agent is running. The value of this attribute applies only to the commands, which meet one of the following conditions:

- The command is not related to a particular broker.
- The command is related to a particular broker, and the broker does not have a specific list of users specified using the **takeActionAuthUsers** attribute.

The default value is an asterisk (*), which authorizes all Tivoli Enterprise Portal users to issue Take Action commands that are associated with this agent.

defaultCollectNodeData

The **defaultCollectNodeData** attribute determines whether node definition detail data collection and parsing are performed. Possible values are YES to enable this data or NO to disable it (case insensitive). This attribute can be overridden for individual message brokers using the "collectNodeData" on page 31 attribute.

If the **defaultCollectNodeData** attribute is set to NO, the following workspaces contains no data:

- WebSphere Message Broker V7.0 or later:
 - “Processing Node Attributes workspace” on page 284
 - “Message Flow Node Topology workspace” on page 278
- WebSphere Message Broker V6 (all releases):
 - “Message Processing Node Information workspace” on page 303
 - “Message Processing Node Attributes workspace” on page 302
 - “Message Flow Topology workspace” on page 301

In addition, the Queue Name and Queue Manager Name attributes in the following workspaces contains no data:

- WebSphere Message Broker V7.0 or later:
 - “Archive Accounting Node Statistics workspace” on page 272
 - “Snapshot Accounting Node Statistics workspace” on page 286
- WebSphere Message Broker V6 (all releases):
 - “Archive Node Accounting workspace” on page 291
 - “Snapshot Node Accounting workspace” on page 309

However, if message brokers in your environment have large message flows with a lot of message processing node data, enabling this option might significantly degrade system performance.

The default value is NO.

maximumMessageLength

The **maximumMessageLength** attribute specifies the maximum length of reply or event message from the broker that the agent processes. Its unit is in kilobytes. Any reply or event messages that exceed this length are processed, and so you cannot view the information that these messages contain in Tivoli Enterprise Portal. The default value is suitable for most environments. However, if your environment contains complex message flows, you might need to increase this value to ensure that all data is processed by the agent.

If an event message is truncated due to being longer than the **maximumMessageLength** value, the truncated message is still processed in order to report broker events. However, event details can be processed.

The default value is 10240 kilobytes. The maximum value is 102400.

defaultPersistentBrokerData

The **defaultPersistentBrokerData** attribute specifies whether data related to brokers is stored persistently by writing it to a file. Possible values are YES and NO (case insensitive). When this option is enabled, the file used to store the data might be large, depending on the volume of broker information that needs to be stored.

The default value is NO.

Remember:

- The agent must be restarted to make the change of this attribute value take effect. Before you start the agent again, delete the following items (if any) from your environment:

- Subscriptions that are not in use
- The reply queue of a broker
- If you change this attribute from YES to NO, also delete the existing persistent data file before restarting the agent.

Tip: The persistent data file is stored in the following directory:

- Windows: *install_dir*\TMAITM6\logs\History\KQI
- UNIX or Linux: *install_dir/arch/qi/hist*

where *install_dir* is the directory where IBM Tivoli Monitoring is installed. The default is C:\IBM\ITM on Windows systems or /opt/IBM/ITM on UNIX or Linux systems. *arch* is the architecture code of the operating system of this computer. For a list of architecture codes, see Appendix D, “Architecture codes,” on page 337.

- If you change this attribute from NO to YES, set the **defaultReplyQueueModel** attribute to the name of a permanent model queue. Make sure that the specified permanent model queue exists, such as SYSTEM.DURABLE.MODEL.QUEUE, which is automatically created by the broker.

defaultRefreshInterval

The **defaultRefreshInterval** attribute specifies the interval in seconds between requesting information not reported as broker events from the broker. This data is not related to the deployment of broker objects. This attribute includes trace and accounting/statistics settings for execution groups and message flows and broker subscription data. If you increase the length of the interval at which this data is collected, data in workspaces might be less up-to-date, but performance can be improved.

This data is only requested from the broker at this interval if a user is viewing the data in a workspace. If no user is viewing a workspace that contains this data, the data is not collected by the agent.

The default value is 300 seconds.

commandTimeoutInterval

The **commandTimeoutInterval** attribute specifies the amount of time (in seconds) that the agent waits for a response from a broker after sending a command before it issues a message indicating that the broker is not responding.

This value is not included in the configuration file by default, instead the value 300 is used. If specified, the value of this attribute must be greater than zero.

Do not modify this attribute unless directed to do so by IBM Software Support.

maximumCommandRetryCount

The **maximumCommandRetryCount** attribute specifies the number of times that the agent reissues commands after failing to receive a response from a broker before giving up.

This value is not included in the configuration file by default, instead the value 1 is used which means the agent does not issue commands after failing to receive the response from a broker. If specified, the value of this attribute must be greater than zero. If its value is set to be greater than 1, the agent issues commands for one

time less than the specified number of times before giving up. For example, if you set the attribute to be six, the agent issues five times after it fails to receive a response from a broker before giving up.

Do not modify this attribute unless you are directed to do so by the IBM Software Support.

maximumAgentCollectionThreads

The **maximumAgentCollectionThreads** attribute specifies the number of agent collection threads that are used by the agent for monitoring brokers. You can modify this value to increase or decrease the number of brokers that the agent can monitor.

This value is not included in the configuration file by default, instead the value 64 is used, which creates enough threads to monitor 10 brokers. If specified, the value of this attribute must be greater than zero.

On AIX[®] systems, do not attempt to monitor more than 10 brokers with a single agent. If you have to monitor more than 10 brokers, split them among several agents. See “Monitoring a large number of brokers” on page 34 for further details.

To use a single agent to monitor more than 10 brokers on operating systems other than AIX systems, increase this value by 6 for each additional broker monitored.

Do not modify this attribute unless absolutely necessary because monitoring a large number of brokers with a single agent might adversely affect monitoring performance. To monitor large number of brokers, create additional agents to monitor some brokers. See “Monitoring a large number of brokers” on page 34 for more details.

persistentDataPath

The **persistentDataPath** specifies the path of persistent data files on z/OS systems only. This value is set automatically during IBM Configuration Assistance Tool (ICAT) configuration to *candle home\data*, where *candle home* is the candle home directory specified by the user. However, this attribute has no default value and must be specified before starting the agent for the first time if not previously configured using ICAT.

defaultWMBInstallDirectory

The **defaultWMBInstallDirectory** attribute provides the installation directory of the brokers that you want to monitor to the WebSphere Message Broker Monitoring agent. The default value is blank. With the default configuration, the monitoring agent can automatically search for the installation directory and discover all brokers in that directory. Specify this attribute only when the monitoring agent cannot find the correct installation directory in a complex environment. You might have to configure this attribute in the following circumstances:

- The work path directory of WebSphere Message Broker (MQSI_WORKPATH) is not the default value.
- Both the 32-bit and 64-bit brokers are installed on a 64-bit Windows system at the same time.

The **defaultWMBInstallDirectory** attribute only applies to distributed systems. This attribute is not in the default *kqi.xml* file. You can add it in the *kqi.xml* file as needed.

Remember: You can specify only one directory for this attribute.

defaultWMQInstallDirectory

The **defaultWMQInstallDirectory** attribute provides the installation directory of WebSphere MQ to the WebSphere Message Broker Monitoring agent. Some JAR files of WebSphere MQ are required by the Configuration Manager Proxy (CMP) API to interact with the brokers. The default value is blank. With the default configuration, the monitoring agent can automatically search for the installation directory. Specify this attribute only when the monitoring agent cannot find the correct installation directory in a complex environment.

The **defaultWMQInstallDirectory** attribute only applies to distributed systems. This attribute is not in the default `kqi.xml` file. You can add it in the `kqi.xml` file as needed.

MonitorBroker

The `MonitorBroker` tag encapsulates parameters that apply to a single monitored broker. Specify one `MonitorBroker` tag for each broker.

On UNIX, Linux, and Windows systems, If no brokers are specified, all brokers running on the same host system as the agents are monitored, as determined by self-discovery. If any brokers are specified explicitly using this tag, any other brokers that you want to monitor must also be specified using `MonitorBroker` tags. All associated attributes are optional except for the **name** attribute.

On z/OS systems, WebSphere Message Broker Monitoring agent does not currently support self-discovery of brokers on z/OS systems. Each broker to be monitored must be explicitly specified using `MonitorBroker` tags and attributes. All associated attributes are optional except for the **name** and **componentDirectory** attributes.

Remember: Only alphanumeric ASCII characters can be included in a broker name on distributed systems, or EBCDIC characters on z/OS systems. Otherwise the navigator items might be displayed under wrong names on Tivoli Enterprise Portal. Alphanumeric ASCII character set consists of the numbers 0-9 , lowercase letters a-z, uppercase letters A-Z, and the underscore character (`_`). If the broker name contains a space, a comma or any other special characters, you must specify the **alias** attribute for the broker. The alias is used to be displayed on Tivoli Enterprise Portal.

name

The **name** attribute specifies the name of the broker monitored by the agent.

The **name** attribute is required for all `MonitorBroker` tags.

alias

The **alias** attribute provides an alternative name for a broker in addition to that specified using the **name** attribute. Only alphanumeric ASCII characters can be included in a broker name on distributed systems, or EBCDIC characters on z/OS systems. Otherwise the navigator items might be displayed under wrong names on Tivoli Enterprise Portal. If the broker name contains a space, a comma or any other special characters, you must specify the **alias** attribute for the broker.

If specified, the alias is displayed as the managed system name (node name) of the broker on Tivoli Enterprise Portal. You can use this attribute if you want to specify a more user-friendly name for your broker to simplify managing your system. This

is especially useful when the **name** attribute of a broker is exceptionally long or you have several brokers with similar names and you want to be able to distinguish them easily in Tivoli Enterprise Portal.

Remember: The alternative name must conform to the following rules:

- The maximum length is 22 characters.
- Only alphanumeric ASCII characters can be included on distributed systems, or EBCDIC characters on z/OS systems. Alphanumeric ASCII character set consists of the numbers 0-9, lowercase letters a-z, uppercase letters A-Z, and the underscore character (_).

For example, if you have a broker named BROKER1 on your system, you can specify an alias as follows:

```
<MonitorBroker name="BROKER1" alias="BK1"> </MonitorBroker>
```

envfileDirectory

This attribute is required only on z/OS systems and ignored if it is specified on other operating systems.

The **envfileDirectory** attribute specifies the z/OS UNIX system services (USS) directory that the broker ENVFILE file is located in.

statisticInterval

The **statisticInterval** attribute overrides the global **defaultStatisticInterval** attribute and determines the minimum interval at which broker statistics are sampled in seconds. If this attribute is not specified, the **defaultStatisticInterval** value is used instead. If this attribute is specified, the value must be greater than 0.

If you are collecting history data for one of the statistics attribute groups, the configured historical data collection interval must be a multiple of the **statisticInterval** value; otherwise, historical data collection might produce unpredictable results. For example, if you configure historical data collection for one of the statistics attribute groups to run every 5 minutes (300 seconds), the **statisticInterval** value must be a submultiple of 300, such as 60 (1 minute), 75 (1.25 minutes), 100 (1.67 minutes), 150 (2.5 minutes), or 300 (5 minutes). For accurate results, the historical data collection interval must be the same value for all statistics attribute groups for which historical data collection is enabled.

Tip: Do not specify a value that is less than 60 to the **defaultStatisticInterval** attribute. Otherwise, the data amount might be enormous.

flowEventInterval

The **flowEventInterval** attribute overrides the global **defaultFlowEventInterval** attribute and determines the interval at which message flow events are sampled, in seconds. If this attribute is not specified, the **defaultFlowEventInterval** value is used instead. If this attribute is specified, the value must be greater than 0.

retainBrokerEvents

The **retainBrokerEvents** attribute overrides the global **defaultRetainBrokerEvents** attribute and determines how many broker events to retain per broker. Report information is available only for retained broker events. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically. If this attribute is not specified, the **defaultRetainBrokerEvents** value is used instead. If the attribute is specified, the value must be greater than 0.

retainFlowEvents

The **retainFlowEvents** attribute overrides the global **defaultRetainFlowEvents** attribute and determines how many message flow events to retain per broker. Report information is available only for retained message flow events. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically. If this attribute is not specified, the **defaultRetainFlowEvents** value is used instead. If this attribute is specified, the value must be greater than 0.

takeActionAuthUsers

The **takeActionAuthUsers** attribute specifies which Tivoli Enterprise Portal users are authorized to issue the Take Action commands to a particular agent running on a particular broker. You can specify multiple values for this attribute to authorize multiple users. Include the asterisk (*) and question mark (?) characters in the values to authorize a group of users. Only Tivoli Enterprise Portal users who have an ID that matches the values of this attribute are authorized to issue Take Action commands handled by the WebSphere Message Broker Monitoring agent. Tivoli Enterprise Portal user IDs are defined in the Tivoli Enterprise Monitoring Server and do not necessarily exist on the node on which the agent is running. The values of this attribute apply only to commands that are not related to a particular broker, or that are related to a broker that does not have a specific list of users specified using the **takeActionAuthUsers** attribute. This attribute overrides the global **defaultTakeActionAuthUsers** attribute.

If this attribute is not specified, the **defaultTakeActionAuthUsers** value is used instead.

historicalAccountingType

For WebSphere Message Broker V7.0 or later, the **defaultHistoricalAccountingType** attribute tells the agent whether to collect historical archive accounting data. For WebSphere Message Broker V6 (all releases), this attribute tells the agent which of the two types of accounting data (snapshot and archive) to log historically. The value of this attribute has no effect unless historical data collection for any of the four accounting attribute groups has been enabled in Tivoli Enterprise Portal. This attribute overrides the global **defaultHistoricalAccountingType** attribute. For more information, see “defaultHistoricalAccountingType” on page 21.

retainRecentSnapshotSamples

The **retainRecentSnapshotSamples** attribute specifies the number of recent snapshot records to keep for any given message flow. It overrides the global **defaultRetainRecentSnapshotSamples** attribute. By default, this attribute is not specified in the configuration file; the value of **defaultRetainRecentSnapshotSamples** is used instead.

retainRecentArchiveSamples

The **retainRecentArchiveSamples** attribute determines the number of recent archive records to keep for any given message flow. It overrides the global **defaultRetainRecentArchiveSamples** attribute. By default this attribute is not specified in the configuration file, and the value of **defaultRetainRecentArchiveSamples** is used instead.

retainRecentPubSubSamples

The **retainRecentPubSubSamples** attribute specifies the minimum number of recent publish and subscribe data records stored by the agent. The higher this value, the

more data is available in the Publish-Subscribe Statistics workspace. This attribute overrides the global **defaultRetainRecentPubSubSamples** attribute.

By default, this attribute is not specified in the configuration file. The global **defaultRetainRecentPubSubSamples** attribute is used instead.

retainRecentResourceSamples

The **retainRecentResourceSamples** attribute determines whether to keep recent resource records for any given message flow. This attribute value can be either 1 or 0. The default resource interval for a broker is 60 minutes. So when this attribute is set to 1, one hour of resource statistics data can be displayed in the Resource Statistics workspaces. When this attribute is set to 0, the WebSphere Message Broker Monitoring agent does not subscribe the resource statistics events that are emitted by WebSphere Message Broker and the Resource Statistics workspaces are empty. You can set this attribute to 0 only for WebSphere Message Broker 7.0 or later.

By default this attribute is not specified in the configuration file, and the value of **defaultRetainRecentResourceSamples** is used instead.

retainMonitoringEvents

The **retainMonitoringEvents** attribute overrides the global **defaultRetainMonitoringEvents** attribute and determines how many monitoring events to retain per broker. Report information is available only for retained monitoring events. When used with situations, these events are never reset automatically and if historical situations are active, you can view them historically.

If this attribute is not specified, the **defaultRetainMonitoringEvents** value is used instead.

If the attribute is specified, the value must be greater than or equal to 0. You can set this attribute to 0 only for WebSphere Message Broker 7.0. When the **retainMonitoringEvents** value is 0, the WebSphere Message Broker Monitoring agent will not subscribe the monitoring events emitted by WebSphere Message Broker and the Monitoring and Audit Events workspace is empty.

collectNodeData

The **collectNodeData** attribute determines whether node definition data collection and parsing are performed. Possible values are YES to enable these options or NO to disable them (case insensitive). If this attribute is not specified for a broker, the default value specified by the global “defaultCollectNodeData” on page 24 attribute is used instead.

If the **collectNodeData** attribute is set to NO, the following workspaces contains no data:

- WebSphere Message Broker V7.0 or later:
 - “Processing Node Attributes workspace” on page 284
 - “Message Flow Node Topology workspace” on page 278
- WebSphere Message Broker V6 (all releases):
 - “Message Processing Node Information workspace” on page 303
 - “Message Processing Node Attributes workspace” on page 302
 - “Message Flow Topology workspace” on page 301

In addition, the following workspaces only contains partial data:

- WebSphere Message Broker V7.0 or later:
 - “Archive Accounting Node Statistics workspace” on page 272
 - “Snapshot Accounting Node Statistics workspace” on page 286
- WebSphere Message Broker V6 (all releases):
 - “Archive Node Accounting workspace” on page 291
 - “Snapshot Node Accounting workspace” on page 309

However, if message brokers in your environment have large message flows with a lot of message processing node data, enabling this option might significantly degrade system performance.

persistentBrokerData

The **persistentBrokerData** attribute specifies whether data related to brokers is stored persistently by writing it to a file. Possible values are YES and NO (case insensitive). When this option is enabled, the file used to store might be large, depending on the volume of broker information that needs to be stored.

If this attribute is not specified, the value of the **defaultPersistentBrokerData** attribute is used instead.

Remember:

- The agent must be restarted to make the change of this attribute value take effect. Before you start the agent again, delete the following items (if any) from your environment:
 - Subscriptions that are not in use
 - The reply queue of a broker
- If you change this attribute from YES to NO, also delete the existing persistent data file before restarting the agent.

Tip: The persistent data file is stored in the following directory:

- Windows: *install_dir*\TMAITM6\logs\History\KQI
- UNIX or Linux: *install_dir/arch/qi/hist*

where *install_dir* is the directory where IBM Tivoli Monitoring is installed. The default is C:\IBM\ITM on Windows systems or /opt/IBM/ITM on UNIX or Linux systems. *arch* is the architecture code of the operating system of this computer. For a list of architecture codes, see Appendix D, “Architecture codes,” on page 337.

- If you change this attribute from NO to YES, set the **replyQueueModel** attribute to the name of a permanent model queue. Make sure that the specified permanent model queue exists, such as SYSTEM.DURABLE.MODEL.QUEUE, which is automatically created by the broker.

refreshInterval

The **refreshInterval** attribute specifies the number of seconds interval between requesting information not reported as broker events from the broker. This data is not related to the deployment of broker objects. This includes trace and accounting/statistics settings for execution groups and message flows and broker subscription data. If you increase the length of the interval at which this data is collected, data in workspaces might be less up-to-date, but performance can be improved.

If this attribute is not specified, the value of **defaultRefreshInterval** is used instead.

WMBInstallDirectory

The **WMBInstallDirectory** attribute provides the installation directory of the broker that you want to monitor to the WebSphere Message Broker Monitoring agent. By default, the monitoring agent can automatically search for the installation directory and discover the broker. Specify this attribute only when the monitoring agent cannot discover the broker in a complex environment. If this attribute is not specified, the monitoring agent will use the **defaultWMBInstallDirectory** value to search for all brokers.

The **WMBInstallDirectory** attribute only applies to distributed systems. This attribute is not in the default `kqi.xml` file. You can add it in the `kqi.xml` file as needed.

WMQInstallDirectory

The **WMQInstallDirectory** attribute provides the installation directory of WebSphere MQ to the WebSphere Message Broker Monitoring agent. Some JAR files of WebSphere MQ are required by the CMP API to interact with the broker. By default, the monitoring agent can automatically find this installation directory. Specify this attribute only when the monitoring agent cannot find it in a complex environment. If this attribute is not specified, the **defaultWMQInstallDirectory** value is used.

The **WMQInstallDirectory** attribute only applies to distributed systems. This attribute is not in the default `kqi.xml` file. You can add it in the `kqi.xml` file as needed.

ConnectQueueManager

The `ConnectQueueManager` tag encapsulates parameters that affect queue managers. You can specify multiple `ConnectQueueManager` tags for different queue managers.

This tag is optional, but if you specify it, the **name** attribute is required. If this tag is not specified, agent-level defaults are used for the remaining attributes, and the agent automatically connects to the associated queue manager of the monitored broker. However, if the default reply and model queue names are not sufficient for a broker queue manager, you must specify all attributes of this tag:

name

The **name** attribute specifies the name of the queue manager that the agent connects to. This attribute is required.

replyQueueName

The **replyQueueName** attribute specifies the name of the queue that is used by the agent to receive publications and reply messages from the broker using this queue manager. If the named queue does not exist, a permanent dynamic queue is created automatically.

If this attribute is changed after starting a WebSphere Message Broker Monitoring agent, remove the subscription associated with the previous queue name and the previous queue. If a single broker is monitored by more than one agent, specify a different reply queue for each agent.

If this attribute is not specified, the **defaultReplyQueueName** value is used instead.

replyQueueModel

The **replyQueueModel** attribute specifies the name of the queue that is used as a model for creating the agent reply queue on this queue manager. The agent can create a queue with the name specified by the **replyQueueName** attribute for the queue manager using the properties of the queue specified by the **replyQueueModel** attribute as a model. Thus, by changing the properties of the queue named by this attribute, you can modify the properties of the agent reply queue used by the queue manager.

The model queue specified by this attribute must be a permanent dynamic model queue with persistent messages enabled when the **persistentBrokerData** attribute is YES, otherwise broker event messages received when the agent is stopped might be lost. In this case, as the broker uses these messages to detect changes in objects deployed in the broker environment while the agent is stopped and update workspace information accordingly. If you cannot use a permanent dynamic model queue with persistent messages enabled, use one of the following options:

- Set the **persistentBrokerData** attribute to NO. In this way, the agent can automatically detect all deployed objects at startup, ensuring that workspace information is always up-to-date. Setting the **persistentBrokerData** attribute to No increases processing overhead on the message broker, and in complex environments might cause long delays in starting the WebSphere Message Broker Monitoring agent.
- When you know that the broker environment has been updated, issue the **QI Refresh Broker Data** take action command to update workspace information. Issuing this take action command causes all broker information to be updated instead of only information related to updated components, and so might take a long time.

If this attribute is not specified, the **defaultReplyQueueModel** value is used instead.

Examples of agent configuration

This section presents several examples to explain how you can customize the configuration for better performance of the WebSphere Message Broker Monitoring agent.

Monitoring a large number of brokers

To minimize performance overhead, run only one WebSphere Message Broker Monitoring agent on each host system. However, if the performance of the agent becomes adversely affected by monitoring a large number of brokers simultaneously, you can split monitoring between several agents.

For most distributed systems, one agent per host system must suffice, because the agent is designed to monitor all brokers on such a system, or to monitor a subset of brokers, as specified in the agent parameter file.

For systems with a large number of brokers, such as z/OS systems, you might need to adjust the agent parameter that limits the number of brokers that are monitored (see “maximumAgentCollectionThreads” on page 27) and verify that monitoring the large number of brokers is not reducing performance. If performance is affected, you can divide monitoring among several agents.

If CPU usage is high or response times for displaying Tivoli Enterprise Portal workspace reports or situation monitoring actions are unacceptably long,

performance might be adversely affected by monitoring too many brokers. You might need to experiment to determine the best number of agents for your enterprise.

Exception: On AIX systems, because of shared memory limitations, do not attempt to monitor more than 10 brokers with a single agent on an AIX host.

Dividing broker monitoring between different agents on Windows systems

To dividing broker monitoring between different agents, do the following steps:

1. Optional: Create the secondary instance of the WebSphere Message Broker Monitoring agent, if you have not done so. For more information about how to create the agent instance, see “Windows systems: Creating multiple instances of the monitoring agent” on page 17.
2. In the Manage Tivoli Enterprise Monitoring Services window, right-click the agent instance that you want to configure, and then click **Reconfigure**.
3. In the Agent Advanced Configuration window, click **OK**, until a message is displayed asking if you want to update the configuration file of the agent instance before configuring the WebSphere Message Broker Monitoring agent. Click **Yes**.
4. The following message is displayed. Click **OK**.

Configuration will wait for you to close your default text editor before continuing.

The configuration file is opened in your default text editor.

5. Edit the configuration file as follows:
 - a. Specify the value of the **agentId** parameter to run more than one instance on a single Windows system. Ensure that the value of the **agentId** parameter is different from that of all the other agent instances running on the same system.
 - b. Use the <MonitorBroker> tag to specify which brokers this agent monitors.

Important: One broker can be monitored by only one agent. If there are two agents monitoring the same broker, the data displayed in workspaces is wrong. For more information about the related parameters, see “agentId” on page 19 and “MonitorBroker” on page 28.

6. Save and close the file.
7. A message is displayed stating that the configuration file edit session is complete. Click **Yes** to configure the agent.
8. Repeat step 2 to step 7, until you finish configuring all the agents and dividing monitoring for all the brokers in your environment.

Dividing broker monitoring between different agents on UNIX and Linux systems

Do the following steps to configure each agent that you want to use to monitor a set of brokers. For each agent, you must specify one broker to be the lead broker, which is used to send commands to the other brokers when the **itmcmd agent stop** and **itmcmd agent start** commands are issued.

1. Optional: Create the secondary instance of the WebSphere Message Broker Monitoring agent using the **itmcmd agent start** command, if you have not done so.

Example:

```
./itmcmd agent -o AGT1 -p WBIBRK1 start qi
```

The *hostname_qi_WBIBRK1_##_AGT1.xml* file is created automatically in this example, where *hostname* is the host name of the Tivoli Enterprise Monitoring Server. For more details about how to create a secondary instance, see “UNIX and Linux systems: Creating multiple instances of the monitoring agent” on page 18.

2. Edit the XML file for each agent as follows:
 - a. Specify the value of the **agentId** parameter to run more than one instance on a single system. Ensure that the value of the **agentId** parameter is different from that of all the other agent instances running on the same system.
 - b. Use the <MonitorBroker> tag to specify which brokers this agent monitors.

Important: One broker can be monitored by only one agent. If there are two agents monitoring the same broker, the data displayed in workspaces is wrong. For more information about the related parameters, see “agentId” on page 19 and “MonitorBroker” on page 28.

In this example (for the *unixhost01_qi_WBIBRK1_##_AGT1.xml* file), the *WBIBRK2* and *WBIBRK3* brokers are added:

```
<KqiAgent version="701"
  agentId="AGT1"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="300"
  defaultStatisticInterval="60"
  defaultFlowEventInterval="15"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  holdTimeForQuery="180"
  defaultReplyQueueName="KQI.AGENT.REPLY.QUEUE"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="YES"
  maximumMessageLength="10240"
  defaultPersistentBrokerData="NO"
  defaultRefreshInterval="300"
  defaultTakeActionAuthUsers="*">
  <MonitorBroker name="WBIBRK1">
  </MonitorBroker>
  <MonitorBroker name="WBIBRK2">
  </MonitorBroker>
  <MonitorBroker name="WBIBRK3">
  </MonitorBroker>
</KqiAgent>
```

3. Save and close the file.
4. Stop and restart the agent, using the **itmcmd agent stop** and **itmcmd agent start** commands with the **-o** option.

Example:

```
itmcmd agent -o AGT1 stop qi
itmcmd agent -o AGT1 start qi
```

5. Repeat step 2 to step 4 until you finish configuring all the agent instances and dividing monitoring for all the brokers in your environment.

Disabling broker data collection

In some extreme situations where a broker environment contains a large quantity of data, you might want to disable the collection of broker data.

You can do this by setting the **collectNodeData** parameter of the WebSphere Message Broker Monitoring agent (see “collectNodeData” on page 31 for further information).

When the broker data collection is disabled, there is no data in the following workspaces:

- WebSphere Message Broker V7.0 or later:
 - “Message Flow Node Topology workspace” on page 278
 - “Processing Node Attributes workspace” on page 284
- WebSphere Message Broker V6 (all releases):
 - “Message Processing Node Information workspace” on page 303
 - “Message Processing Node Attributes workspace” on page 302
 - “Message Flow Topology workspace” on page 301

In addition, there is only partial data in the following workspaces:

- WebSphere Message Broker V7.0 or later:
 - “Archive Accounting Node Statistics workspace” on page 272
 - “Snapshot Accounting Node Statistics workspace” on page 286
- WebSphere Message Broker V6 (all releases):
 - “Archive Node Accounting workspace” on page 291
 - “Snapshot Node Accounting workspace” on page 309

Remember: When snapshot accounting is enabled, the queue can become fast when the agent is not running.

Enabling persistent broker data collection

To enable persistent broker data collection for all brokers in your environment, do the following steps:

1. Navigate to the installation directory of the WebSphere Message Broker Monitoring agent and open the `kqi.xml` configuration file in a standard text editor.
2. Set the **defaultPersistentBrokerData** parameter to YES.
3. Set the **defaultReplyQueueModel** parameter to the name of a permanent model queue. Make sure that the specified permanent model queue exists, such as `SYSTEM.DURABLE.MODEL.QUEUE`, which is automatically created by the broker.
4. Save and close the file.
5. Recycle the WebSphere Message Broker Monitoring agent.

This procedure enables persistent broker data for all brokers in your environment that do not have the **persistentBrokerData** parameter set in the `kqi.xml` file. To disable persistent broker data, set the **defaultPersistentBrokerData** parameter to NO.

To enable persistent broker data storage for a single message broker, do the following steps:

1. Navigate to the installation directory of the WebSphere Message Broker Monitoring agent and open the `kqi.xml` configuration file in a standard text editor.
2. Locate the `MonitorBroker` tag for the message broker that you want to enable the persistent broker data collection for.

Tip: If there is no `MonitorBroker` tag in the configuration file, add it to the file. See “`MonitorBroker`” on page 28 for detailed information about this tag.

3. Set the **`persistentBrokerData`** parameter to YES.
4. Set the **`replyQueueModel`** parameter to the name of a permanent model queue. Make sure that the specified permanent model queue exists, such as `SYSTEM.DURABLE.MODEL.QUEUE`, which is automatically created by the broker.
5. Save and close the file.
6. Recycle the WebSphere Message Broker Monitoring agent.

To disable persistent broker data collection, set the **`persistentBrokerData`** parameter to NO.

If persistent broker data collection is enabled and the agent is not running, the queue in which event messages are stored gradually becomes full. If the queue becomes full, event messages might be lost, causing inaccuracies in the information displayed in Tivoli Enterprise Portal when the agent is started again.

Remember: If the WebSphere Message Broker Monitoring agent stops abnormally when persistent broker data collection is enabled, error messages might be included in the message broker log when the agent is started again. To avoid this problem, do the following tasks before starting the agent again:

- Using the Message Broker Toolkit, delete the `$SYS/Broker/<broker_label>/#` subscription.
- Delete the reply queue that is used by the WebSphere Message Broker Monitoring agent to store persistent data.

Chapter 3. Monitoring with the CandleMonitor node

The CandleMonitor node is an optional component of the WebSphere Message Broker Monitoring agent. This node collects message flow performance statistics for a broker and provides a mechanism for generating user-defined events within a message flow. These events can be used by situations to detect abnormalities in the message flow. For example, statistics that are generated by a CandleMonitor node can be used to create a situation that automatically stops a message flow that has messages flowing along an abnormal path.

This section provides instructions for installing, enabling, and positioning the CandleMonitor node to monitor message flow statistics and events, turning monitoring off and on, and customizing the CandleMonitor node.

The CandleMonitor node is a simple passthrough node with one input terminal and one output terminal, which is placed in a message flow to collect statistics (see Figure 6). Messages passing through the CandleMonitor node are propagated between its input and output terminals without change. The CandleMonitor node is implemented in the `kqipnode.lil` (or `kqipnode64.lil`) file.

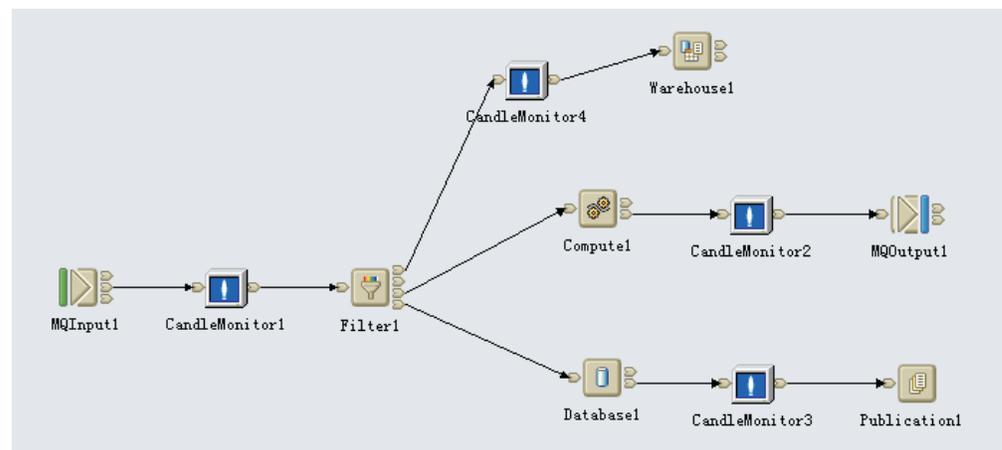


Figure 6. Example of a monitored flow

You can use the CandleMonitor node to take the following actions:

- View message flow reports at broker, execution group, message flow, and node levels
- Detect high input and output rates in message flows
- Detect high average time spent in message flows
- Detect high queue times for input to message flows
- Determine which part of a message flow is taking more time
- View statistics from the last sampling interval and average values taken over a particular period of time
- Be alerted to non-normal or low-use paths
- Detect events and trigger automatic responses

By default, statistics are sampled once per minute and events are sampled once every 15 seconds.

Remember: With broker V5 or later versions, accounting statistics can be made available in the Tivoli Enterprise Portal without using a CandleMonitor node. Some of the accounting statistics are similar to those provided by the CandleMonitor node. However, the CandleMonitor node can also be used to monitor subflows and define message flow events. Message flows without a CandleMonitor node are not represented in the Statistics workspaces.

Prerequisites

Before you can use the CandleMonitor node to monitor message flows, perform the following tasks:

1. Make the CandleMonitor node available in the broker environment.
 - For broker environment on distributed systems, see “Making the CandleMonitor node available in broker environments.”
 - For broker environment on z/OS systems, see *IBM Tivoli OMEGAMON XE for Messaging on z/OS: Planning and Configuration Guide, SC23-7951, SC32-1830*.
2. Make the CandleMonitor node available in the Message Brokers Toolkit, as described in “Making the CandleMonitor node available in the Message Brokers Toolkit” on page 44.

Exception: If the broker to which the CandleMonitor node is deployed runs on AIX systems, enable the shared memory on AIX systems before the node is deployed to the broker. Because the WebSphere Message Broker is a 32-bit application that is memory intensive, the broker cannot attach to more than 11 shared memory segments. When it runs short of process address space, the broker cannot attach to the queue manager-related shared memory resources.

To enable the shared memory on AIX systems, do the following steps:

1. Stop the broker by running the following command, where *broker* is the name of your broker:

```
mqsistop broker
```
2. Run the following command to ensure that the broker is running in an environment with the extended memory variable exported:

```
export EXTSHM=ON
```
3. Restart the broker by running the following command, where *broker* is the name of your broker:

```
mqsistart broker
```
4. On the DB2[®] server, ensure that the shared memory support is enabled.

Making the CandleMonitor node available in broker environments

This section describes how to make the CandleMonitor node available in your broker environment on distributed systems.

WebSphere Message Broker software includes message flow accounting and statistics that partially overlap with the statistics provided by the CandleMonitor node and can be monitored in Tivoli Enterprise Portal without including the CandleMonitor node in message flows. However, the CandleMonitor node provides additional statistics and the ability to monitor subflows and define message flow events. Message flows without a CandleMonitor node are not represented in Statistics workspaces.

Important: You must use a user ID with root authority when installing the CandleMonitor node.

Follow the corresponding instructions to install the CandleMonitor node in your broker environment, according to the version of broker that you use:

- 32-bit brokers on Windows systems: “Making the CandleMonitor node available in 32-bit broker environments on Windows systems”
- 64-bit brokers on Windows systems: “Making the CandleMonitor node available in 64-bit broker environments on Windows systems”
- brokers on UNIX or Linux systems: “Making the CandleMonitor node available in broker environments on UNIX and Linux systems” on page 42

Making the CandleMonitor node available in 32-bit broker environments on Windows systems

If the broker is 32-bit on a Windows system, do the following steps to install the CandleMonitor node in the broker environment:

1. Verify that the `kqipnode.lil` file was copied to the `bin` directory of the broker during the installation of WebSphere Message Broker Monitoring agent. If the file was not copied to the `bin` directory of the broker during installation, do it now by performing the following steps:
 - a. Stop the broker.
 - b. Copy the `kqipnode.lil` file from the `install_dir\TMAITM6` directory to the `bin` directory of the broker.

Tip: This is an example of a typical `bin` location: `C:\Program Files\IBM\MQSI\7.0\bin`.

2. Verify that the `kqipnode.lil` file in the `bin` directory of the broker *exactly* matches (in size, date, and the time last modified) the one that is in the `install_dir\TMAITM6` directory.

Attention: A mismatch of the release-level of these files can cause a failure of the broker statistics and message flow events reporting.

3. Start the broker again.
4. Repeat step 1 to step 3 for each Windows system with a broker to be monitored.

The CandleMonitor node is now installed in your broker environment.

Remember: Before you can use the CandleMonitor node to monitor message flows, it must also be made available in the Message Brokers Toolkit. For instructions, see “Making the CandleMonitor node available in the Message Brokers Toolkit” on page 44.

Making the CandleMonitor node available in 64-bit broker environments on Windows systems

If the broker is 64-bit on a Windows system, do the following steps to install the CandleMonitor node in the broker environment:

1. Verify that the `kqipnode64.lil`, `kbb.dll`, and `k1x.dll` files were copied to the `bin` directory of the broker during the installation of WebSphere Message Broker Monitoring agent. If the files were not copied to the `bin` directory of the broker during installation, do it now by performing the following steps:
 - a. Stop the broker.

- b. Copy the three files from the *install_dir*\TMAITM6\kqipnode64 directory to the bin directory of the broker, where *install_dir* is the installation directory of the WebSphere Message Broker Monitoring agent.

Tip: This is an example of a typical bin location: C:\Program Files\IBM\MQSI\7.0\bin.

2. Verify that the kqipnode64.lil file in the bin directory of the broker *exactly* matches (in size, date, and the time last modified) the one that is in the *install_dir*\TMAITM6\kqipnode64 directory.

Important: A mismatch of the release-level of these files can cause a failure of the broker statistics and message flow events reporting.

3. Check if Microsoft .NET Framework 3.5 is installed on your system. If not, install it now.
 - a. Log on the Windows system.
 - b. Click **Start > Administrative Tools**.
 - c. Click **Server Manager**.
 - d. In the Server Manager window, click **Features** on the left navigation pane.
 - e. On the right side of the window, click **Add Features**.
 - f. In the Select Features window, select **.NET Framework 3.5 Features** and click **Install**.
4. Start the broker again.
5. Repeat step 1 to step 4 for each Windows system with a broker to be monitored.

The CandleMonitor node is now installed in your broker environment.

Remember: Before you can use the CandleMonitor node to monitor message flows, it must also be made available in the Message Brokers Toolkit. For instructions, see “Making the CandleMonitor node available in the Message Brokers Toolkit” on page 44.

Making the CandleMonitor node available in broker environments on UNIX and Linux systems

To make the CandleMonitor node available in a broker environment running on a UNIX or Linux system, do the following steps:

1. Navigate to the /usr/bin directory.
2. Run the following commands to create a set of soft links:

```
(for 32-bit execution groups)ln -sf install_dir/arch/qi/bin/kqipnode.lil
broker_dir/lil32_dir/kqipnode.lil
(for 64-bit execution groups)ln -sf install_dir/arch/qi/bin/kqipnode.lil
broker_dir/lil64_dir/kqipnode.lil
ln -sf install_dir/arch/qi/bin/kqipnode.cfg
broker_dir/lil/kqipnode.cfg
ln -sf install_dir/arch/qi/bin/langcode/CandleMonitorNode.cat
broker_dir/messages/langcode/CandleMonitorNode.cat
```

where:

- *arch* is the architecture code of the operating system, see Appendix D, “Architecture codes,” on page 337 for a list of architecture codes.
- *broker_dir* is the full path of the broker installation directory.
- *install_dir* is the full path of the IBM Tivoli Monitoring installation directory.
- *langcode* is the code for the installed language pack, see Appendix C, “Language codes,” on page 335 for a list of language codes.

- *lil32_dir* is the message broker directory for 32-bit plugin node. Its value for WebSphere Message Broker 6.0, 7.0, and 8.0 is *lil*. Its value for WebSphere Message Broker 6.1 is *lil32*.
 - *lil64_dir* is the message broker directory for 64-bit plugin node. Its value for WebSphere Message Broker 6.1, 7.0, and 8.0 is *lil*. Its value for WebSphere Message Broker 6.0 is *lil64*.
3. Create another link by running one of the following commands depending on which operating system you are using:
 - On AIX systems:


```
ln -sf install_dir/arch/qi/bin/langcode/CandleMonitorNode.cat
        /usr/lib/nls/msg/langcode/CandleMonitorNode.cat
```
 - On Solaris systems:


```
ln -sf install_dir/arch/qi/bin/langcode/CandleMonitorNode.cat
        /usr/lib/locale/C/LC_MESSAGES/CandleMonitorNode.cat
```
 - On HP-11 systems:


```
ln -sf install_dir/arch/qi/bin/langcode/CandleMonitorNode.cat
        /usr/lib/nls/msg/C/CandleMonitorNode.cat
```
 - On Intel-based Linux systems:


```
ln -sf install_dir/arch/qi/bin/langcode/CandleMonitorNode.cat
        /usr/share/locale/C/LC_MESSAGES/CandleMonitorNode.cat
```
 - On Linux for zSeries systems:


```
ln -sf install_dir/arch/qi/bin/langcode/CandleMonitorNode.cat
        /usr/share/locale/C/LC_MESSAGES/CandleMonitorNode.cat
```
 4. Optional: To take full advantage of 64-bit architectures, if you are running the CandleMonitor node on a 64-bit AIX, Solaris, or HP-UX (PA-RISC, not Itanium) systems, you must also create links to an additional 64-bit version of the CandleMonitor node file. To do this, repeat step 2, replacing the *kqipnode.lil* file name with *kqipnode64.lil*.
 5. If you are running Linux on zSeries systems, do the following steps to define the required environment variable:
 - a. Open the profile of the user ID that is used to start the broker. By default, the *mqsiprofile* file is located in the */opt/ibm/mqsi/*nn*/bin/* directory, where *nn* indicates the version of your WebSphere Message Broker. If you are using a profile other than the default one, open that profile.
 - b. Append the following lines to the file and save it:


```
MQSI_PRELOAD=/opt/gcc295/lib/libstdc++-libc6.2-2.so.3
export MQSI_PRELOAD
```
 - c. Run the profile of the user ID that is used to start the broker.
 - d. Run the DB2 profile by executing the following command:


```
DB2_Instance_Home/sql1lib/db2profile
```

where *DB2_Instance_Home* is the home directory of the DB2 instance.
 6. Start the broker again.

Remember: After the CandleMonitor node is installed, it must be made available in the Message Brokers Toolkit. For instructions, see “Making the CandleMonitor node available in the Message Brokers Toolkit” on page 44.

Making the CandleMonitor node available in the Message Brokers Toolkit

Before you can insert the CandleMonitor node into message flows, you must make the node available in the Message Brokers Toolkit.

The CandleMonitor node is stored in a file that is created when you install the WebSphere Message Broker Monitoring agent. Several versions of this file are created during installation of the monitoring agent, for use with different versions of the toolkit. The different versions are as follows:

- WebSphere Message Broker V7.0 or later:
 - Windows: Use the `kqicm700.exe` CandleMonitor node file.

Default location: `C:\IBM\ITM\TMAITM6`

- Linux: Use the `kqicm700.tar` CandleMonitor node file.

Default location: `/opt/IBM/ITM/arch_code/qi/bin`

- WebSphere Message Broker V6 (all releases):
 - Windows: Use the `kqicm600.exe` CandleMonitor node file.

Default location:

- Message Brokers Toolkit V6.0: `C:\IBM\ITM\TMAITM6`
- Message Brokers Toolkit V6.1: `C:\Program Files\IBM\WMBT610\plugins`
- Linux: Use the `kqicm600.tar` CandleMonitor node file.

Default location:

- Message Brokers Toolkit V6.0: `/opt/IBM/ITM/arch_code/qi/bin`
- Message Brokers Toolkit V6.1: `/opt/IBM/WMBT610/plugins`

Making the CandleMonitor node available in Message Brokers Toolkit V6.0

To make the CandleMonitor node available in Message Brokers Toolkit V6.0, do the following steps:

1. Exit the Message Brokers Toolkit if it is open.
2. If the WebSphere Message Broker Monitoring agent and the Message Brokers Toolkit are running on the same system, depending on the operating system on which the WebSphere Message Broker Monitoring agent is installed, perform one of the following procedures to make the CandleMonitor node available in the Message Brokers Toolkit:

- If the WebSphere Message Broker Monitoring agent is installed on Windows systems, go to the directory where the `kqicm600.exe` file is located (by default, it is located in the `C:\IBM\ITM\TMAITM6` directory) and run the following command:

```
kqicm600.exe -d "WMBT_installdir\evtoolkit\eclipse\plugins"
```

where `WMBT_installdir` is the Message Brokers Toolkit installation directory.

- If the WebSphere Message Broker Monitoring agent is installed on a Linux or UNIX system, go to the directory where the `kqicm600.tar` file is located (by default, it is located in the `/opt/IBM/ITM/arch/qi/bin`, where `arch` specifies the architecture code of the operating system of this computer. See Appendix D, "Architecture codes," on page 337 for a list of architecture codes) and run the following commands:

```
cp kqicm600.tar WMBT_installdir/evtoolkit/eclipse/plugins
cd WMBT_installdir/evtoolkit/eclipse/plugins
tar -xf kqicm600.tar
```

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

The following two subdirectories are created:

- `com.candle.monitor_6.0.0` (contains all the necessary plug-in support files)
 - `com.candle.monitor.n11_6.0.0` (contains all the native language support files)
3. If the WebSphere Message Broker Monitoring agent and the Message Brokers Toolkit are running on different systems, depending on the operating system on which the WebSphere Message Broker Monitoring agent is installed, perform one of the following procedures to make the CandleMonitor node available in Message Brokers Toolkit:

- If the WebSphere Message Broker Monitoring agent is installed on a Windows system, do the following steps:
 - a. Create a temporary directory on the system where the WebSphere Message Broker Monitoring agent is installed.
 - b. Go to the directory where the `kqicm600.exe` file is located on the system where the WebSphere Message Broker Monitoring agent is installed (by default, it is located in the `C:\IBM\ITM\TMAITM6` directory) and run the following command:

```
kqicm600.exe -d "temp_dir"
```

where *temp_dir* is the directory that you created in the previous step.

The following two subdirectories are created in the *temp_dir* directory:

- `com.candle.monitor_6.0.0` (contains all the necessary plug-in support files)
 - `com.candle.monitor.n11_6.0.0` (contains all the native language support files)
- c. Copy the two directories that were created in the previous step and their contents to the following directory on the computer where the Message Brokers Toolkit is installed:
- If the Message Brokers Toolkit is running on Windows systems, the destination directory is *WMBT_installdir*\evtoolkit\eclipse\plugins.
 - If the Message Brokers Toolkit is running on a UNIX or Linux system, the destination directory is *WMBT_installdir*/evtoolkit/eclipse/plugins.

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

- If the WebSphere Message Broker Monitoring agent is installed on a Linux or UNIX system, perform the following steps:
 - a. Go to the directory where the `kqicm600.tar` file is located on the system where the WebSphere Message Broker Monitoring agent is installed (by default, it is located in the `/IBM/ITM/arch/qi/bin`, where *arch* specifies the architecture code of the operating system of this computer; see Appendix D, “Architecture codes,” on page 337 for a list of architecture codes) and run the following command:

```
tar -xf kqicm600.tar
```

The following two subdirectories are created in the current directory:

- com.candle.monitor_6.0.0 (contains all the necessary plug-in support files)
 - com.candle.monitor.n11_6.0.0 (contains all the native language support files)
- b. Copy the two subdirectories that were created in the previous step and their contents to the following directory on the system where the Message Brokers Toolkit is installed:
- If the Message Brokers Toolkit is installed on Windows systems, the destination directory is *WMBT_installdir\evtoolkit\eclipse\plugins*.
 - If the Message Brokers Toolkit is installed on a UNIX or Linux system, the destination directory is *WMBT_installdir/evtoolkit/eclipse/plugins*.

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

4. Start the Message Brokers Toolkit again using the `-clean` option.

Now you can see the CandleMonitor node in the message flow editor palette under the IBM category in the Broker Application Development Perspective.

Remember: When a new user-defined extension plug-in is added to the `\plugins` directory, or when the `plugin.xml` file is changed, you must start the Toolkit using the `-clean` option again. To do this, specify the `-clean` option when you start the Toolkit from command line.

Making the CandleMonitor node available in Message Brokers Toolkit V6.1

To make the CandleMonitor node available in Message Brokers Toolkit V6.1, do the following procedure:

1. Exit the Message Brokers Toolkit if it is open.
2. If the WebSphere Message Broker Monitoring agent and the Message Brokers Toolkit are running on the same system, depending on the operating system on which the WebSphere Message Broker Monitoring agent is installed, perform one of the following procedures to make the CandleMonitor node available in Message Brokers Toolkit:

- If the WebSphere Message Broker Monitoring agent is installed on Windows systems, go to the directory where the `kqicm600.exe` file is located and run the following command:

```
kqicm600.exe -d "WMBT_installdir\plugins"
```

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

- If the WebSphere Message Broker Monitoring agent is installed on a Linux or UNIX system, go to the directory where the `kqicm600.tar` file is located and run the following commands:

```
cp kqicm600.tar WMBT_installdir/plugins
cd WMBT_installdir/plugins
tar -xf kqicm600.tar
```

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

The following two subdirectories are created:

- com.candle.monitor_6.0.0 (contains all the necessary plug-in support files)
- com.candle.monitor.n11_6.0.0 (contains all the native language support files)

3. If the WebSphere Message Broker Monitoring agent and the Message Brokers Toolkit are running on different systems, depending on the operating system on which the WebSphere Message Broker Monitoring agent is installed, perform one of the following procedures to make the CandleMonitor node available in Message Brokers Toolkit:

- If the WebSphere Message Broker Monitoring agent is installed on a Windows system, perform the following steps:
 - a. Create a temporary directory on the system where the WebSphere Message Broker Monitoring agent is installed.
 - b. Go to the directory where the `kqicm600.exe` file is located on the system where the WebSphere Message Broker Monitoring agent is installed and run the following command:

```
kqicm600.exe -d "temp_dir"
```

where *temp_dir* is the directory that you created in the previous step.

The following two subdirectories are created in *temp_dir*:

- `com.candle.monitor_6.0.0` (contains all the necessary plug-in support files)
 - `com.candle.monitor.n11_6.0.0` (contains all the native language support files)
- c. Copy the two directories that were created in the previous step and their contents to the following directory on the system where the Message Brokers Toolkit is installed:
 - If the Message Brokers Toolkit is installed on Windows systems, the destination directory is *WMBT_installdir*\plugins.
 - If the Message Brokers Toolkit is installed on a Linux or UNIX system, the destination directory is *WMBT_installdir*/plugins.

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

- If the WebSphere Message Broker Monitoring agent is installed on a Linux or UNIX system, perform the following steps:
 - a. Go to the directory where the `kqicm600.tar` file is located on the system where the WebSphere Message Broker Monitoring agent is installed and run the following command:

```
tar -xf kqicm600.tar
```

The following two subdirectories are created in the current directory:

- `com.candle.monitor_6.0.0` (contains all the necessary plug-in support files)
 - `com.candle.monitor.n11_6.0.0` (contains all the native language support files)
- b. Copy the two directories that were created in the previous step and their contents to the following directory on the system where the Message Brokers Toolkit is installed:
 - If the Message Brokers Toolkit is installed on Windows systems, the destination directory is *WMBT_installdir*\plugins.
 - If the Message Brokers Toolkit is installed on a UNIX or Linux system, the destination directory is *WMBT_installdir*/plugins.

where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

4. Restart the Message Brokers Toolkit using the `-clean` option.

Now you can see the CandleMonitor node in the message flow editor palette under the IBM category in the Broker Application Development perspective.

Remember: When a new user-defined extension plug-in is added to the `\plugins` directory, or when the `plugin.xml` file is changed, you must restart the Toolkit using the `-clean` option. To do this, specify the `-clean` option when you start the Toolkit from command line.

Making the CandleMonitor node available in Message Brokers Toolkit V7.0 or later

To make the CandleMonitor node available in Message Brokers Toolkit V7.0 or later, do the following procedure:

1. Exit the Message Broker Toolkit if it is open.
2. Find the CandleMonitor node file. The file name and location vary depending on the operating system where the WebSphere Message Broker Monitoring agent is running:

- Windows systems: `install_dir\TMAITM6\kqicm700.exe`
- UNIX or Linux systems: `install_dir/arch/qi/bin/kqicm700.tar`

where `install_dir` is the installation directory of the WebSphere Message Broker Monitoring agent; `arch` specifies the architecture code of the operating system of this computer. For a list of architecture codes, see Appendix D, “Architecture codes,” on page 337.

3. Extract the CandleMonitor node file. Two subdirectories are included in this file:

- `com.candle.monitor_7.0.0` (contains all the necessary plugin support files)
- `com.candle.monitor.nl1_7.0.0` (contains all the native language support files)

Tip: The version number in the subdirectory name, such as 7.0.0, indicates the latest version of WebSphere Message Broker.

4. Copy the two subdirectories (`com.candle.monitor` and `com.candle.monitor.nl1`) from the toolkit workspace folder to the following folder depending on the operating system where the toolkit is installed:

- Windows systems: `WMBT_installdir\plugins`
- UNIX or Linux systems: `WMBT_installdir/plugins`

where `WMBT_installdir` is the installation directory of WebSphere Message Broker Toolkit.

5. Restart the Message Broker Toolkit.

Now you can see the CandleMonitor node in the message flow editor palette under the IBM category in the Broker Application Development perspective.

Placing the CandleMonitor node in message flows

The primary use of the CandleMonitor node is to produce the statistics that are displayed in the high-level statistics workspaces in Tivoli Enterprise Portal. To ensure that the correct statistics are gathered for the following workspaces, the CandleMonitor node must be correctly placed in message flows:

- WebSphere Message Broker V7.0 or later:
 - “Monitor Node Broker Statistics workspace” on page 280
 - “Monitor Node Base Statistics workspace” on page 279
 - “Monitor Node Execution Group Statistics workspace” on page 281

- “Monitor Node Events workspace” on page 280
- “Monitor Node Message Flow Statistics workspace” on page 282
- WebSphere Message Broker V6 (all releases):
 - “Broker Statistics workspace” on page 295
 - “CandleMonitor Node Statistics workspace” on page 297
 - “Execution Group Statistics workspace” on page 299
 - “Message Flow Events workspace” on page 300
 - “Message Flow Statistics workspace” on page 301

For information about how to place the CandleMonitor node for monitoring different aspects of the message flow, see the following instructions:

- “Monitoring the input or output of a message flow”
- “Monitoring subflows” on page 50
- “Monitoring other aspects of a message flow” on page 52

A secondary use of the CandleMonitor node is to produce user-defined message flow events that can be detected by Message Flow Event situations and viewed in the Monitor Node Events workspace/Message Flow Events workspace. For information about how to use the CandleMonitor node for this purpose, see “Producing event messages” on page 52.

Important: You must provide unique label names to CandleMonitor nodes in a message flow. The names must be unique with respect to any other names in the entire message flow, including its subflows. If the CandleMonitor node label names are not unique, data for different nodes is combined in illogical ways. Do not use the default names that are assigned to your CandleMonitor nodes.

Monitoring the input or output of a message flow

When you are designing a message flow that is to be monitored by a broker, you must place a CandleMonitor node immediately after the MQInput node, so that the CandleMonitor node can gather the information that is required for input rate calculations. Another important location for placing the CandleMonitor node is immediately in front of any MQOutput, MQReply, or Publication nodes, so that the CandleMonitor node can gather the information that is required to calculate output rate calculations.

Tip: If you want to place only a single CandleMonitor node in a message flow, the input position is the best choice, because most statistics are generated from this position.

To monitor the input of a message flow, perform the following procedure:

1. Place a CandleMonitor node immediately after the MQInput node.
2. Set the **type** attribute of the CandleMonitor node placed in this position to input.
3. Optional: If the input queue to a message flow contains messages that have put date and times that do not accurately reflect when the message was put into the input queue, and you are using a CandleMonitor node with the **type** attribute set to input, set the **collectQueueTime** attribute to no, so that queue times are not calculated for these messages. When queue time statistics are determined, queue times are calculated using the put date and time of the message in the queue. Put date and times are not accurate when origin context is preserved for a message during the put operation by the application

performing this operation. Inaccuracies of put date and time commonly occur when an application is a message mover that transfers messages from one queue to another, or when an application passes or sets origin context for a message. If the input queue to *Message Flow B* is the output queue of *Message Flow A*, the broker passes the origin context so that the put date and time for the message in *Message Flow B* is not an accurate calculation of queue time. The **collectQueueTime** attribute for a CandleMonitor node placed in *Message Flow B* must be set to no.

To monitor the output of a message flow, do the following steps:

1. Place the CandleMonitor node immediately in front of any MQOutput, MQReply, or Publication nodes.
2. Set the **type** attribute of a CandleMonitor node placed in this position to output.

Monitoring subflows

A subflow is a section of a message flow that includes one or more message processing nodes. In general terms, a subflow can be any section of a message flow that can be separately identified. You can consider the message flow as being like a main routine and a subflow as being a subroutine. You can explicitly delineate the subroutine in the Message Brokers Toolkit by making a separate message flow, which is then embedded in the main flow (referred to here as *Type I*). Or, the message flow can have sections of nodes that you want to monitor as subflows, even though they are not explicitly delineated into a separate flow (referred to in this example as *Type II*).

When a message flow is deployed to a broker, the broker regards the entire message flow as a single entity. There is no obvious delineation in the broker for dividing the flow into separate subflows. The name assigned to a Type I subflow is not known to the broker (this entity is displayed only in the Message Brokers Toolkit and configuration manager). Any given message processing node is not aware of the other message processing nodes around it. Therefore, for the CandleMonitor node to be useful in monitoring both types of subflows, you must provide the required information by customizing the node.

To gather correct subflow statistics for either type of subflow, the subFlowOutput CandleMonitor node is required in the subflow; it is not optional, as is the case with the output node for a main message flow.

CandleMonitor node Statistics (the lowest level, most detailed report) combines data for all instances of the same node that are part of a subflow that has been embedded multiple times in the same message flow.

You can assign the **subFlowName** attribute to an input or output CandleMonitor node. When you use input and output node types in combination you do not need to insert two CandleMonitor nodes in a flow at the same position when a subflow comes either at a beginning or end of a message flow. The combination output node is probably used more, because a Type I subflow might have an output that is the destination for a message going through the whole message flow as well as through the subflow. (For a description of the effects of assigning a **subFlowName** attribute to a node type other than subFlowInput or subFlowOutput, see “subFlowName” on page 259.)

Monitoring a Type I subflow

To monitor a Type I subflow, do the following procedure:

1. Place a CandleMonitor node immediately after the Input Terminal in the flow, and place another CandleMonitor node immediately in front of the Output Terminal or other output of the flow (see Figure 7).
2. Assign the same value (subFlow1 in Figure 7) to the **subFlowName** attribute of all nodes in the subflow.

Tip: For Type I message flows, you can set the **subFlowName** attribute to the name that was specified in the Message Brokers Toolkit for the message flow that represents the subflow.

3. Set the **type** attribute of the CandleMonitor node placed in the input position to subFlowInput.
4. Set the **type** attribute of the CandleMonitor node placed in the output position to subFlowOutput.

Exception: If the output is a node such as MQOutput and represents the end of the message flow for a message going down a particular path, set the **type** attribute of the CandleMonitor node placed in the output position to output.

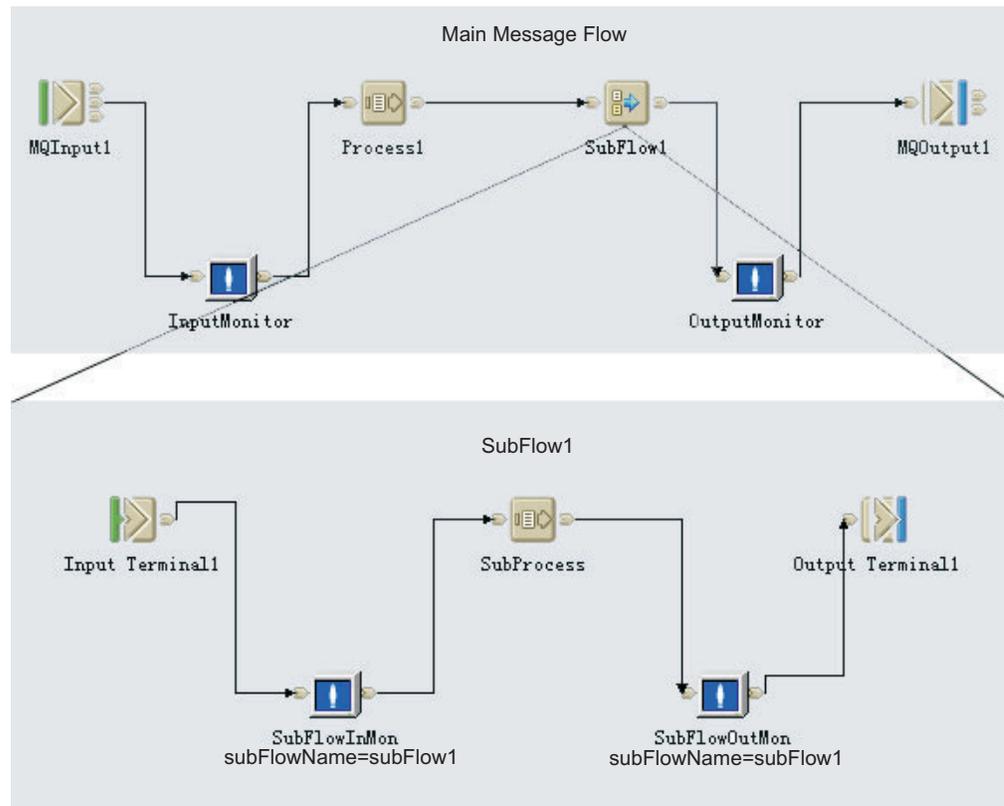


Figure 7. Type I subflow

Remember: Subflows embedded in subflows, or nested subflows, are supported. However, the **subFlowName** attribute for any nested subflows must be different for different subflows. You must be aware of the names assigned to subflow monitoring nodes and not use them again.

Monitoring a Type II subflow

To monitor a Type II subflow, do the following steps:

1. Place a CandleMonitor node at any designated starting place in the message flow.
2. Set the **type** attribute of the CandleMonitor node to subFlowInput, and specify the **subFlowName** attribute.

Remember: For Type II message flows, you need to use a different name that uniquely describes what is being monitored from the name that was specified in the Message Brokers Toolkit for the message flow to represent the subflow.

3. Place another CandleMonitor node at a corresponding end point in the message flow.
4. Set the **type** attribute of the CandleMonitor node to subFlowOutput, and specify the **subFlowName** attribute. (see Figure 8)

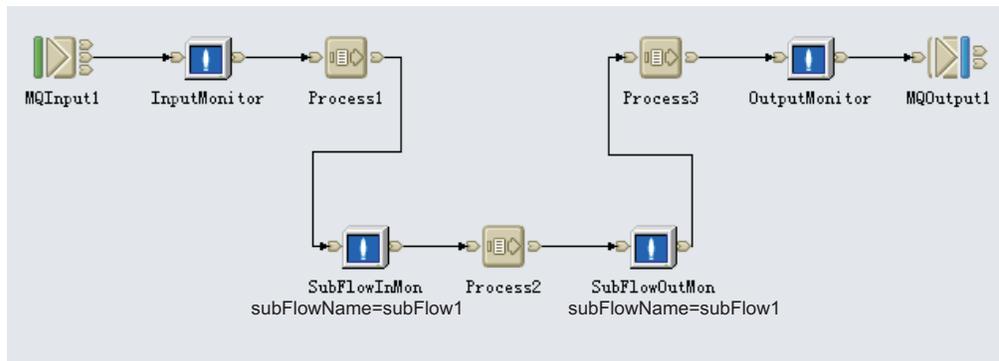


Figure 8. Type II subflow

Remember:

- If the section of message flow that is to be monitored has multiple input connectors or multiple output connectors, you must add multiple CandleMonitor nodes in the same way with the same **subFlowName**.
- Subflows embedded in subflows, or nested subflows, are supported. However, the **subFlowName** attribute for any nested subflows must be different for different subflows. You must be aware of the names assigned to subflow monitoring nodes and not use them again.

Monitoring other aspects of a message flow

You can place the CandleMonitor node anywhere in a message flow and gather statistics for that particular portion of the message flow.

To monitor aspects of a message flow other than the input or output, perform the following procedure:

1. Place a CandleMonitor node in a place other than the beginning or end of a flow.
2. Set the **type** attribute of the CandleMonitor node to other. The statistics can be viewed in the low-level statistics and Monitor Node Base Statistics workspace/CandleMonitor Node Statistics workspace in Tivoli Enterprise Portal.

Producing event messages

You can use the CandleMonitor node to produce user-defined message flow events that can be detected by Message Flow Event situations and viewed in the Monitor

Node Events workspace/Message Flow Events workspace. The WebSphere Message Broker Monitoring agent includes other information about events to help identify which portion of the message flow is affected, as well as to isolate which message is being processed at the time the event occurs.

To use the CandleMonitor node to produce user-defined message flow events, do the following steps:

1. Place a CandleMonitor node at any designated place.

Tip: Do not place the CandleMonitor node in portions of the message flow that are used regularly during typical processing, because this might produce a large number of events. Place the CandleMonitor node in paths of a message flow that represent processing failures or other irregular conditions that warrant an alert notification. In this capacity, the CandleMonitor node serves only as an alert mechanism; it cannot perform error recovery because it is a pass-through node only.

2. Set the **type** attribute of the CandleMonitor node to other.
3. Set the **eventMessage** attribute to the message text that you want to display in the workspace when a message enters the CandleMonitor node.

Tip: Do not set the **eventMessage** attribute for any input or output type node, because this setting might produce an unnecessarily large number of events.

Guidelines for monitoring with the CandleMonitor node

See the following guidelines for monitoring with the CandleMonitor node:

- For production flow statistics, place the CandleMonitor node immediately after MQInput (or other input) nodes and set the **type** attribute to input.
- For output rates and counts, place the CandleMonitor node immediately in front of MQOutput, MQReply, and Publication nodes and set the **type** attribute to output.
- For any other placement in the flow, do not set the **type** attribute to input or output, because inaccurate data might be generated.
- Generally, an **eventMessage** attribute must be specified only for nodes with a type attribute of subFlowInput, subFlowOutput, and other. Do not set this attribute for the typical processing part of a flow or for a node with a type attribute of input or output (unless the node is not used frequently and there is a specific reason for doing so) because it might have an adverse effect on system performance.
- For message flow development, nodes that have the type of other can be used to determine areas of a flow that have poor performance.
- If you are going to shut down the broker or agent, the broker or agent must complete the entire shutdown process whenever possible.

Customizing a CandleMonitor node

Some aspects of the CandleMonitor node behaviors are controlled by configuration variables. You can modify the values of these variables to suit your requirements. After changing the value of any of these variables, you must restart the broker before the changes can take effect.

Important: All of the kqipnode variable names and values are case sensitive. You must enter the variable names exactly as shown. For example, if you are specifying the no override value for the **KQIActivateNode** attribute, you must specify NoOverride.

Windows systems: Changing the values of configuration variables

On Windows operating systems, configuration variables are available in the registry that is created when you configure the monitoring agent.

To change the values of configuration variables on Windows systems, do the following steps:

1. Stop the WebSphere Message Broker Monitoring agent that has variables you want to edit.
2. From the **Start** menu, click **Programs > IBM Tivoli Monitoring > Manage Tivoli Monitoring Services**.
3. Right-click the WebSphere Message Broker Monitoring agent and click **Advanced > Edit Variables**.
4. Do one of the following procedures, depending on whether the variable that you want to edit is in the list:
 - If the variable is in the list, do the following steps:
 - a. Select the variable and click **Edit**.
 - b. Enter the new value in the **Value** field and click **OK**.
 - If the variable is not in the list, do the following steps:
 - a. Click **Add**.
 - b. In the **Variable** list, click the arrow to display the list.
 - c. Select the variable that you want to add from the variable list.
 - d. Enter the new value in the **Value** field and click **OK**.

See “Configuration variables” on page 261 for descriptions of the variables and their valid values.

5. Click **OK**.
6. You are prompted if you want to update the `kqi.xml` file before you configure the WebSphere Message Broker Monitoring agent. Click **No**.

Important: All of the kqipnode variable names and values are case sensitive. You must enter the variable names exactly as shown. For example, if you are specifying the no override value for the **KQIActivateNode** attribute, you must specify NoOverride.

UNIX or Linux systems: Changing the values of configuration variables

On UNIX and Linux operating systems, the configuration variables are stored in the `kqipnode.cfg` file in the `<install_dir>/<arch_code>/qi/bin` directory, where `<install_dir>` is IBM Tivoli Monitoring installation directory and `<arch_code>` is architecture code for the operating system on which the agent is running. For a list of architecture codes, see Appendix D, “Architecture codes,” on page 337.

To change the values of configuration variables on UNIX or Linux systems, edit the `kqipnode.cfg` file directly. The format of the configuration file is as follows:

variable value
variable value

See “Configuration variables” on page 261 for descriptions of the variables and their valid values.

Important: All of the `kqipnode` variable names and values are case sensitive. You must enter the variable names exactly as shown. For example, if you are specifying the no override value for the `KQIActivateNode` attribute, you must specify `NoOverride`.

Deleting the CandleMonitor node from Message Brokers Toolkit

This section describes how to delete the CandleMonitor node from different versions of Message Brokers Toolkit.

Message Brokers Toolkit V6.0: Deleting the CandleMonitor node

To delete the CandleMonitor node, do the following steps:

1. Close the toolkit if it is active.
2. Perform one of the following steps, depending on which type of system the Message Brokers Toolkit is installed on:
 - If the Message Brokers Toolkit is installed on a Windows system, delete any subdirectories with names beginning with `com.candle.monitor` from the following directory (or an alternative location if the Message Brokers Toolkit is not installed in the default location):

```
WMBT_installdir\evtoolkit\eclipse\plugins
```

where, *WMBT_installdir* is the Message Brokers Toolkit installation directory.

- If the Message Brokers Toolkit is installed on a Linux or UNIX system, delete any subdirectories with names beginning with `com.candle.monitor` from the following directory (or an alternative location if the Message Brokers Toolkit is not installed in the default location):

```
WMBT_installdir/evtoolkit/eclipse/plugins
```

where, *WMBT_installdir* is the Message Brokers Toolkit installation directory.

Message Brokers Toolkit V6.1: Deleting the CandleMonitor node

To delete the CandleMonitor node, do the following steps:

1. Close the toolkit if it is active.
2. Perform one of the following steps, depending on which type of system the Message Brokers Toolkit is installed on:
 - If the Message Brokers Toolkit is installed on a Windows system, delete any subdirectories with names beginning with `com.candle.monitor` from the following directory (or an alternative location if the Message Brokers Toolkit is not installed in the default location), where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

```
WMBT_installdir\plugins
```

- If the Message Brokers Toolkit is installed on a Linux or UNIX system, delete any subdirectories with names beginning with `com.candle.monitor` from the

following directory (or an alternative location if the Message Brokers Toolkit is not installed in the default location), where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

WMBT_installdir/plugins

Message Brokers Toolkit V7.0 or later: Deleting the CandleMonitor node

To delete the CandleMonitor node, do the following steps:

1. Close the toolkit if it is active.
2. Perform one of the following steps, depending on which type of system the Message Brokers Toolkit is installed on:
 - If the Message Brokers Toolkit is installed on a Windows system, delete any subdirectories with names beginning with `com.candle.monitor` from the following directory (or an alternative location if the Message Brokers Toolkit is not installed in the default location), where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

WMBT_installdir\evtoolkit\eclipse\plugins

- If the Message Brokers Toolkit is installed on a Linux or UNIX system, delete any subdirectories with names beginning with `com.candle.monitor` from the following directory (or an alternative location if the Message Brokers Toolkit is not installed in the default location), where *WMBT_installdir* is the Message Brokers Toolkit installation directory.

WMBT_installdir/evtoolkit/eclipse/plugins

WebSphere Message Broker V8 support: Known limitations and problems

The following limitations or problems are identified when you use the CandleMonitor node with WebSphere Message Broker V8.

CandleMonitor node statistics is not available

Limitation: When a CandleMonitor node is placed in a message flow and this message flow is deployed multiple times within an execution group, the agent cannot get statistical data to display in the following workspaces:

- Monitor Node Base Statistics
- Monitor Node Broker Statistics
- Monitor Node Event Statistics
- Monitor Node Execution Group Statistics
- Monitor Node Message Flow Statistics
- Monitor Node Sub-Flow Statistics

To avoid this problem, if a message flow is deployed with a CandleMonitor node, deploy it only once in an execution group. For example, if you deploy this message flow directly within an execution group, make sure that other containers (application or library) in this execution group do not contain this message flow.

CandleMonitor node cannot be placed in deployable subflows

Problem: CandleMonitor node cannot be placed in deployable subflows using WebSphere Message Brokers Toolkit 8.0.

| **Solution:** This problem happens when you use WebSphere Message Brokers
| Toolkit 8.0.0.0. To use CandleMonitor node in deployable subflows, apply the fix of
| PMR 63949,000,672 or upgrade to WebSphere Message Brokers Toolkit 8.0.0.1 or
| later.

Chapter 4. Using situations and Take Action commands

This section describes the predefined situations and Take Action commands that are included with WebSphere Message Broker Monitoring agent.

Predefined situations

The WebSphere Message Broker Monitoring agent provides two types of predefined situations, agent-level situations and broker-level situations. Most predefined situations are designed to help you to monitor critical activities and serve as templates for creating customized situations for your own use. There are two agent-level situations, `QI_Automation_Start_Component` and `QI_Product_Events`. The broker-level predefined situations for this agent have different names according to the version of brokers. The situation names begin with the letters **WMB** for brokers of version 7.0 or later, and **QI** for brokers of earlier versions.

- For brokers of version 7.0 or later:
 - `WMB_Average_Flow_Time_High`
 - `WMB_Broker_Not_Started`
 - `WMB_Broker_QMgr_Not_Connected`
 - `WMB_Exception_Terminal_Invoked`
 - `WMB_Monitor_Node_Events`
 - `WMB_MsgFlow_Elapsed_Time_High`
- For brokers of earlier versions:
 - `QI_Average_Flow_Time_High`
 - `QI_Broker_Not_Started`
 - `QI_Broker_QMgr_Not_Connected`
 - `QI_Exception_Terminal_Invoked`
 - `QI_Message_Flow_Events`
 - `QI_MsgFlow_Elapsed_Time_High`
 - `QI_Publication_Expiry_Event`
 - `QI_Status_Stop_Event`
 - `QI_Subscription_Expiry_Event`

Predefined situations are activated once they are distributed to the node that you wish to monitor. Once they are configured correctly, the situation alerts provided with the WebSphere Message Broker Monitoring agent will trigger event notification. You can edit these situations to better reflect your site-specific standards or requirements.

Remember: When you distribute a situation to the managed system, in the available managed system group list, `MQSI_BROKER_V7` is the group of all managed systems for WebSphere Message Broker V7.0 or later. `MQSI_BROKER` is the group of managed systems for WebSphere Message Broker V6 (all releases).

Tip: If you choose to modify a predefined situation, make a copy first to ensure fallback if necessary.

Alerts

The following situations issue alerts based on critical status criteria. Except where noted, these situations apply to broker-managed systems. Alerts are raised if the following conditions are met:

- A Product Events row occurs. This situation applies to agent-managed systems. (QI_Product_Events)
- The Component.Component Status is found to be not started. If the situation becomes true, the proper command is issued to start the component. This situation applies to agent-managed systems. (QI_Automation_Start_Component)
- The Message Flow Statistics.Average Flow Seconds value is found to exceed a threshold. (WMB_Average_Flow_Time_High/QI_Average_Flow_Time_High)
- The Broker Information.Broker Status value is found to be not started. (WMB_Broker_Not_Started/QI_Broker_Not_Started)
- The Broker Information.Queue Manager Status value is found to be not connected. (WMB_Broker_QMgr_Not_Connected/QI_Broker_QMgr_Not_Connected)
- Messages have gone down a failure path in a message flow. You can create a new situation based on this situation using parameters that are appropriate for your environment. In this case the links included in the predefined situation are invalid. You can create a new link using the Link Wizard. (WMB_Exception_Terminal_Invoked/QI_Exception_Terminal_Invoked)
- A Message Flow Events row occurs. (WMB_Monitor_Node_Events/QI_Message_Flow_Events)
- The maximum time that any message takes to go through a given message flow during an archive statistics interval exceeds 100 milliseconds. You can create a new situation based on this situation using a threshold value that is appropriate for your environment. In this case the links included in the predefined situation are not valid. You can create a new link using the Link Wizard. (WMB_MsgFlow_Elapsed_Time_High/QI_MsgFlow_Elapsed_Time_High)
- A Broker Events row occurs for a Status Stop event. (QI_Status_Stop_Event)
- A Broker Events row occurs for a Publication Expiration event type. (QI_Publication_Expiry_Event)
- A Broker Events row occurs for a Subscription Expiration event type. (QI_Subscription_Expiry_Event)

Take Action commands

You can use Take Action commands to send commands from the Tivoli Enterprise Portal to systems in your managed enterprise. You can use it to, for example, start or stop a component. These commands otherwise must be entered from a command prompt, or from the Message Brokers Toolkit.

You can issue the commands from the Take Action view, from the situation window when a situation becomes true, from the Navigator, or from a row in a table view. For more information about how to run a Take Action command, see Tivoli Enterprise Portal online help.

The Take Action commands for the WebSphere Message Broker Monitoring agent have names beginning with the characters **QI**. For more details about the Take Action commands included in the WebSphere Message Broker Monitoring agent, see the following references:

- “Commands for agent managed systems” on page 264

- “Commands for broker-managed systems” on page 265
- “Command restrictions on z/OS systems” on page 270

Authorizing Take Action users

You can configure the WebSphere Message Broker Monitoring agent so that only a designated set of users can issue Take Action commands. You can specify authorized Take Action users in the following ways:

- At the agent level, use the **defaultTakeActionAuthUsers** parameter.
- At the monitored broker level, use the **takeActionAuthUsers** parameter.

See “Agent parameter descriptions” on page 18 for details.

Sending a Take Action command

Perform the following steps to send a Take Action command from the Tivoli Enterprise Portal to systems in your managed enterprise:

1. Select the Navigator item that is associated with the component or application on which you want to run the command.
2. Right-click the Navigator item. You can also right-click a row in a table view or a bar in a bar chart.
3. Click **Take Action > Select**. The Take Action window is displayed.
4. In the **Name** field, click the arrow to display the list.
5. Select the command that you want to run from the **Name** list. For example, **QI Start Broker**.
6. In the Edit Argument Values window, enter the name of the component on which you want to run the command in the **Value** column. For example, if you chose **QI Start Broker** in the previous step, enter the name of the broker that you want to start. Click **OK**.
7. Click the host system in **Destination Systems** where the component is located and click **OK**.

A message is displayed indicating the status of the action after the Take Action command is sent.

Using Take Action commands in situations

You can use Take Action commands in situations that you create. The *reflex automation* term refers to a situation that issues a command.

To issue a command to an agent from a situation, enter the *QI:syntax* form of the command on the **Action** page of the Situations Editor. Figure 9 on page 62 illustrates an example of using the **QI Start Component** command in a situation.

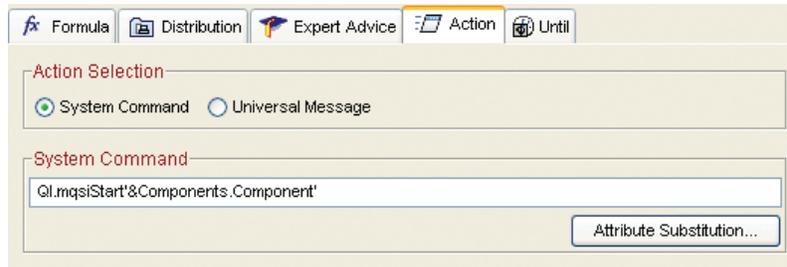


Figure 9. Example of using the QI Start Component command in a situation

This example displays the predefined situation, `QI_Automation_Start_Component`, which substitutes an attribute from the Components workspace of the product. The **Attribute Substitution** button is used to choose the attribute (this prevents misspellings and inserts the ampersand character which is used for attribute substitution). The single quotation marks enclosing the command parameter are required. When you are using a Take Action command in a situation, create a working, manual version of the Take Action command before you attempt to automate its use by adding it to a situation.

Scenarios of using Situations and Take Action commands

This section provides several examples of monitoring situations that are common to many broker product environments, and illustrates how situations and Take Action commands can be used in these circumstances.

Each scenario presents a case in which you want to monitor some aspect or component of your broker product environment, followed by a description of how to set up the WebSphere Message Broker Monitoring agent to gather important information.

Preventing inadvertent use of trace active

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to prevent the trace active feature from being used inadvertently and adversely affecting broker performance.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that the necessary statistics can be gathered, and then perform the following steps:

1. Define a situation for the following workspaces that verifies that the Trace Level and User Trace Level attribute values are not none. An alert is triggered whenever tracing is active so that you can determine whether the tracing activity is valid. For information about defining a situation, see the Tivoli Enterprise Portal online help or *Tivoli Enterprise Portal User's Guide, SC32-9409*.

- WebSphere Message Broker V7.0 or later:
 - Broker Status
 - Execution Group Status
 - Message Flow Status
- WebSphere Message Broker V6 (all releases):
 - Broker Information
 - Execution Group Information
 - Message Flow Information

2. Deploy the situation to the broker managed systems where you want to detect active trace.

Determining when a message flow has failed

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to determine when a message flow has failed and to notify you of a failure in a message flow.

To determine when a message flow has failed, do the following steps so that the WebSphere Message Broker Monitoring agent can gather the appropriate statistics:

1. Place CandleMonitor nodes after failure terminals for processing nodes in the message flow.
2. Set the **type** attribute of these CandleMonitor nodes to other.
3. Describe the failures in a meaningful way in the **eventMessage** attribute of the CandleMonitor nodes.
4. Start the WebSphere Message Broker Monitoring agent again.

After you have positioned the CandleMonitor nodes and the WebSphere Message Broker Monitoring agent is running, do the following steps:

1. Define situations for message flow events so that an alert is raised each time a message flows down a path that is failing.
2. Deploy the situations that you define in the previous step to the system on which the monitored message flow is running.
3. Open the Monitor Node Events workspace/Message Flow Events workspace to see data about the message being processed when the events that you define occur.

Collecting requested system trace data for a broker on a remote system

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to collect system trace data requested by IBM Software Support for a broker that is located on a remote system.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that the necessary statistics can be gathered, and then do the following steps:

1. Log on to the Tivoli Enterprise Portal.
2. In the Navigator view, right-click the Navigator item of a broker and click **Take Action > Select**. The Take Action window is displayed.
3. In the **Name** field, click the arrow to display the list.
4. Select **QI Change Trace Broker** from the **Name** list. The Edit Argument Values window is displayed.
5. In the Edit Argument Values window, specify the required argument values for the command, and then click **OK**. For detailed information about the **QI Change Trace Broker** command, see “WMB Change Trace Broker/QI Change Trace Broker command” on page 266.
6. In the Take Action window, select the broker managed system for the broker that needs to be traced from the **Destination Systems** list and click **OK**.

Stopping a message flow that has a full output queue

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to quickly stop a message flow that is failing because it has a full output queue. This scenario is intended for users who do not always start the Configuration Manager and Message Brokers Toolkit so that they can conserve system resources.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that the necessary statistics can be gathered, and then do the following steps:

1. Log on to the Tivoli Enterprise Portal.
2. In the Navigator view, right-click the Navigator item of a broker and click **Take Action > Select**. The Take Action window is displayed.
3. In the **Name** field, click the arrow to display the list.
4. Select **QI Stop Message Flow(s)** from the **Name** list. The Edit Argument Values window is displayed.
5. In the Edit Argument Values window, set the arguments to the appropriate execution group and message flow and click **OK**. For detailed information about the **QI Stop Message Flow(s)** command, see “WMB Stop Message Flow(s)/QI Stop Message Flow(s) command” on page 268.
6. In the Take Action window, select the broker managed system for the broker with the failing message flow from the **Destination Systems** list and click **OK**.

Automatically starting a broker that is stopped

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to automatically restart a broker whenever it is stopped.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that necessary statistics can be gathered, and then perform the following steps:

1. Define a situation for the Broker Status workspace/Broker Information workspace that checks whether the broker status is stopped. For information about how to define a situation, see Tivoli Enterprise Portal online help and *Tivoli Enterprise Portal User's Guide, SC32-9409*.
2. Set the action for the situation to use the **QI Start Broker** command to restart the broker. For additional details about the **QI Start Broker** command, see “WMB Start Broker/QI Start Broker command” on page 265.

Tip: There is a predefined situation for the Components workspace that you can use for the same purpose, but that situation also starts the Configuration Manager and User Name Server, if either one stops.

Starting and stopping message flows at periodic intervals

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to automatically start and stop message flows at certain times of the day.

For individual message flows

To start and stop individual message flows at intervals, after you have configured the WebSphere Message Broker Monitoring agent, start the agent so that you can gather the necessary statistics, and then perform the following procedure:

1. Define a situation for the **Status** attribute in the Message Flow Status workspace/Message Flow Information workspace. Use the **Local Time** attribute group to specify when a message flow must be started or stopped. For information about defining a situation, see Tivoli Enterprise Portal online help or *Tivoli Enterprise Portal User's Guide, SC32-9409*.
2. Set the action for the situation to use the **QI Start Message Flow(s)** or **QI Stop Message Flow(s)** command to start or stop the message flow; specify both the execution group name and the message flow name. For additional details about the **QI Start Message Flow(s)** or **QI Stop Message Flow(s)** commands, see "WMB Start Message Flow(s)/QI Start Message Flow(s) command" on page 268 and "WMB Stop Message Flow(s)/QI Stop Message Flow(s) command" on page 268.

For all message flows within an execution group

To start and stop all message flows within an execution group at intervals, after you have configured the WebSphere Message Broker Monitoring agent, start the agent so that you can gather the necessary statistics, and then do the following steps:

1. Define a situation for the **Started Message Flows** attribute in the Execution Group Status workspace/Execution Group Information workspace. Use the **Local Time** attribute group to specify when all message flows in the execution group must be started or stopped. For information about defining a situation, see Tivoli Enterprise Portal online help or *Tivoli Enterprise Portal User's Guide, SC32-9409*.
2. Set the action for the situation to use the **QI Start Message Flow(s)** or **QI Stop Message Flow(s)** command to start or stop all message flows. Specifying only the execution group name argument and do not specify a message flow name argument. For additional details about the **QI Start Message Flow(s)** or **QI Stop Message Flow(s)** commands, see "WMB Start Message Flow(s)/QI Start Message Flow(s) command" on page 268 and "WMB Stop Message Flow(s)/QI Stop Message Flow(s) command" on page 268.

Chapter 5. Monitoring with workspaces

The WebSphere Message Broker Monitoring agent is installed with default views that are displayed in workspaces. Where applicable, links are provided in the workspace to link from a parent view to a more detailed view of a selected row, or to a related workspace (for example, a workspace containing historical information).

You can customize the format and appearance of the views in a workspace.

Tip: Customize a copy of the workspace. Use the **Save as** command first to copy the workspace and give it a new name, and then make changes to the copy. You can set the workspace that you create in this manner to be the default workspace for a given Navigator item. If you do not replace the product-provided workspaces, you can see new features that might become available in later versions of WebSphere Message Broker Monitoring agent without having to revert to the original workspace. For information on customizing workspaces and views, see the Tivoli Enterprise Portal online help.

The following sections provide information about workspace groups. For detailed descriptions of individual attributes in a workspace, see “Attributes” on page 137.

Before you begin

Before you begin, there are several things to understand about how the workspace works:

- Some prior conditions must be met, so that the WebSphere Message Broker Monitoring agent can collect data for certain workspaces. For information about the related workspaces, see “Data availability.”
- Two types of statistical data are provided, broker accounting statistics data and CandleMonitor node statistics data. Both types of data can provide information about message flows in a broker. To decide which type best suits your environment, see “Comparison between broker accounting statistics data and CandleMonitor node statistics data” on page 69.

Data availability

Put your short description here; used for first paragraph and abstract.

The availability of some data and statistics depends on prior conditions. If these conditions are not met, no information is available in the workspace. In particular:

- For archive accounting statistics to be available in the following workspaces, configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the **WMB Change Flow Stats/QI Change Flow Stats** command. For detailed information about this Take Action command, see “WMB Change Flow Stats/QI Change Flow Stats command” on page 267:

- WebSphere Message Broker V7.0 or later:
 - “Archive Accounting Message Flow Statistics workspace” on page 271
 - “Archive Accounting Thread Statistics workspace” on page 274
 - “Archive Accounting Node Statistics workspace” on page 272

- "Archive Accounting Terminal Statistics workspace" on page 273
- WebSphere Message Broker V6 (all releases):
 - "Archive Message Flow Accounting workspace" on page 290
 - "Archive Thread Accounting workspace" on page 293
 - "Archive Node Accounting workspace" on page 291
 - "Archive Terminal Accounting workspace" on page 292
- For snapshot accounting statistics to be available in the following workspaces, configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the **WMB Change Flow Stats/QI Change Flow Stats** command. For detailed information about this Take Action command, see "WMB Change Flow Stats/QI Change Flow Stats command" on page 267:
 - WebSphere Message Broker V7.0 or later:
 - "Snapshot Accounting Message Flow Statistics workspace" on page 285
 - "Snapshot Accounting Thread Statistics workspace" on page 288
 - "Snapshot Accounting Node Statistics workspace" on page 286
 - "Snapshot Accounting Terminal Statistics workspace" on page 287
 - WebSphere Message Broker V6 (all releases):
 - "Snapshot Message Flow Accounting workspace" on page 308
 - "Snapshot Thread Accounting workspace" on page 310
 - "Snapshot Node Accounting workspace" on page 309
 - "Snapshot Terminal Accounting workspace" on page 309
- For user statistics to be available, issue the **WMB Create User Statistics/QI Create User Statistics** command for the Tivoli Enterprise Portal user ID. For detailed information about this Take Action command, see "WMB Create User Statistics/QI Create User Statistics command" on page 269.
- For message flow and subflow statistics and events to be available in the following workspaces, CandleMonitor nodes must be included in the message flow (see Chapter 3, "Monitoring with the CandleMonitor node," on page 39):
 - WebSphere Message Broker V7.0 or later:
 - "Monitor Node Broker Statistics workspace" on page 280
 - "Monitor Node Base Statistics workspace" on page 279
 - "Monitor Node Execution Group Statistics workspace" on page 281
 - "Monitor Node Events workspace" on page 280
 - "Monitor Node Message Flow Statistics workspace" on page 282
 - "Monitor Node Sub-Flow Statistics workspace" on page 282
 - WebSphere Message Broker V6 (all releases):
 - "Broker Statistics workspace" on page 295
 - "CandleMonitor Node Statistics workspace" on page 297
 - "Execution Group Statistics workspace" on page 299
 - "Message Flow Events workspace" on page 300
 - "Message Flow Statistics workspace" on page 301
 - "Sub-Flow Statistics workspace" on page 311
- (WebSphere Message Broker V6 all releases only) For publish-subscribe statistics to be available in the following workspaces, issue the **QI Change Properties** command. For detailed information about this Take Action command, see **QI Change Properties** command:

- “Publish-Subscribe Statistics workspace” on page 307
- “Multicast Summary Statistics workspace” on page 304
- “Multicast Group Statistics workspace” on page 304
- “Multicast Topic Statistics workspace” on page 305
- For accounting origin information to be available in accounting workspaces, participating message flows must be configured to provide the appropriate origin identifier. If you do not specify a value, the Accounting Origin column in the Accounting workspaces contains the anonymous value.

Comparison between broker accounting statistics data and CandleMonitor node statistics data

The WebSphere Message Broker Monitoring agent provides two types of statistical data to monitor message flows in a broker:

- Broker accounting and statistics data
- CandleMonitor node statistics data

You can use the information in this topic to make a choice that best suits your environment between the two data types.

Broker accounting and statistics data

When statistics are collected for the broker and XML format is specified as the output destination, the WebSphere Message Broker Monitoring agent reports on accounting and statistics data that is produced by the broker. The agent automatically subscribes to the broker to receive this type of data.

For the accounting and statistics data, you only need to use the **mqsichangeflowstats** command to enable statistics collection at the broker, and after that you can see the data in the Tivoli Enterprise Portal workspaces. With the **mqsichangeflowstats** command, you can indicate whether you want to do the following things:

- Collect statistics for a specific message flow
- Collect statistics for all message flows
- Collect thread related statistics
- Collect node related statistics (including terminals for the nodes)

All these data are supported by the WebSphere Message Broker Monitoring agent.

Tip: You can issue the **mqsichangeflowstats** command to the broker-managed systems at any time by using the predefined **QI Change Flow Stats** command. For more information about how to issue this Take Action command for Tivoli Enterprise Portal, see “Sending a Take Action command” on page 61 and “WMB Change Flow Stats/QI Change Flow Stats command” on page 267.

There are two types of data collection: snapshot and archive. Although they are used for different purposes, data of both types support the same levels of details about message flows, threads, nodes, and terminals.

Archive data is intended for use in more long-term accounting and statistics data. This data is the type that you want to collect continuously for general monitoring of message flows. Archive data is collected at an interval that you can configure, with the minimum interval being 1 minute. The default interval for this data is 60 minutes. To change this interval, use the **mqsichangebroker** command with the **-v**

parameter or the **QI Change Broker** command from Tivoli Enterprise Portal. The broker must be stopped when the command is issued.

Snapshot data is the type that you want to collect for a short period of time when you are troubleshooting a problem in one or more message flows. The snapshot data is collected every 20 seconds, and you cannot change this interval. Performance can be affected by collecting snapshot data, and the agent can be impacted by the amount of data published. So you must be careful to enable snapshot data collection only for particular message flows being debugged when you need it and to disable it when finished debugging. You can enable or disable the snapshot data collection with the **QI Change Flow Stats** command from Tivoli Enterprise Portal.

There are four levels of data: message flow, thread, node, and terminal. The attributes for the accounting and statistics data vary for different levels. The data of message flow level include elapsed and CPU timings, input and output message counts and sizes, and various special or error counts. The data of thread level report on threads that process message flows and includes CPU and Elapsed timings, and message size and rate. The node level data report on elapsed and CPU timings for each node in a message flow. The terminal level data present counts of invocations of the various terminals for nodes in the message flow.

A set of workspaces is available to display the following accounting and statistics data. And situations can be targeted at the most current interval data to automatically detect problems.

- Data collected during the most current interval
- Data collected during several most recent intervals for trending
- Historical data (when historical data collection is enabled)

The historical data collection function can track the archive statistics for accounting purposes without purchasing or developing a second application.

If you intend to use the accounting origin support to organize your data, you must configure participating message flows to provide the appropriate origin identifier. As described in the WebSphere Message Broker documentation, this involves coding an ESQL statement in either a Compute, Database, or Filter node that sets the value that you want. In addition, you must specify the **-b basic** parameter on the **mqsichangeflowstats** command that you use to start data collection.

CandleMonitor node statistics data

CandleMonitor node statistics data are produced by the CandleMonitor node that are placed in message flows. The CandleMonitor node can be placed multiple times within message flows, depending on the amount of monitoring that you want. To use the CandleMonitor node for monitoring, you must do the following things:

1. Make the CandleMonitor node available in the Message Broker Toolkit.
2. Modify the message flow.
3. Redeploy the message flow to the broker.

Tip: For more information about the usage of CandleMonitor node, see Chapter 3, “Monitoring with the CandleMonitor node,” on page 39.

For typical monitoring, it is sufficient to place one CandleMonitor node at the beginning of the flow (after input node) and one CandleMonitor node at the end of the flow (before the output node). Only the message flows with at least one CandleMonitor node are represented in the reported data.

You can also configure the CandleMonitor node to produce user-defined message flow events for situation detection and Tivoli Enterprise Portal display of problems. For example, a message flows down a failure path, and the CandleMonitor node can automatically report exceptions that are propagated from any node as message flow events. Configuration parameters for activating certain nodes can also be used to disable different types of nodes from reporting data to scale back data collection. So you can place many nodes during message flow development (for example, to monitor sub-flows in a more granular pattern) and leave them in place later when moving to production, although they are inactive.

The CandleMonitor node implementation is provided at the broker by the `kqipnode.lil` file, which must be made available to the broker before deploying a message flow with the node. When the broker initializes the `kqipnode.lil` file, it will set up a shared memory area for recording the statistics. When a message flows through the CandleMonitor node, a little processing of data is required here. Instead, the WebSphere Message Broker Monitoring agent reads the shared memory and does all the calculations and summarization of data. So the CandleMonitor node has little impact on the message flow. By default, the agent does this work at a 1-minute interval. If there is any message flow event that is posted by a CandleMonitor node, the WebSphere Message Broker Monitoring agent also picks up the event from shared memory. And this occurs every 15 seconds by default. You can use the agent configuration parameters to modify these intervals.

Basically, the same set of statistics giving elapsed timings, input and output message counts, and queue timings are available at the following summarization levels:

- CandleMonitor node
- Sub-flow
- Message flow
- Execution group
- Broker

The base data that are collected by the agent are reported in Monitor Node Base Statistics workspace/CandleMonitor Node Statistics workspace. Data in Sub-Flow Statistics workspace are summarized for each subflow that is delineated by the CandleMonitor nodes of subflow type. Data in Monitor Node Message Flow Statistics workspace/Message Flow Statistics workspace are summarized for each message flow with at least one CandleMonitor node of input type. The statistics are summarized also at the execution group level in Monitor Node Execution Group Statistics workspace/Execution Group Statistics workspace and at the broker level in Monitor Node Broker Statistics workspace/Broker Statistics workspace.

You can also use the **WMB Create User Statistics/QI Create User Statistics** command to create what are called collectively as user statistics. These are the same statistics and levels, except that you collect the data when you want by issuing a Take Action command named **WMB Sample User Statistics/QI Sample User Statistics**. Issue one **WMB Sample User Statistics/QI Sample User Statistics** command when you want the interval to begin; issue another **WMB**

Sample User Statistics/QI Sample User Statistics command when you want the interval to end. This type of collection is useful, for example, to gather statistics for a certain set of messages flowing.

Cumulative statistics with **Overall** attribute names are maintained internally at the CandleMonitor node level. These statistics are simultaneously reset to 0 for all CandleMonitor nodes when a deploy operation to a broker involves any message flow containing a CandleMonitor node, when any CandleMonitor node detects an overflow condition for statistics, or when the Reset Statistics Take Action command has been issued.

A reset of statistics includes all statistics that are maintained for the broker to preserve the integrity of summarized statistics. The CandleMonitor node produces an Event Log message when a reset occurs because of a numeric overflow condition, and the monitoring agent logs a message when a reset is detected.

At the time of a reset of statistics, workspaces displays that the **Overall** statistics have started over from zero.

Historical workspaces displays data before the reset combined with data following the reset for the interval in which the reset occurred. This ensures that no historical data is lost. In subsequent intervals, the historical workspaces displays the **Overall** values as having started over from zero.

Table 1. Comparison between broker accounting statistics and CandleMonitor node statistics

Category	Similarities	Differences	
		Broker accounting and statistics	CandleMonitor node statistics
Data attributes	Both types have the following attributes: <ul style="list-style-type: none"> • Elapsed Timings • Message Counts • Message Sizes • Byte Rates 	<ul style="list-style-type: none"> • Special error counts at the message flow level • CPU Timings • Invocation counts for nodes and terminals 	<ul style="list-style-type: none"> • Queue Time • Message Flow Events

Table 1. Comparison between broker accounting statistics and CandleMonitor node statistics (continued)

Category	Similarities	Differences	
		Broker accounting and statistics	CandleMonitor node statistics
Data levels	Both types have the message flow level.	<ul style="list-style-type: none"> • Thread • Node • Terminal <p>Data of all levels are available with archive and snapshot accounting.</p>	<ul style="list-style-type: none"> • Broker • Execution group • Sub-flow • CandleMonitor node <p>Data of all levels are available with regular and user-defined statistics. Note: The levels of statistics are all only summarizations of the same base statistics collected, and they are only applicable with respect to where the CandleMonitor node has been deployed. The applications and libraries feature of the broker is not reflected in CandleMonitor node data, which might cause some confusion if the same message flow name occurs in multiple applications or libraries.</p>
Detection of message flowing on a failure path		Accounting and statistics data at the terminal level can be used in a situation to detect that a failure terminal has an invocation count that is greater than zero, which means that a message flowed to a failure path. However no information about the message is available.	CandleMonitor node statistics can be positioned along a failure path in a message flow with the event attribute set to an event message. This message flow event can be displayed at the Tivoli Enterprise Portal and can be detected by situations. The message ID and correlation ID are among the message data available.

Table 1. Comparison between broker accounting statistics and CandleMonitor node statistics (continued)

Category	Similarities	Differences	
		Broker accounting and statistics	CandleMonitor node statistics
Collection interval	The interval can be changed for some or all of the data collection.	Archive interval has a minimum of 1 minutes and a default of 60 minutes. Snapshot interval cannot be changed, and the default is 20 seconds.	The default interval is 1 minutes for statistics (which can be set to less than 1 minute) and 15 seconds for message flow events. The interval must be set in the <code>kqi.xml</code> agent configuration file with the defaultStatisticsInterval and defaultFlowEventInterval parameters. For more information about these parameters, see "Agent parameter descriptions" on page 18.
Performance		Use Archive accounting for regular monitoring. Snapshot accounting should be used only for problem determination and not for regular monitoring.	CandleMonitor node has little impact on a message flow. The interval does not impact the broker performance.
Agent installation		Accounting and statistics data have no extra installation steps.	CandleMonitor node requires root authority for installation of the <code>kqipnode.lil</code> file at the broker on UNIX systems. On z/OS systems, additional steps are required to integrate the node into the broker environment. You must make the CandleMonitor node available in the Message Broker Toolkit separately.
Configuration	Configuration parameters are provided in the <code>kqi.xml</code> configuration file. The WebSphere Message Broker Monitoring agent must be restarted to make the parameter change take effect.	Accounting and statistics data are configured by the mqsichangeflowstats command, which is also available as a Take Action command from the Tivoli Enterprise Portal interface. The configuration can be done dynamically.	Primary configuration of the CandleMonitor node is within the Message Broker Toolkit to implement message flows with the node. The <code>kqipnode.cfg</code> configuration file is available at the broker. The broker must be restarted to make the configuration change take effect.

Table 1. Comparison between broker accounting statistics and CandleMonitor node statistics (continued)

Category	Similarities	Differences	
		Broker accounting and statistics	CandleMonitor node statistics
Reason most often chosen	Both types provide data that are required to determine the situation of a message flow.	Accounting and statistics data are chosen because they are easy to configure with dynamic configuration changes possible. Accounting and statistics data includes CPU timings at the message flow, thread and node levels, various message flow error counts, and node and terminal invocation counts. Also, they support the application and libraries feature of the broker. When using these statistics, you are recommended to use the companion WebSphere MQ monitoring agent to determine queue times.	CandleMonitor node statistics data are chosen because of the desire to use the message flow events feature. Other reasons for choosing in the past have diminished in recent broker releases.

Workspace summary

Use the references in this section to find information about the following workspaces:

- “Agent and application status workspaces”
- “Broker and message flow information workspaces” on page 76
- “Event workspaces” on page 76
- “Statistics workspaces” on page 77
- “Accounting workspaces” on page 78
- “Publish and Subscribe workspaces (WebSphere Message Broker V6 all releases only)” on page 79

Agent and application status workspaces

- The Agent Status workspace summarizes event and broker product component information at the agent level. See “Agent Status Workspace (agent-level summary)” on page 290.
- The Components workspace presents a list of broker product components and their state at the agent level. See “Components workspace” on page 276.
- The Broker Summary workspace summarizes broker status and definition data when multiple brokers are monitored by the same agent. See “Broker Summary workspace (multiple monitored brokers)” on page 296.

- The Broker Status workspace summarizes event and definition information at the broker level. See “Broker Status workspace (broker-level summary)” on page 295.

Broker and message flow information workspaces

- WebSphere Message Broker V7.0 or later:
 - The Broker Status workspace contains information about a broker-managed system, such as the status of the broker, its process ID, information about the operating system where the broker runs, its job on z/OS systems, its associated queue manager. See “Broker Status workspace” on page 275.
 - The Message Flow Node Topology workspace provides a graphical view of a message flow (including its nodes and connection between nodes) and a table view of detailed information about the message processing nodes. See “Message Flow Node Topology workspace” on page 278.
 - The Processing Node Attributes workspace lists all attribute values for the selected message processing node within a message flow. See “Processing Node Attributes workspace” on page 284.
- WebSphere Message Broker V6 (all releases):
 - The Broker Information workspace summarizes broker activity, presents the name of the queue manager that is associated with the broker, and indicates whether the agent can connect to the queue manager of the broker. See “Broker Information workspace” on page 294.
 - The Execution Group Information workspace presents execution groups that are defined for the broker and definition information for each execution group. See “Execution Group Information workspace” on page 298.
 - The Message Flow Information workspace presents defined message flows by execution group and definition information for each. See “Message Flow Information workspace” on page 300.
 - The Message Processing Node Information workspace presents defined message processing nodes by their execution group and message flow. See “Message Processing Node Information workspace” on page 303.
 - The Message Processing Nodes with Queue workspace lists all message processing nodes in the system that have the same queue name (when you link from a row in the Message Processing Node Information workspace). See “Message Processing Nodes with Queue workspace” on page 303.
 - The Message Processing Node Attributes workspace presents *all* attribute values for a message processing node (when you link to the workspace from a row in the Message Processing Nodes/Message Processing Node Information workspace). See “Message Processing Node Attributes workspace” on page 302.

Event workspaces

- WebSphere Message Broker V7.0 or later:
 - The Product Events workspace presents events that are generated by the agent when a problem occurs that affects the agent ability to collect data. See “Product Events workspace” on page 285.
 - The Broker Status Events workspace lists events that are generated by a broker as they occur. See “Broker Status Events workspace” on page 276.
 - The Monitor Node Events workspace lists events that are generated by a CandleMonitor node as they occur. See “Monitor Node Events workspace” on page 280.

- WebSphere Message Broker V6 (all releases):
 - The Product Events workspace presents events that are generated by the agent when a problem occurs that affects the agent ability to collect data. See “Product Events workspace” on page 285.
 - The Broker Events workspace lists events that are generated by a broker as they occur. See “Broker Events workspace” on page 293.
 - The Message Flow Events workspace lists events that are generated by a CandleMonitor node as they occur. See “Message Flow Events workspace” on page 300.

Statistics workspaces

- WebSphere Message Broker V7.0 or later:
 - The Monitor Node Broker Statistics workspace provides summarized message flow statistics at the broker level. See “Monitor Node Broker Statistics workspace” on page 280.
 - The Monitor Node Base Statistics workspace provides message flow statistics at the individual node level at which they are collected. See “Monitor Node Base Statistics workspace” on page 279.
 - The Monitor Node Execution Group Statistics workspace provides summarized message flow statistics at the execution group level. See “Monitor Node Execution Group Statistics workspace” on page 281.
 - The Monitor Node Message Flow Statistics workspace provides summarized message flow statistics at the message flow level. See “Monitor Node Message Flow Statistics workspace” on page 282.
 - The Monitor Node Sub-Flow Statistics workspace provides summarized message flow statistics at the subflow level. See “Monitor Node Sub-Flow Statistics workspace” on page 282.

Remember: After a message flow is restarted, its statistics are reset in the preceding workspaces.

- WebSphere Message Broker V6 (all releases):
 - The Broker Statistics workspace provides summarized message flow statistics at the broker level. See “Broker Statistics workspace” on page 295.
 - The CandleMonitor Node Statistics workspace provides message flow statistics at the individual node level at which they are collected. See “CandleMonitor Node Statistics workspace” on page 297.
 - The Execution Group Statistics workspace provides summarized message flow statistics at the execution group level. See “Execution Group Statistics workspace” on page 299.
 - The Message Flow Statistics workspace provides summarized message flow statistics at the message flow level. See “Message Flow Statistics workspace” on page 301.
 - The Sub-Flow Statistics workspace provides summarized message flow statistics at the subflow level. See “Sub-Flow Statistics workspace” on page 311.

Remember: Message flow accounting and statistics are provided that partially overlap with the statistics provided by the CandleMonitor node and can be monitored in Tivoli Enterprise Portal without including the CandleMonitor node in message flows. However, the CandleMonitor node provides additional statistics, and the ability to monitor sub-flows and define message flow events. Message flows without a CandleMonitor node are not represented in statistics workspaces.

Accounting workspaces

Accounting workspaces provide statistics at message flow, node, thread, and terminal level. The data in these workspaces is sampled at snapshot (short term) or archive (long term) intervals. Top-level snapshot and archive workspaces present current accounting data for that interval. Each workspace links to workspaces that display the data for the most recent collection intervals and for the historical collection intervals, if those intervals are configured.

- WebSphere Message Broker V7.0 or later:
 - The Snapshot Accounting Message Flow Statistics and Archive Accounting Message Flow Statistics workspaces provide statistics at the message flow level. See “Snapshot Accounting Message Flow Statistics workspace” on page 285 and “Archive Accounting Message Flow Statistics workspace” on page 271.
 - The Snapshot Accounting Node Statistics and Archive Accounting Node Statistics workspaces provide statistics at the processing node level. See “Snapshot Accounting Node Statistics workspace” on page 286 and “Archive Accounting Node Statistics workspace” on page 272.
 - The Snapshot Accounting Terminal Statistics and Archive Accounting Terminal Statistics workspaces provide statistics at the terminal level. See “Snapshot Accounting Terminal Statistics workspace” on page 287 and “Archive Accounting Terminal Statistics workspace” on page 273.
 - The Snapshot Accounting Thread Statistics and Archive Accounting Thread Statistics workspaces provide statistics at the thread level. See “Snapshot Accounting Thread Statistics workspace” on page 288 and “Archive Accounting Thread Statistics workspace” on page 274.
- WebSphere Message Broker V6 (all releases):
 - The Snapshot Message Flow Accounting and Archive Message Flow Accounting workspaces provide statistics at the message flow level. See “Snapshot Message Flow Accounting workspace” on page 308 and “Archive Message Flow Accounting workspace” on page 290.
 - The Snapshot Node Accounting and Archive Node Accounting workspaces provide statistics at the processing node level. See “Snapshot Node Accounting workspace” on page 309 and “Archive Node Accounting workspace” on page 291.
 - The Snapshot Terminal Accounting and Archive Terminal Accounting workspaces provide statistics at the terminal level. See “Snapshot Terminal Accounting workspace” on page 309 and “Archive Terminal Accounting workspace” on page 292.
 - The Snapshot Thread Accounting and Archive Thread Accounting workspaces provide statistics at the thread level. See “Snapshot Thread Accounting workspace” on page 310 and “Archive Thread Accounting workspace” on page 293.

Exception: For WebSphere Message Broker V7.0 or later, historical data cannot be collected for the following workspaces:

- Snapshot Accounting Message Flow Statistics
- Snapshot Accounting Node Statistics
- Snapshot Accounting Terminal Statistics
- Snapshot Accounting Thread Statistics

Retained data in accounting workspaces

How many table rows are retained and displayed in accounting workspaces are controlled by three factors:

- Agent configuration

In the agent configuration file, the **defaultRetainRecentSnapshotSamples** parameter determines the size of recent snapshot accounting tables. The **defaultRetainRecentArchiveSamples** parameter determines the size of recent archive accounting tables. Sometimes the number of rows displayed in recent snapshot accounting tables might be larger than the **defaultRetainRecentSnapshotSamples** value. It is normal if the extra rows are less than four, because a broker publishes snapshot accounting data every 20 seconds and the WebSphere Message Broker Monitoring agent checks and cleans the retained table every 60 seconds.

- The refresh frequency on Tivoli Enterprise Portal

If you stay in one of the accounting workspaces and keep refreshing the workspace frequently, the retained data will not be cleaned up and the agent will return more and more data in the workspace. It has the same effect when you configure the workspace to automatically refresh data after a certain time period by using the menu option on Tivoli Enterprise Portal. Again, if the specified time period is very short, more and more data is returned in the workspace after each refresh.

The **holdTimeForQuery** parameter also determines the maximum interval (in seconds) of refresh frequency. It specifies the length of time that the agent must retain accounting data. If you refresh the workspace more frequently than the interval specified by the **holdTimeForQuery** parameter, all the previously retained data will remain and will not be cleaned up.

Do not enable automatic refresh of accounting workspaces, unless the refresh interval is longer than the **holdTimeForQuery** value. Otherwise, the monitoring agent might require a lot of memory to retain data.

- Historical data collection configuration

The **defaultHistoricalAccountingType** parameter determines which type of accounting data to be collected historically. Do not set this parameter to All or Snapshot unless necessary. This parameter is useful only when the historical data collection has been enabled. If historical data collection is enabled for any of the accounting tables, the accounting data is kept in the agent memory till the data is collected by historical collection. Always set the same collection interval for all the accounting data.

Publish and Subscribe workspaces (WebSphere Message Broker V6 all releases only)

- The ACL Entries workspace presents ACL entries per topic, which define access to publish and subscribe topics. See “ACL Entries workspace” on page 290.
- The Multicast Group Statistics workspace presents current multicast group statistics. See “Multicast Group Statistics workspace” on page 304
- The Multicast Summary Statistics workspace presents current multicast summary statistics. See “Multicast Summary Statistics workspace” on page 304
- The Multicast Topic Statistics workspace presents current multicast topic statistics. See “Multicast Topic Statistics workspace” on page 305
- The Neighbors workspace presents a list of neighbors to the broker in the publish and subscribe topology, and their connection attributes. See “Neighbors workspace” on page 305.

- The Publish-Subscribe Statistics workspace presents current publish and subscribe statistics (excluding multicast data). See “Publish-Subscribe Statistics workspace” on page 307
- The Retained Publications workspace presents a list of topic and subscription points that contain a publication that was retained. See “Retained Publications workspace” on page 307.
- The Subscriptions workspace presents a list of subscriptions to topics. See “Subscriptions workspace” on page 311.

Resource Statistics workspaces (WebSphere Message Broker V7.0 or later only)

Resource Statistics workspaces provide the statistics of system resources that are collected by a broker. The statistical information includes performance and operating details of resources that are used by execution groups. You can use the resource statistics to ensure that your systems are using the available resources in the most efficient manner.

Statistics are collected for the following resource types:

- File

The File Resource Statistics workspace provides the statistical information for the local file system of any file actions that are done by any file node. See “File Resource Statistics workspace” on page 277.
- JDBC connection pools

The JDBC Connection Pools Resource Statistics workspace provides statistical information about each JDBC Provider configurable service. See “JDBC Connection Pools Resource Statistics workspace” on page 277.
- Java virtual machine (JVM)

The JVM Resource Statistics workspace provides the statistical information about the JVM resources that are used by execution groups. See “JVM Resource Statistics workspace” on page 278.
- ODBC

The ODBC Resource Statistics workspace provides the statistical information about each ODBC DSN that has been accessed since the execution group starts. See “ODBC Resource Statistics workspace” on page 283.
- Parsers

The Parsers Resource Statistics workspace provides the statistical information about the parser resources within each execution group. See “Parsers Resource Statistics workspace” on page 284.
- SOAP

The SOAP Input Resource Statistics workspace provides the statistical information about SOAP nodes on a per-operation basis. See “SOAP Input Resource Statistics workspace” on page 289.

Creating a workspace using a predefined workspace as a template

You can create your own workspaces to display information about a specific set of attributes. To create a new workspace using a predefined one as a template, perform the following steps:

1. Open the predefined workspace that you want to use as a template.
2. To create a copy of the predefined workspace, click **File > Save Workspace As**.

3. Enter a workspace name and, optionally, a description. The workspace name is displayed on the title bar.
4. Optional: Select one or more of the following workspace options:
 - **Assign as default for this Navigator Item:** Select this option if you want this workspace to be displayed when this Navigator Item is clicked.
 - **Do not allow modifications:** Select this option to prevent this workspace from being modified in the future.
 - **Only selectable as the target of a workspace link:** Select this option if you do not want this workspace to be displayed unless it is linked to from another workspace.
5. Click **OK**. A copy of the predefined workspace is created with the name that you entered.
6. Open the new workspace and click **Edit > Properties** to customize it to meet your requirements.

Creating the user statistics workspace

In a development environment, sampling statistics at a set interval might not provide the flexibility that you need. You can use the user statistics feature, implemented by three Take Action commands, to collect statistics that are associated with a particular Tivoli Enterprise Portal logon ID. These user statistics are displayed in the following statistics workspaces that they are derived from and can be linked to from related statistics workspaces, as listed in one of the following tables, depending on the version of WebSphere Message Broker.

Table 2. Workspaces with user statistics links (for WebSphere Message Broker V7 or later)

Workspace	User statistics links
Monitor Node Broker Statistics	<ul style="list-style-type: none"> • User Monitor Node Broker Statistics • User Monitor Node Execution Group Statistics
Monitor Node Execution Group Statistics	<ul style="list-style-type: none"> • User Monitor Node Execution Group Statistics • User Monitor Node Message Flow Statistics
Monitor Node Message Flow Statistics	<ul style="list-style-type: none"> • User Monitor Node Message Flow Statistics • User Monitor Node Sub-Flow Statistics • User Monitor Node Base Statistics
Monitor Node Sub-Flow Statistics	<ul style="list-style-type: none"> • User Monitor Node Sub-Flow Statistics • User Monitor Node Base Statistics
Monitor Node Base Statistics	User Monitor Node Base Statistics

Table 3. Workspaces with user statistics links (for WebSphere Message Broker V6 all releases)

Workspace	User statistics links
Broker Statistics	<ul style="list-style-type: none"> • User Broker Statistics • User Execution Group Statistics
Execution Group Statistics	<ul style="list-style-type: none"> • User Execution Group Statistics • User Message Flow Statistics

Table 3. Workspaces with user statistics links (for WebSphere Message Broker V6 all releases) (continued)

Workspace	User statistics links
Message Flow Statistics	<ul style="list-style-type: none"> • User Message Flow Statistics • User Sub-Flow Statistics • User CandleMonitor Node Statistics
Sub-Flow Statistics	<ul style="list-style-type: none"> • User Sub-Flow Statistics • User CandleMonitor Node Statistics
CandleMonitor Node Statistics	User CandleMonitor Node Statistics

To create the user statistics workspace, issue a **WMB Create User Statistics/QI Create User Statistics** command for the user's logon ID before the user statistics workspaces can be accessed (see "WMB Create User Statistics/QI Create User Statistics command" on page 269).

Remember:

- If you try to follow a link to a workspace before the command has been issued for your logon ID, no data is available in the workspace that you link to.
- The user statistics are not collected historically.

Scenarios of monitoring with workspaces

This section provides examples of monitoring situations that are common to many broker product environments, and explains how the WebSphere Message Broker Monitoring agent can be used in these circumstances.

Each scenario presents a case in which you need to monitor some aspect or component of your broker product environment, followed by a description of how to set up WebSphere Message Broker Monitoring agent to gather important information.

Monitoring application message flow performance

This scenario describes how you can use the WebSphere Message Broker Monitoring agent in a performance monitoring situation in which an application message flow is taking excessively long time to process messages.

Do the following steps to enable the WebSphere Message Broker Monitoring agent to gather the appropriate statistics:

1. Place one CandleMonitor node immediately after the MQInput node so that the entry of the messages into the message flow is recorded.
2. Set the **type** attribute of this CandleMonitor node to input.
3. Place another CandleMonitor node immediately in front of the MQOutput node so that the exit of the messages from the message flow is recorded.
4. Set the **type** attribute of this CandleMonitor node to output.
5. Start the WebSphere Message Broker Monitoring agent.

After you have positioned the CandleMonitor nodes and the WebSphere Message Broker Monitoring agent is running, perform the following procedure:

1. Open the Monitor Node Message Flow Statistics/Message Flow Statistics workspace and examine the values of the **Current Average Queue Time** and

- Current Average Flow Time** attributes. For a detailed description of these attributes and their values, see “Monitor Node Message Flow Statistics attributes” on page 174/“Message Flow Statistics attributes” on page 223.
2. If you think that the times reported by the **Current Average Queue Time** attribute are excessively long, increase the Additional Instances value for the monitored message flow. The Additional Instances parameter specifies the number of additional instances of the message flow that the execution group must run so that more messages can be processed concurrently.
 3. If you think that the times reported by the **Current Average Flow Time** attribute are excessively long, then you need to debug the monitored message flow. To perform this task, do the following steps:
 - a. Insert CandleMonitor nodes before each processing node in the message flow. This can help you determine which node is causing the problem.
 - b. Set the **type** attribute of the new CandleMonitor nodes to other.
 - c. Start the WebSphere Message Broker Monitoring agent again.
 - d. Open the Monitor Node Sub-Flow Statistics/Sub-Flow Statistics workspace, and examine the flow times of any subflows in the message flow. For more information about this workspace, see “Monitor Node Sub-Flow Statistics workspace” on page 282/“Sub-Flow Statistics workspace” on page 311.
 - e. Compare statistics for the subflows to determine which section of the message flow is causing problems.

Determining application delivery failure of messages

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to determine the cause of the problem when messages that are destined for an application are not being received by that application.

Perform the following steps to enable the WebSphere Message Broker Monitoring agent to gather the appropriate statistics:

1. Place one CandleMonitor node immediately after the MQInput node so that the entry of the messages into the message flow is recorded.
2. Set the **type** attribute of this CandleMonitor node to input.
3. Place another CandleMonitor node immediately in front of the MQOutput node so that the exit of the messages from the message flow is recorded.
4. Set the **type** attribute of this CandleMonitor node to output.
5. Start the WebSphere Message Broker Monitoring agent.

After you have placed and set the CandleMonitor nodes, and the WebSphere Message Broker Monitoring agent is running, perform the following procedure:

1. Open the Message Flow Status workspace/Message Flow Information workspace and examine the **Status** attribute. The Status column for the monitored message flow must have the value **Started**. If it does not, start the message flow manually.
2. Open the Monitor Node Message Flow Statistics workspace/Message Flow Statistics workspace and compare the **Current Message Input Count** and **Current Message Output Count** attributes. A discrepancy between these values indicates that messages are being lost in the monitored message flow.

For a detailed description of these attributes and their values, see “Monitor Node Message Flow Statistics attributes” on page 174/“Message Flow Statistics attributes” on page 223.

3. Open the Monitor Node Events/Message Flow Events workspace and determine whether any exceptions have been triggered by a node in the monitored message flow.

For information about the workspace, see “Monitor Node Events workspace” on page 280/“Message Flow Events workspace” on page 300.

Debugging a message flow

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to debug the operation of a message flow.

To determine which node in the message flow is causing a problem, do the following steps so that the WebSphere Message Broker Monitoring agent can gather appropriate statistics for you to analyze:

1. Place CandleMonitor nodes on each side of nodes or subflows that you think are functioning incorrectly in the message flow.
2. Set the **type** attribute of a CandleMonitor node placed in front of a node or subflow to subFlowInput.
3. Set the **type** attribute of a CandleMonitor node placed following a node or subflow to subFlowOutput.
4. Set the **subFlowName** attribute of each pair of nodes to the same value, which must be unique within the message flow.
5. Start the WebSphere Message Broker Monitoring agent.

After you have positioned the CandleMonitor nodes and the WebSphere Message Broker Monitoring agent is running, do the following steps:

1. Open the Monitor Node Sub-Flow Statistics workspace/Sub-Flow Statistics workspace, and examine the flow times of all subflows in the message flow.
For more information about this workspace, see “Monitor Node Sub-Flow Statistics workspace” on page 282/“Sub-Flow Statistics workspace” on page 311.
2. Compare statistics for different subflows to determine which section of the message flow is problematic.

Verifying the broker configuration

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to verify that the broker configuration matches your configuration as depicted in the Message Brokers Toolkit.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that the necessary statistics can be gathered, and then perform the following steps:

1. Examine the various informational workspaces, such as Execution Group Status/Execution Group Information, Message Flow Status/Message Flow Information, Message Processing Nodes/Message Processing Node Information, Neighbors, Subscriptions, Retained Publications, and ACL Entries. These workspaces contain data from the broker perspective as opposed to the Message Brokers Toolkit perspective.
2. Compare the data in the workspaces with what you expect to determine whether the correct configuration is deployed to the broker.

Planning broker capacity

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to plan the capacity for your brokers.

To determine the most recent message that was sent between two applications, do the following steps so that the WebSphere Message Broker Monitoring agent can gather the necessary statistics:

1. Place one CandleMonitor node immediately following the MQInput node so that the entry of the message into the message flow is recorded.
2. Set the **type** attribute of the CandleMonitor node to input.
3. Place one CandleMonitor node immediately in front of the MQOutput node so that the exit of the message from the message flow is recorded.
4. Set the **type** attribute of the CandleMonitor node to output.
5. Start the WebSphere Message Broker Monitoring agent again.

After you have positioned the CandleMonitor nodes and the WebSphere Message Broker Monitoring agent is running, do the following steps:

1. Open statistics workspaces, such as Monitor Node Message Flow Statistics/Message Flow Statistics for individual message flows, Monitor Node Execution Group Statistics/Execution Group Statistics for summarization over an entire execution group, or Monitor Node Broker Statistics/Broker Statistics for statistics summarizing the operation of an entire broker.
2. Examine the statistics workspaces and their historical workspaces (which you can link to from the statistics workspaces).
3. If you find that for a particular message flow the **Current Message Input Rate** value is higher than expected, or that the **Current Average Queue Time** value is unacceptably long, add more instances to the flow in the Message Brokers Toolkit.
4. If you find that for a particular execution group the **Current Message Input Rate** value is higher than expected, or that the **Current Average Queue Time** value is unacceptably long, add a new execution group and assign instances as appropriate.
5. If you find that for a particular broker the **Current Message Input Rate** value is higher than expected, or that the **Current Average Queue Time** value is unacceptably long, consider adding a new broker with a configuration that can offload some processing from the current broker.

Ensuring reasonable message flow response times

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to ensure that the response times of your message flows are reasonable.

To monitor message flow response times, perform the following steps so that the WebSphere Message Broker Monitoring agent can gather the necessary statistics:

1. Place one CandleMonitor node immediately following the MQInput node so that the entry of the message into the message flow is recorded.
2. Set the **type** attribute of the CandleMonitor node to input.
3. Place one CandleMonitor node immediately in front of the MQOutput node so that the exit of the message from the message flow is recorded.
4. Set the **type** attribute of the CandleMonitor node to output.
5. Start the WebSphere Message Broker Monitoring agent again.

After you have positioned the CandleMonitor nodes, and the WebSphere Message Broker Monitoring agent is running, do the following steps:

1. Open the Monitor Node Message Flow Statistics/Message Flow Statistics workspace for individual message flows. For a detailed description of these attributes and their values, see “Monitor Node Message Flow Statistics attributes” on page 174/“Message Flow Statistics attributes” on page 223.
2. Examine the workspaces and their historical equivalents (you can link to the historical information from the statistics workspaces).
3. If the **Current Average Flow Time** for a particular message flow is higher than expected, debug the message flow as described in “Debugging a message flow” on page 84. Compare the Average Flow Times for each section of the message flow to identify where problems are occurring.

Verifying subscriptions without accessing the Message Brokers Toolkit

This scenario describes how you can use the WebSphere Message Broker Monitoring agent to verify your subscriptions when you do not have access to the Message Brokers Toolkit.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that the appropriate information can be gathered, and then perform the following steps:

1. Open the Subscriptions workspace for each broker from which you expect the application to receive publications.
2. If your subscription is displayed in a single broker, examine the report data to determine if you have specified the **Local** option for the subscription; look in the **Local Only** column for the value **Yes**. This setting prevents the subscription from being propagated to neighboring brokers.

Determining if an application has subscription access to a topic

This scenario describes how you can use the WebSphere Message Broker Monitoring agent when your application attempts to subscribe to a topic, and fails for security reasons.

After you have configured the WebSphere Message Broker Monitoring agent, start the agent so that necessary statistics can be gathered, and then do the following step:

- Open the ACL Entries workspace to determine if the application user ID matches a principal that allows subscription access to the topic or if its user ID is a member of a group that matches a principal that allows subscription access to the topic.

Displaying a graphical view of your environment

You can use the topology workspaces to get a graphical view of your broker environment. For different versions of brokers, the ITCAM agent for WebSphere Message Broker provides different sets of workspaces to present the topology overview with various tiers.

- For brokers of version 7.0 or later:
 - Broker Status (broker topology, execution group topology, and message flow topology)
 - Message Flow Node Topology (message flow topology)

- For brokers of earlier versions:
 - Broker Information
 - Message Flow Topology
 - Execution Group Topology

Viewing the topology of your broker environment

The Broker Status (for WebSphere Message Broker V7 or later) or Broker Information (for WebSphere Message Broker V6 all releases) workspace provides a graphical overview of your broker environment, including execution groups and messages flows.

Figure 10 illustrates an example of the broker environment topology displayed in the Broker Status workspace (for WebSphere Message Broker V7 or later). In the broker topology view of the Broker Status workspace, you can click a message flow node to link to the Message Flow Node Topology workspace for more information about the message flow.

From the broker topology view of the Broker Information workspace (for WebSphere Message Broker V6 all releases), you can right-click any object in this topology view to link to a workspace containing more information about the item.

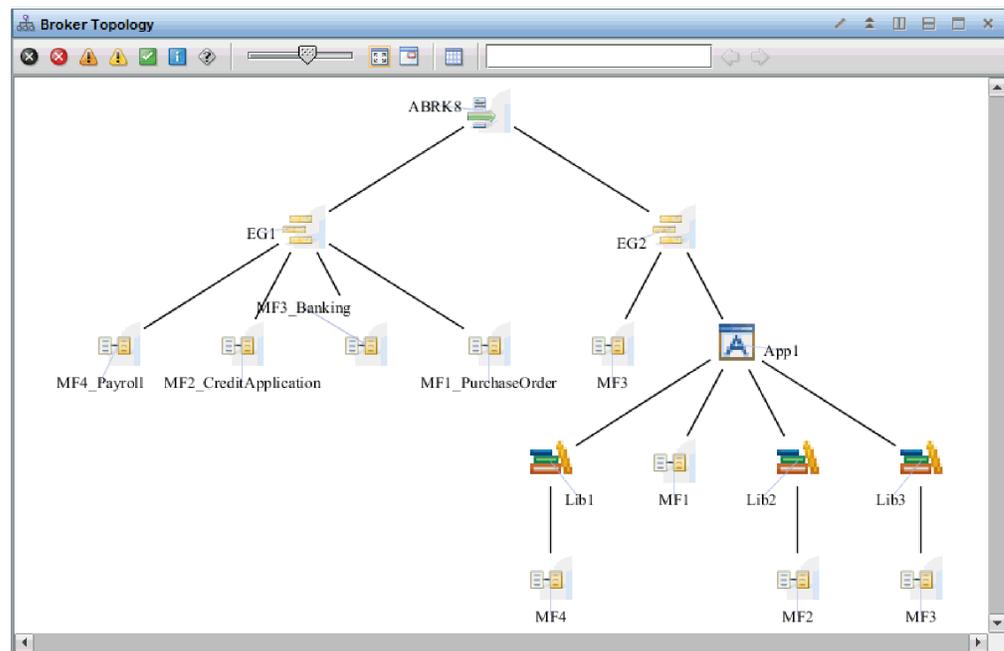


Figure 10. The broker topology view

The broker topology view has three to five tiers. Each tier contains icons representing different objects in the broker environment, represented by standard WebSphere Message Broker icons. The different tiers are as follows:

- **Tier 1:** The message broker. This tier contains a single icon representing the monitored message broker.
- **Tier 2:** Execution groups. This tier presents all the execution groups that exist within the broker.
- **Tier 3:** Applications/Libraries/Message flows. This tier presents all the applications/libraries/message flows that are contained within each execution group.

- **Tier 4:** Libraries/Message flows. This tier presents all the libraries or message flows that are contained within each application or library.
- **Tier 5:** Message flows. This tier presents all the message flows that are contained within each library.

If either the broker or a message flow is stopped, a red X is displayed in the upper-left corner of its icon. A red X is displayed in the upper-left corner of the icon representing an execution group if all of its user-defined message flows are stopped. If some but not all of the user-defined message flows in an execution group are started, a yellow exclamation mark (!) is displayed in the upper left corner of the icon representing the execution group. Figure 11 presents the message broker icons for a started and a stopped message broker.

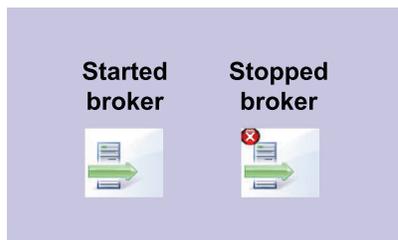


Figure 11. Started and stopped broker icons in the broker topology view

Exception: No default system message flows is displayed in the broker topology view.

Viewing the topology of a message flow

You can get a graphical overview of a single message flow, including its nodes and the links between them, in the following ways:

- WebSphere Message Broker V7 or later:
 - In the Message Flow Status table of the Broker Status workspace, right-click a row and click **Link To > Message Flow in Broker Status** or **Link To > Message Flow Node Topology**.
 - In the Broker Topology view of the Broker Status workspace, right-click a message flow node and click **Link To > Message Flow Node Topology**
- WebSphere Message Broker V6 (all releases):
 - In the Broker Topology view of the Broker Information workspace, right-click the message flow that you are interested in, and then click **Link To > Message Flow Topology**.
 - In the Message Flow Information workspace, right-click a row in the table view, and then click **Link To > Message Flow Topology**.

Figure 12 on page 89 illustrates an example of the message flow topology view displayed in the Message Flow Node Topology workspace for WebSphere Message Broker V7 or later.

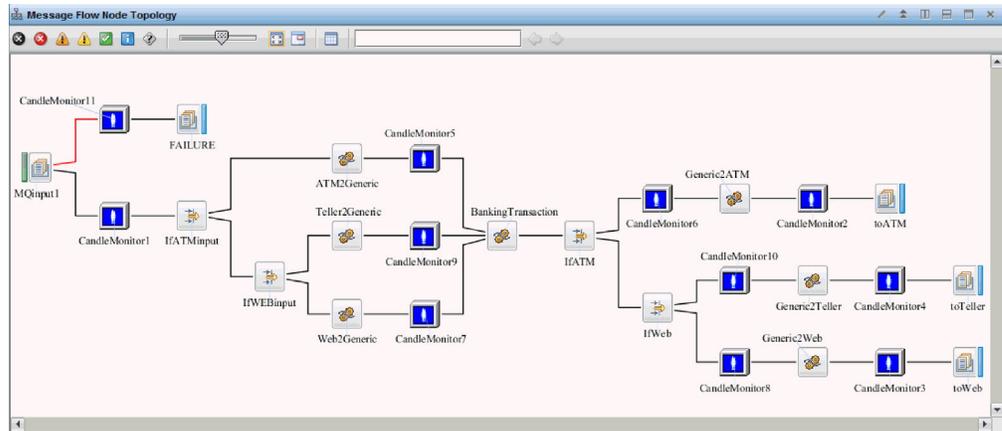


Figure 12. The message flow topology view

The message flow topology view presents each node in the message flow and the links between them. No matter how many terminals on two nodes are connected, there is only one link between the two nodes in the message flow topology view. The color of each link indicates its function. The icons and links used to represent the topology are the same in the WebSphere Message Brokers Toolkit. For more information see WebSphere Message Brokers Toolkit documentation available from <http://www.ibm.com>.

The only icons that are not the same as WebSphere Message Brokers Toolkit icons are the icons that represent the CandleMonitor node and user-defined nodes. These are shown in Table 4.

Table 4. A selection of message flow topology nodes

Icon	Node	Notes
	CandleMonitor node	
	User-defined node	All user-defined nodes in the message flow are denoted by this icon, regardless of function.

Because the information included in the message flow topology view is collected from the WebSphere Message Broker Monitoring agent, it reflects the objects that are deployed in the Message Broker environment. Objects that exist as part of a message flow design in the WebSphere Message Brokers Toolkit, but are not deployed to the environment, cannot be displayed. For this reason, subflows are displayed in the topology view as part of the main message flow, and are not displayed separately as they are in the Message Brokers Toolkit.

Viewing the topology of an execution group

You can get a graphical overview of an execution group, including its message flows in the following ways:

- WebSphere Message Broker V7 or later:
In the Execution Group Status table of the Broker Status workspace, right-click a row and click **Link To > Execution Group in Broker Status**.

- WebSphere Message Broker V6 (all releases):
 - In the Broker Topology view of the Broker Information workspace, right-click the execution group that you are interested in, and then click **Link To > Execution Group Topology**.
 - In the Execution Group Information workspace, right-click a row in the table view, and then click **Link To > Execution Group Topology**.

Figure 13 illustrates an example of the execution group topology view displayed in the Broker Status workspace for WebSphere Message Broker V7 or later.

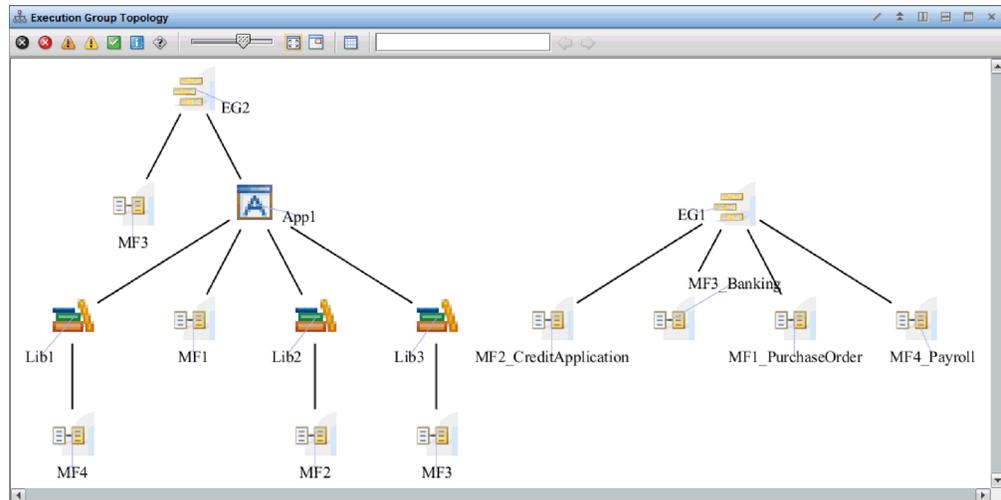


Figure 13. The execution group topology view

If a message flow is stopped, a red X is displayed in the upper left corner of its icon. A red X is displayed in the upper left corner of the icon representing the execution group if all of its user-defined message flows are stopped. If some but not all of the user-defined message flows in an execution group are started, a yellow exclamation mark (!) is displayed in the upper left corner of the icon representing the execution group.

Exception: No default system message flows is displayed in the Execution Group Topology view.

Chapter 6. Collecting historical data

Tivoli Enterprise Portal provides options for configuring the collection and storage of historical data from the WebSphere Message Broker Monitoring agent. You can use these options to specify settings that are related to historical data collection, including the attributes for which historical data is collected, the collection interval, warehousing interval, the length of time that historical data must be stored and what reports are generated from the data.

Important: To view certain historical workspaces within Tivoli Enterprise Portal, you must configure historical data collection for the attribute groups that contain attributes that are displayed in those workspaces.

Initial settings for historical collection

The following historical tables are disabled by default after installation. To enable them, you must first enable historical data collection for the indicated attribute groups. Each attribute group corresponds to a product workspace.

- WebSphere Message Broker V7.0 or later:
 - If a Broker Status Events row occurs, an entry is added to the historical log.
 - If a Monitor Node Events row occurs, an entry is added to the historical log.
 - If a Product Events row occurs, an entry is added to the historical log. This situation applies to agent-managed systems.
 - A Broker Status log entry is written every 15 minutes (one row per broker).
 - A Components log entry is written every 15 minutes (one row per broker product component that was created on the system). This situation applies to agent-managed systems.
 - An Execution Group Status log entry is written every 15 minutes (one row per execution group).
 - A Message Flow Status log entry is written every 15 minutes (one row per message flow).
 - A File Resource Statistics log entry is written every 20 seconds (one row per execution group).
 - A JDBC Connection Pools Resource Statistics log entry is written every 20 seconds (one row per execution group).
 - A JVM Resource Statistics log entry is written every 20 seconds (four rows per execution group).
 - A ODBC Resource Statistics log entry is written every 20 seconds (one row per execution group).
 - A Parsers Resource Statistics log entry is written every 20 seconds (three rows or more per execution group).
 - A SOAPInput Resource Statistics log entry is written every 20 seconds (one row per execution group).
- WebSphere Message Broker V6 (all releases):
 - If a Broker Events row occurs, an entry is added to the historical log.
 - If a Message Flow Events row occurs, an entry is added to the historical log.
 - If a Product Events row occurs, an entry is added to the historical log. This situation applies to agent-managed systems.

- An ACL Entries log entry is written every 15 minutes (one row ACL entry).
- A Broker Information log entry is written every 15 minutes (one row per broker).
- A Components log entry is written every 15 minutes (one row per broker product component that was created on the system). This situation applies to agent-managed systems.
- An Execution Group Information log entry is written every 15 minutes (one row per execution group).
- A Message Flow Information log entry is written every 15 minutes (one row per message flow).
- A Message Processing Node Information log entry is written every 15 minutes (one row per message processing node).
- A Neighbors log entry is written every 15 minutes (one row per neighbor connection).
- A Subscriptions log entry is written every 15 minutes (one row per subscription).
- A Retained Publications log entry is written every 15 minutes (one row per retained publication).

Remember: The following historical tables are pure event tables. Historical data collection of the pure event tables is not affected by the collection interval that you set in the Historical Collection Configuration window on Tivoli Enterprise Portal.

- WebSphere Message Broker V7.0 or later:
 - Broker Status Events
 - Monitor Node Events
 - Product Events
 - File Resource Statistics
 - JDBC Connection Pools Resource Statistics
 - JVM Resource Statistics
 - ODBC Resource Statistics
 - Parsers Resource Statistics
 - SOAPInput Resource Statistics
- WebSphere Message Broker V6 (all releases):
 - Broker Events
 - Message Flow Events
 - Product Events

The following tables listed must always be configured with the same historical data collection interval for accurate results. The configured historical data collection interval must also be a multiple of the value of the **defaultStatisticInterval** (or **statisticInterval**) agent parameter; otherwise, historical data collection might produce unpredictable results. For example, if you configure historical data collection for these statistics attribute groups to run every 5 minutes (300 seconds), the **defaultStatisticInterval** (or **statisticInterval**) parameter value must be a submultiple of 300, such as 60 (1 minute), 75 (1.25 minutes), 100 (1.67 minutes), 150 (2.5 minutes), or 300 (5 minutes).

- WebSphere Message Broker V7.0 or later:
 - A Monitor Node Broker Statistics log entry is written every 15 minutes (one row per broker).

- | – A Monitor Node Execution Group Statistics log entry is written every 15
- | minutes (one row per execution group).
- | – A Monitor Node Message Flow Statistics log entry is written every 15 minutes
- | (one row per message flow per execution group).
- | – A Monitor Node Sub-Flow Statistics log entry is written every 15 minutes
- | (one row per monitored subflow per message flow containing the subflow).
- | – A Monitor Node Base Statistics log entry is written every 15 minutes (one
- | row per CandleMonitor Node).
- WebSphere Message Broker V6 (all releases):
 - A Broker Statistics log entry is written every 15 minutes (one row per broker).
 - An Execution Group Statistics log entry is written every 15 minutes (one row
 - per execution group).
 - A Message Flow Statistics log entry is written every 15 minutes (one row per
 - message flow per execution group).
 - A Sub-Flow Statistics log entry is written every 15 minutes (one row per
 - monitored subflow per message flow containing the subflow).
 - A CandleMonitor Node Statistics log entry is written every 15 minutes (one
 - row per CandleMonitor Node).

The following tables gather data only if data is produced. By default, the agent collects historical data only for use as archive data, not snapshot. See “defaultHistoricalAccountingType” on page 21 and “historicalAccountingType” on page 30 for more information.

- | • WebSphere Message Broker V7.0 or later:
 - | – An Archive Accounting Message Flow Statistics log entry is written for each
 - | message flow archive statistics report that is produced by the broker.
 - | – An Archive Accounting Thread Statistics log entry is written for each thread
 - | archive statistics report that is produced by the broker.
 - | – An Archive Accounting Node Statistics log entry is written for each node
 - | archive statistics report that is produced by the broker.
 - | – An Archive Accounting Terminal Statistics log entry is written for each
 - | terminal archive statistics report that is produced by the broker.
- WebSphere Message Broker V6 (all releases):
 - A Message Flow Accounting log entry is written for each message flow
 - accounting and statistics report that is produced by the broker.
 - A Thread Accounting log entry is written for each thread accounting and
 - statistics report that is produced by the broker.
 - A Node Accounting log entry is written for each node accounting and
 - statistics report that is produced by the broker.
 - A Terminal Accounting log entry is written for each terminal accounting and
 - statistics report that is produced by the broker.
 - A Publish-Subscribe Statistics log entry is written for each publish/subscribe
 - statistics report that is produced by the broker.
 - A Multicast Summary Statistics log entry is written for each multicast
 - summary statistics report that is produced by the broker.
 - A Multicast Group Statistics log entry is written for each multicast group
 - statistics report that is produced by the broker.
 - A Multicast Topic Statistics log entry is written for each multicast topic
 - statistics report that is produced by the broker.

Remember:

- For WebSphere Message Broker V7.0 or later, historical data can be collected only for archive accounting data, not snapshot accounting data, although the snapshot data can be displayed correctly in the historical workspaces. Do not collect historical snapshot data, otherwise, the data amount might be enormous.
- The collection interval of archive accounting data is determined by the broker with the `mqsichangebroker -v` command. If the collection interval of the archive accounting data is longer than that of the historical data, the historical versions of these archive accounting workspaces might be empty during one historical collection interval. This is because during this historical collection interval, no data is produced by the broker.

Starting historical data collection

Your user ID must have Configure History permission to open the History Collection Configuration window. If you do not have the permission, you will not see the menu item or tool for historical configuration.

To start collecting historical data for an attribute group on one or more managed systems that are monitored by a WebSphere Message Broker Monitoring agent, do the following steps:

1. To open the History Collection Configuration window, click **Edit > History Configuration** from the Tivoli Enterprise Portal.
2. Click **WebSphere Message Broker** in the left side of the window.
3. Click **Create new collection setting**. The **Create New Collection Settings** window is opened.
4. In the **Name** field, enter a name of up to 256 characters.
5. Optional: In the **Description** field, enter a description of up to 64 characters for the collection.
6. Select an attribute group from the **Attribute Group** list. Only attribute groups that are appropriate for historical collection and reporting are displayed in the list.
7. Click **OK**. The configuration tabs for the collection are displayed.
8. Complete the fields in the **Basic** tab:
 - **Collection Interval** is the frequency of data transmission to the short-term history file on the computer where the data is saved (Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server). The options are every one, five, 15, or 30 minutes, every hour, or once per day. The default interval is 15 minutes. The shorter the interval is, the faster and larger the history file grows. A short interval should be used only for an attribute group that is critical in your work.
 - **Collection Location** is where the short-term history data file resides: at the TEMA (Tivoli Enterprise Monitoring Agent) or the TEMS (Tivoli Enterprise Monitoring Server). The default location is TEMA, which minimizes the performance impact on the Tivoli Enterprise Monitoring Server from historical data management.
 - **Warehouse Interval** determines whether the collected data is warehoused and how often. The options are 15 minutes, 30 minutes, 1 hour, 12 hours, 1 day or Off.
9. In the **Distribution** tab page, select managed systems for which you want to start historical data collection from the **Available Systems** list and click the left arrow to move them to the **Start collection on** list.

Remember: In the available managed system group list, MQSI_BROKER_V7 is the group of all managed systems for WebSphere Message Broker V7.0 or later. MQSI_BROKER is the group of managed systems for WebSphere Message Broker V6 (all releases).

10. Click **OK** to start data collection on the managed systems.

Stopping historical data collection

Your user ID must have Configure History permission to open the History Collection Configuration window. If you do not have this permission, you will not see the tool for historical configuration.

You can edit historical collection for an attribute group to stop data collection on some or all of the managed systems in the distribution list. The managed systems that remain in the distribution list continue to collect historical data. To stop historical data collection on one or more managed systems that are monitored by a WebSphere Message Broker Monitoring agent, do the following steps:

1. To open the History Collection Configuration window, click **Edit > History Configuration** in the Tivoli Enterprise Portal.
2. Click the plus sign (+) to expand the **WebSphere Message Broker** branch on the left side of the window.
3. Click the collection that the managed systems for which you want to stop historical data collection are assigned to.
4. Click the **Distribution** tab.
5. In the **Start collection on** list, select the managed systems that you want to stop historical data collection and click the right arrow to move it to the **Available Systems** or **Available Managed System Group** list.

Remember: MQSI_BROKER_V7 is the group of all managed systems for WebSphere Message Broker V7.0 or later. MQSI_BROKER is the group of managed systems for WebSphere Message Broker V6 (all releases).

6. To save your changes, click **Apply** to keep the window open or **OK** to close it.

Historical data collection no longer occurs on the managed systems that you move out of the **Start collection on** list for the historical collection definition.

Viewing historical data for a selected time frame

In historical workspaces, you can choose to display only historical data that is collected over a particular period of time in which you are interested. To do this, do the following steps:

1. Navigate to the historical workspace for which you want to view data from a particular period of time.
2. Click the **Specify Time Span for Query** button, located in the top left corner of each view in the historical workspace. The Select the Time Span window is displayed.
3. Select what data you want to be displayed in the table. There are the following available options:
 - **Real time**
If you select this option, only the data collected during the most recent sampling period is displayed in the table.
 - **Last**

If you select this option, you can choose to display all historical data going back to a certain date and time. For example, all data collected over the past 24 hours.

- **Custom**

If you select this option you can specify the exact period for which you want historical data to be displayed.

4. Optional: Do one of the following steps, depending on which option you have selected in the previous step.

- If you selected the **Last** option, enter the time period for which you want data to be displayed in the field provided, and select the units in which it is specified (for example, hours or days). You can also specify the following parameters:

- **Use Detailed data**

If you select this option, the data from the detailed data tables is displayed in the table without summarization. You can also select the column that you want to be used in determining whether data falls within the selected period from the Time column list of columns containing timestamps.

- **Use summarized data**

If you select this option, the data from the summarized data tables is displayed in the table. This data is aggregated by the time frame configured in the Historical Collection Configuration window. If you configured shift times when you installed IBM Tivoli Monitoring, you can also select for which shifts and days data is displayed. See your IBM Tivoli Monitoring documentation for further information.

- If you selected the **Custom** option, you can specify the following parameters:

- **Use Detailed data**

If you select this option, the data from the detailed data tables is displayed in the table without summarization. You can also select the column that you want to be used in determining whether data falls within the selected period from the Time column list of columns containing timestamps.

- **Use summarized data**

If you select this option, the data from the summarized data tables is displayed in the table. From the Interval list, select the time period over which you want the data to be aggregated. If you configured shift times when you installed IBM Tivoli Monitoring, you can also select for which shifts and days data is displayed. See your IBM Tivoli Monitoring documentation for further information.

- In the **Start time and End time** fields, select the time period for which you want data to be displayed.

Exception: Summarization is not performed on the attributes of string, timestamp, and enumeration types. It is meaningless to view summarized data in some workspaces such as Broker Status Event.

5. Optional: To apply the time span to all other views in this workspace that use the same query, select the **Apply to all views associated with this view's query** check box.
6. To exit the Select the Time Span window, click **OK**. The workspace is refreshed to reflect the time span you selected.

For more information about this function, see your IBM Tivoli Monitoring documentation.

Offline collection of historical data

From version 7 of the WebSphere Message Broker Monitoring agent, historical data is collected for some workspaces even when the agent is stopped, provided that the **persistentBrokerData** option is YES. Because the reply queue used by the agent is persistent, message flow accounting, publish-subscribe statistics, and broker event data is still delivered to the queue when the agent is stopped. After the agent is started again, stored data is processed and can be viewed in historical information workspaces. Historical information for the following workspaces is collected when the agent is stopped:

- WebSphere Message Broker V7.0 or later:
 - Archive Accounting Message Flow Statistics
 - Archive Accounting Thread Statistics
 - Archive Accounting Node Statistics
 - Archive Accounting Terminal Statistics
 - Broker Status Events
 - Resource Statistics
- WebSphere Message Broker V6 (all releases):
 - Archive/Snapshot Message Flow Accounting
 - Archive/Snapshot Thread Accounting
 - Archive/Snapshot Node Accounting
 - Archive/Snapshot Terminal Accounting
 - Publish-Subscribe Statistics
 - Multicast Summary Statistics
 - Multicast Group Statistics
 - Multicast Topic Statistics
 - Broker Events

Chapter 7. Running reports with Tivoli Common Reporting

Tivoli Common Reporting is a reporting feature available to users of Tivoli products, and provides a consistent approach to viewing and administering reports. Cognos data model and sample reports for WebSphere Message Broker Monitoring agent are provided in a report package for use with Tivoli Common Reporting 2.1.1. The report package is a set of historical reports for both raw and summarized data that is collected in Tivoli Data Warehouse. You can use the sample reports to create you own reports on the data model.

Prerequisites

Before you can run reports with Tivoli Common Reports, make sure that the following requirements are met in your environment.

- The WebSphere Message Broker Monitoring agent is installed and running.
- The Warehouse Proxy Agent is started.
- The Summarization and Pruning Agent is started.
- Historical data collection is enabled for the related attribute group.
- Summarized tables and views are created and populated in the Tivoli Data Warehouse.

Tips:

- To check whether the required summarized tables and views have been created, run the following queries against Tivoli Data Warehouse. If the tables and views have been correctly created, you can see the result sets that contain HV, DV, WV, QV, and YV in each base table, for example, Broker_Statistics, Broker_Statistics_HV, Broker_Statistics_DV, Broker_Statistics_WV, Broker_Statistics_MV, Broker_Statistics_QV, and Broker_Statistics_YV.
 - DB2

```
select distinct "TABNAME" from SYSCAT.TABLES where
  "TABSHEMA" = 'ITMUSER'
```
 - Oracle

```
select distinct "TABLE_NAME" from USER_TABLES
```
 - SQL Server

```
select TABLE_NAME "VIEWNAME" from INFORMATION_SCHEMA.TABLES
```
- Cognos reports can be run against yearly, quarterly, monthly, weekly, daily, and hourly summarization intervals. You can decide which summarization interval is important to you to run reports against, and enable summarization for the related attribute group.
 - WebSphere Message Broker V7.0 or later:
 - Accounting Message Flow Statistics
 - Accounting Node Statistics
 - Accounting Terminal Statistics
 - Accounting Thread Statistics
 - Broker Status
 - Broker Status Events
 - Components
 - Execution Group Status

- | - File Resource Statistics
- | - JDBC Connection Pools Resource Statistics
- | - JVM Resource Statistics
- | - Message Flow Status
- | - Monitor Node Base Statistics
- | - Monitor Node Broker Statistics
- | - Monitor Node Events
- | - Monitor Node Execution Group Statistics
- | - Monitor Node Message Flow Statistics
- | - Monitor Node Sub-Flow Statistics
- | - ODBC Resource Statistics
- | - Parsers Resource Statistics
- | - Product Events
- | - SOAPInput Resource Statistics
- | - WebSphere Message Broker V6 (all releases):
 - | - ACL Entries
 - | - Broker Events
 - | - Broker Information
 - | - Broker Statistics
 - | - CandleMonitor Node Statistics
 - | - Components
 - | - Execution Group Information
 - | - Execution Group Statistics
 - | - Message Flow Accounting
 - | - Message Flow Events
 - | - Message Flow Information
 - | - Message Flow Statistics
 - | - Message Processing Node Information
 - | - Multicast Group Statistics
 - | - Multicast Summary Statistics
 - | - Multicast Topic Statistics
 - | - Neighbors
 - | - Node Accounting
 - | - Product Events
 - | - Publish-Subscribe Statistics
 - | - Retained Publications
 - | - Sub-Flow Statistics
 - | - Subscriptions
 - | - Terminal Accounting
 - | - Thread Accounting

Installing Cognos reports for WebSphere Message Broker Monitoring agent

Installing Cognos reports for WebSphere Message Broker Monitoring agent includes installing agent-specific reports, configuring Cognos data source, and creating Tivoli Reporting and Analytics Model (TRAM) dimensions. The Cognos reports for WebSphere Message Broker Monitoring agent must be installed on Tivoli Common Reporting server.

A report package for the WebSphere Message Broker Monitoring agent is provided in a folder named REPORTS on the installation disk. Use one of the following folders within the REPORTS folder to install Cognos reports depending on the database type of Tivoli Data Warehouse:

- DB2 and SQL Server: ITCAM_Agents_for_WebSphere_Messaging_v71
 - Oracle: ITCAM_Agents_for_WebSphere_Messaging_v71_for_Oracle
1. If the report package folder is on a remote system, do one of the steps depending on the operating system where the Tivoli Common Reporting server is installed:
 - Windows: Map the remote folder to a local drive.
 - Systems other than Windows: Mount the remote system.
 2. In the report package folder, do one of the following steps to start the installation program
 - Windows: Double-click the setup_windows.exe file.
 - Systems other than Windows: Run the setup_<platform>.bin file.
 3. Select the language of the installation program and click **OK**.
 4. In the Welcome window, click **Next**.
 5. In the Choose the Installation Folder window, specify the path to the Tivoli Common Reporting component directory and click **Next**.

Important: After this step, you cannot step back to this window to change the installation folder again. If you want to change the installation folder afterwards, cancel the installation program and start it again from Step 2.

6. In the Choose the reports for the installation window, select **WebSphere Message Broker Monitoring Agent Reports** and click **Next**.
7. In the Cognos Engine Configuration window, type the Tivoli Common Reporting user name and password, and then click **Next**.
8. In the Cognos Data Source TDW Configuration window, provide all required database information to define the Cognos data source, and click **Next**.

Remember: If you choose to skip this step, you must configure the connection to a database to access your data after the installation is complete. For instructions about how to configure a data connection, see Tivoli Common Reporting documentation.

9. In the Data Script runDbScript Configuration window, provide all required database information to create TRAM dimensions and click **Next**.

Important: The user ID that is used to run the database script must have administrator access to the IBM_TRAM schema. The script requires the administrator access to delete the IBM_TRAM schema related objects and re-create them.

Remember: Do not skip this step, otherwise, you must manually add TRAM dimensions, which are required for running Cognos reports and using the data model.

10. Review the installation information and click **Install**.
11. After the installation is complete, click **Done** to exit.

Working with reports

Topics in this section provide instructions about how to run Cognos reports. If you did not use the installation wizard to install Cognos reports (see “Installing Cognos reports for WebSphere Message Broker Monitoring agent” on page 101), you must do the following steps before you can run Cognos reports:

1. Save the report package for WebSphere Message Broker Monitoring agent to the `TCR_component_dir\cognos\deployment` directory.
2. Import the WebSphere Message Broker Monitoring agent report package to Tivoli Common Reporting.
3. Create the Cognos data source for your report.

Tips:

- The report package for WebSphere Message Broker Monitoring agent is provided as a compressed file named `WebSphereBrokerReports.zip` on the installation disk in the following folders:
 - For DB2 and SQL Server: `REPORTS\ITCAM_Agents_for_WebSphere_Messaging_v71\reports\cognos_reports\Messaging\packages\WebSphereBrokerReports.zip`
 - For Oracle: `REPORTS\ITCAM_Agents_for_WebSphere_Messaging_v71_for_Oracle\reports\cognos_reports\Messaging\packages\WebSphereBrokerReports_Oracle.zip`
- For more information about how to work with reports using IBM Cognos 8 Business Intelligence Reporting, see the IBM Tivoli Common Reporting Information Center.

Creating or editing Web-based reports

You can create and customize your own reports using the Cognos Report Studio, which is a web-based report editor.

1. Log on to the Tivoli Common Reporting interface, and go to **Common Reporting**.
2. In the Work with reports window on the right, select **Report Studio** from the **Launch** list.
3. Use the menu controls to create a report or edit existing reports by formatting the layout and manipulating the data that appears in the report.
4. Save your report, and run it anytime you want to present on its underlying data.

Creating ad hoc reports

After you import the report package provided for WebSphere Message Broker Monitoring agent, you can create ad hoc reports by using simple queries and formatting.

To create ad hoc reports, perform the following steps:

1. Log on to the Tivoli Common Reporting interface, and go to **Common Reporting**.
2. In the Work with reports window on the right, select **Query Studio** from the **Launch** list.
3. Select the WebSphere Message Broker Monitoring Agent package. A new window is open and you can create a report.
4. From the navigation on the left, drag the data items that you want to include in your report.

Remember: Always drag the attributes that belong to the same query to your report. If you drag two attributes from different queries, for example, one from Broker Statistics Daily and the other from Node Accounting Daily, the resulting data might be wrong. However, you can drag any time-related attributes from the Time identifier to your report.

5. Once you finish editing the report data and appearance, save the report by specifying a name, and optionally a description, and a screen tip.

Tip: When you do an ad hoc query, you can either see the live data as you drag the items, or switch the mode to show placeholders for the data and then run the report.

Some other things that you can do in a report

- Combining metrics in one table

You can combine different metrics into one table. For example, you can drag Trace Level from the Message Flow Status table, and Avg Elapsed Microseconds from the Accounting Message Flow Statistics table to the same table.

- Grouping data

You can group the data by clicking the Type column and then the Group icon



- Creating a chart

To create a chart, click the Chart icon . A chart is created for each section with appropriate groups.

For more information about using Query Studio, see Query Studio User Guide at the IBM Cognos 8 v4 Business Intelligence Information Center.

Data model of WebSphere Message Broker Monitoring agent

The data model for WebSphere Message Broker Monitoring agent is a star schema with dimensions or identifiers that are separated from facts or metrics. Metrics are measurable (numeric) attributes, which can be aggregated by identifiers. The relationship among the metric tables is defined using two common identifiers, resources and time.

The WebSphere Message Broker Monitoring agent collects various metrics. All the metrics are modeled for the agent. The metrics are classified into two categories, key metrics and extended metrics.

Key metrics

The key metrics are the most important or the most frequently used metrics. The key metrics are divided into three groups, resource usage, performance, and

availability. You can either see the raw metrics or summarized (daily and hourly) metrics.

Extended metrics

In addition to the key metrics, the extended metrics include all the other metrics of the WebSphere Message Broker Monitoring agent.

Identifiers

Identifiers are used to link metrics data across different agents. The two primary identifiers that are used by the WebSphere Message Broker Monitoring agent are resource and time.

Resource identifiers include execution group and host name. Time identifiers include various attributes of time, by which the metrics can be grouped, such as standard timestamp, date, minute, hour, day, week of year, month, quarter, or year.

Attributes

Attributes describe the identifiers. For example, the detailed information about a channel can be described by the attributes, such as channel type, query type, or command level.

Sample reports

Eight sample reports are provided in the report package for WebSphere Message Broker Monitoring agent. For the agent to get data to display in the sample reports, historical data collection must be enabled for the related attribute groups.

Table 5. Historical attribute groups for sample reports

Sample report	Attribute group
"Broker Daily Availability report" on page 105	<ul style="list-style-type: none"> WebSphere Message Broker V7.0 or later: Broker Status (detailed data) WebSphere Message Broker V6 (all releases): Broker Information (detailed data)
"Broker Weekly Availability report" on page 109	
"Broker Execution Group Daily Availability report" on page 105	WebSphere Message Broker V7.0 or later only: Execution Group Status (detailed data)
"Broker Execution Group Weekly Availability report" on page 105	
"Broker Message Flow Daily Availability report" on page 106	WebSphere Message Broker V7.0 or later only: Message Flow Status (detailed data)
"Broker Message Flow Weekly Availability report" on page 107	
"Broker Top n Elapsed Microseconds report" on page 108	<ul style="list-style-type: none"> WebSphere Message Broker V7.0 or later: Accounting Message Flow Statistics (summarized data) WebSphere Message Broker V6 (all releases): Message Flow Accounting (summarized data)
"Broker Message Flow Detail report" on page 106	

Broker Daily Availability report

This report shows the daily availability of the broker that you specified, including the availability of the whole day and the availability of 24 hours. Use this report to check the broker status in one day.

Table 6. Parameters of Broker Daily Availability report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select the day you want to check the broker status.
Sampling Interval	Sampling Interval	Specify the collection interval in the TEP History Collection Configuration panel.
Resource Selection	Host Name	Specify the host name of the broker that you want to check.
	Broker Name	Specify the name of the broker that you want to monitor.
Record Type	Record Type	Select 1 for snapshot report and 2 for archived report. Note: WebSphere Message Broker V7.0 or later can only use archived report.

Broker Execution Group Daily Availability report

This report shows the daily availability of the execution group that you specified, including the availability of the whole day and the availability of 24 hours. Use this report to check the execution group status in one day. This report is for WebSphere Message Broker V7.0 or later only.

Table 7. Parameters of Broker Execution Group Daily Availability report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select the day you want to check the execution group status.
Sampling Interval	Sampling Interval	Specify the collection interval in the TEP History Collection Configuration panel.
Resource Selection	Host Name	Specify the host name of the broker that you want to check.
	Broker Name	Specify the name of the broker that you want to monitor.
	Execution Group Name	Select the execution group that runs within the broker.

Broker Execution Group Weekly Availability report

This report shows the weekly availability of the execution group that you specified, including the availability of a whole week and the availability of each day in the week. Use this report to check the execution group status in one week. This report is for WebSphere Message Broker V7.0 or later only.

Table 8. Parameters of Broker Execution Group Weekly Availability report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select one day of a week that you want to check the execution group status.
Sampling Interval	Sampling Interval	Specify the collection interval in the TEP History Collection Configuration panel.
Resource Selection	Host Name	Specify the host name of the broker that you want to check.
	Broker Name	Specify the name of the broker that you want to monitor.
	Execution Group Name	Select the execution group that runs within the broker.

Broker Message Flow Daily Availability report

This report shows the daily availability of the message flow that you specified, including the availability of the whole day and the availability of 24 hours. Use this report to check the message flow status in one day. This report is for WebSphere Message Broker V7.0 or later only.

Table 9. Parameters of Broker Message Flow Daily Availability report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select the day you want to check the message flow status.
Sampling Interval	Sampling Interval	Specify the collection interval in the TEP History Collection Configuration panel.
Resource Selection	Host Name	Specify the host name of the message flow that you want to check.
	Broker Name	Specify the name of the broker.
	Execution Group Name	Specify the name of the execution group.
	Message Flow Name	Specify the name of the message flow that you want to check.
Status	Warning	Specify an integer as the warning threshold. If the message flow availability percentage is lower than this threshold, the value is marked as warning status.
	Critical	Specify an integer as the critical threshold. If the message flow availability percentage is lower than this threshold, the value is marked as critical status.

Broker Message Flow Detail report

The report shows the usage of the selected message flow during the specified time period. The line chart shows maximum elapsed microseconds, average elapsed microseconds, and minimum elapsed microseconds that the message flow takes to process in different colors. The table shows detailed information about elapsed time.

You can use this report to identify the elapsed time trend over a specified period. The report can be run hourly, daily, weekly, monthly, quarterly, or yearly.

Table 10. Parameters of Broker Message Flow Detail report

Parameter group	Parameter group	Usage
Date Range	Report Period	Select the time span for the report from the predefined time range, such as Last Week, Current Month, Last 30 Days.
	Start Date	Select a start date from the calendar and start time from the time widget. You must specify both date and time.
	End Date	Select an end date from the calendar and start time from the time widget. You must specify both date and time.
Summarization Selection	Summarization Type	Select the summarization types, such as Hourly, Daily, Weekly, Monthly, Quarterly, Yearly from the list.
	Shift Period	If shifts are enabled, the hourly table displays the shift period as 1 or 2, depending on the peak and off-peak hours that are configured in the data warehouse. The daily table consists of 1 and 2 corresponding to the peak and off-peak hours, and -1 corresponding to the summarized value for that day. If shifts are not enabled, the default value is -1.
	Vacation Period	If the vacation period is not enabled, the default value is -1. You can enter the value 1 or 2, if the vacation period is enabled.
Resource Selection	Broker Name	Specify the name of the broker that you want to monitor.
	Execution Group Name	Specify the name of the execution group that runs within the broker.
	Message Flow Name	Specify the name of the message flow that is associated with the execution group.
Record Type	Record Type	Select 1 for snapshot report and 2 for archived report. Note: WebSphere Message Broker V7.0 or later can only use archived report.

Broker Message Flow Weekly Availability report

This report shows the weekly availability of the message flow that you specified, including the availability of a whole week and the availability of each day in the week. Use this report to check the message flow status in one week. This report is for WebSphere Message Broker V7.0 or later only.

Table 11. Parameters of Broker Message Flow Weekly Availability report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select one day of a week that you want to check the message flow status.
Sampling Interval	Sampling Interval	Specify the collection interval in the TEP History Collection Configuration panel.

Table 11. Parameters of Broker Message Flow Weekly Availability report (continued)

Parameter group	Parameter name	Usage
Resource Selection	Host Name	Specify the host name of the broker that you want to check.
	Broker Name	Specify the name of the broker that you want to monitor.
	Execution Group Name	Specify the name of the execution group.
	Message Flow Name	Specify the name of the message flow that you want to check.
Status	Warning	Specify an integer as the warning threshold. If the message flow availability percentage is lower than this threshold, the value is marked as warning status.
	Critical	Specify an integer as the critical threshold. If the message flow availability percentage is lower than this threshold, the value is marked as critical status.

Broker Top n Elapsed Microseconds report

This report shows the top n message flows that take the most average time in microseconds to process. By default, this report shows the top 20 message flows for the selected broker. The chart view of the report shows the top n message flows and highlights the average elapsed time in different colors according to the threshold that you specified. The table shows more details about each message flow.

You can use this report to identify the overused message flows of a collection of systems. The report can be run hourly, daily, weekly, monthly, quarterly, or yearly. You can also click the concerned message flow in the chart view and navigate to the Message Flow Detail report to see more details of the selected message flow.

Table 12. Parameters of Broker Top n Elapsed Microseconds report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select the time span for the report from the predefined time range, such as Last Week, Current Month, Last 30 Days.
	Start Date	Select a start date from the calendar and start time from the time widget. You must specify both date and time.
	End Date	Select an end date from the calendar and start time from the time widget. You must specify both date and time.

Table 12. Parameters of Broker Top n Elapsed Microseconds report (continued)

Parameter group	Parameter name	Usage
Summarization Selection	Summarization Type	Select the summarization types, such as Hourly, Daily, Weekly, Monthly, Quarterly, Yearly from the list.
	Shift Period	If shifts are enabled, the hourly table displays the shift period as 1 or 2, depending on the peak and off-peak hours that are configured in the data warehouse. The daily table consists of 1 and 2 corresponding to the peak and off-peak hours, and -1 corresponding to the summarized value for that day. If shifts are not enabled, the default value is -1.
	Vacation Period	If the vacation period is not enabled, the default value is -1. You can enter the value 1 or 2, if the vacation period is enabled.
Number of Systems	Number of Systems to Display	Specify an integer as the number of top resources that you want to see.
Record Type	Record Type	Select 1 for snapshot report and 2 for archived report. Note: WebSphere Message Broker V7.0 or later can only use archived report.
Elapsed Microseconds Prompt Status	Warning	Specify an integer as the warning threshold. If the average elapsed time in microseconds of a message flow exceeds this threshold, the value is marked as warning status.
	Critical	Specify an integer as the critical threshold. If the average elapsed time in microseconds of a message flow exceeds this threshold, the value is marked as critical status.

Broker Weekly Availability report

This report shows the weekly availability of the broker that you specified, including the availability of a whole week and the availability of each day in the week. Use this report to check the broker status in one week.

Table 13. Parameters of Broker Weekly Availability report

Parameter group	Parameter name	Usage
Date Range	Report Period	Select one day of a week you want to check the broker status.
Sampling Interval	Sampling Interval	Specify the collection interval in the TEP History Collection Configuration panel.
Resource Selection	Host Name	Specify the host name of the broker that you want to check.
	Broker Name	Specify the name of the broker that you want to monitor.
Record Type	Record Type	Select 1 for snapshot report and 2 for archived report. Note: WebSphere Message Broker V7.0 or later can only use archived report.

Known problems and workarounds

This section contains problems that might occur when you work with Cognos reports.

Arithmetic overflow errors in ad hoc querying

If you drag certain columns in an ad hoc query and it returns an arithmetic overflow error, switch to preview with limited data or to preview with no data and add Standard Timestamp to the query.

Certain columns might average or sum up to a number that is larger than the size supported by the database. So the SQL error of arithmetic overflow is returned. If you see the data by hourly timestamp or daily timestamp, or set a query to limit the data, the aggregated value is forced to be within the supported size.

No data available in ad hoc querying on two tables

The No data available message is displayed in an ad hoc querying on two tables, but the resulting data indicates that the two tables are queried.

This error occurs because there is no relationship defined between the two tables. Make sure all your ad hoc queries have at least one identifier.

Errors of missing table or attribute

Make sure that all the prerequisites are met and the warehouse is collecting historical data. If you enable historical data collection appropriately, you are able to use the data model for WebSphere Message Broker Monitoring agent.

If you have all the required tables but still get this error, it might be because the WebSphere Message Broker Monitoring agent is not compatible with the version that is used in the generic model.

To check the query that runs, open the report in the Report Studio. Click **Tools > Show Generated SQL/MDX**. The queries in the report are displayed. You can view the native SQL.

The table schema is not ITMUser

If the table schema that you use is not ITMUser, use the framework manager to update the schema to what you used.

1. Extract the report package to your local system that the framework manager is installed on.
2. In the model\WebSphere Message Broker Monitoring Agent v7.1 Data Model folder, open the WebSphere Message Broker Monitoring Agent v7.1 Data Model.cpf file with the framework manager.
3. In the Project Viewer view, expand **Data Sources** and click **TDW**.
4. In the Property view, change the schema property from ITMUser to the schema that you used and save the changes.
5. In the Project Viewer view, expand **Packages**, right-click the WebSphere Message Broker Monitoring Agent package and click **Publish Packages** to publish the package to the Cognos server.

Duplicated broker names for the Broker Name selection

If the managed system node name that is related to a broker is changed, duplicate broker names might exist in the **Broker Name** field on the **Parameter Selection** page. This is because the node name change affect the data correlation. The data for the same broker is not correlated for together until after the old node name has rolled out of the warehouse over time.

To distinguish the duplicated broker names, edit the report in the report studio to change the Display Value property of the **Broker Name** parameter to Origin Node on the **Parameter Selection** page.

Remember: A change to any of the following values can cause the change of the managed system node name that is associated with the agent data for brokers:

- WebSphere Message Broker Monitoring agent ID
- Name of the broker
- Alias name of the broker

Chapter 8. Configuring in a cluster environment on Windows systems

This section contains instructions for configuring the WebSphere Message Broker Monitoring agent to run in a Microsoft Cluster Service (MSCS) cluster environment on Windows systems.

MSCS clusters are different from WebSphere MQ clusters, as follows:

WebSphere MQ clusters

WebSphere MQ clusters are groups of two or more queue managers running on one or more computers, providing automatic interconnection, and allowing queues to be shared for load balancing and redundancy.

MSCS clusters

MSCS clusters are groups of two or more computers, connected together and configured in such a way that, if one fails, MSCS performs a failover, transferring the state data of applications from the failing computer to another computer in the cluster and re-initiating their operation there.

You can use MSCS to connect servers into a cluster, giving higher availability of data and applications, and making it easier to manage the system. MSCS can automatically detect and recover from server or application failures. For more information about MSCS clusters, see “MSCS clusters.”

The WebSphere Message Broker Monitoring agent supports both active/active and active/passive clustering. If you are configuring the agent in an active/active cluster environment, see “Active/active clustering” on page 115 for instructions. If you are configuring the agent in an active/passive cluster environment, see “Active/passive clustering” on page 120 for instructions.

MSCS clusters

By using the Microsoft Cluster Service (MSCS), you can connect servers into a cluster, which provides higher availability of data and applications, and makes it easier to manage the system. MSCS can automatically detect and recover from server or application failures.

MSCS supports failover of virtual servers, which correspond to applications, Web sites, print queues, or file shares (including their disk spindles, files, IP addresses, and other aspects).

A failover is an automatic operation that switches to a redundant or standby system in an event of a software, hardware, or network interruption. By using the failover operation, MSCS detects a failure in an application on one computer in the cluster, and shuts down the disrupted application in an orderly manner, transfers its state data to the other computer, and re-initiates the application there.

To understand how a failover works, start by looking at a two-computer cluster. A *two-computer cluster* comprises two computers (for example, A and B) that are jointly connected to a network for client access by using a virtual IP address. They might also be connected to each other by one or more private networks. A and B share at least one disk for the server applications on each to use. Another shared

disk must be set up as a redundant array of independent disks (RAID) Level 1, for the exclusive use of MSCS; this shared disk is known as the *quorum disk*. You use MSCS to monitor both computers to check that the hardware and software are running correctly.

In a simple setup of a two-computer cluster, both computers have all the applications installed on them, but only computer A runs with live applications; computer B is just running and waiting. If computer A encounters any one of a range of problems, MSCS shuts down the disrupted application in an orderly manner, transfers its state data to the other computer, and re-initiates the application on the second computer. Applications can be made cluster-aware so that they interact fully with MSCS and fail over gracefully.

A typical setup for a two-computer cluster is shown in Figure 14.

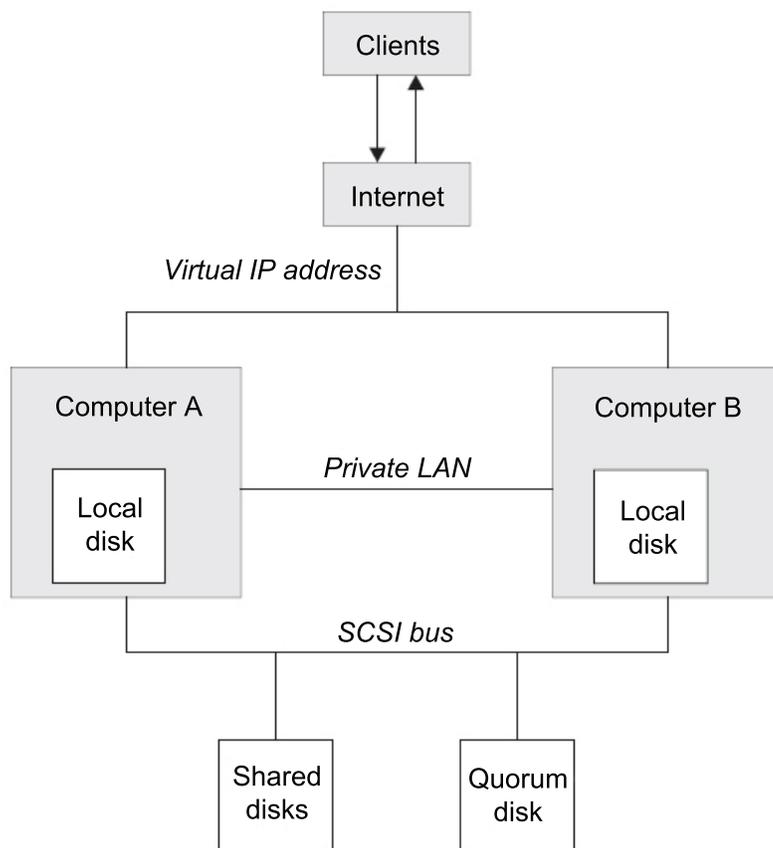


Figure 14. A two-computer MSCS cluster

Each computer can access the shared disk, but only one at a time, under the control of MSCS. In the event of a failover, MSCS switches the access to the other computer. The shared disk is typically a RAID, but need not be.

Each computer is connected to the external network for client access, and each has an IP address. However, an external client, communicating with this cluster, sees only one virtual IP address, and MSCS routes the IP traffic within the cluster appropriately.

MSCS also communicates between the two computers, either over one or more private connections or over the public network, to monitor their states by using the heartbeat signal, synchronize their databases, and perform other related tasks.

Active/active clustering

This section describes how you can configure the WebSphere Message Broker Monitoring agent to run in an active/active cluster environment.

Prerequisites

Before you begin configuring the WebSphere Message Broker Monitoring agent to run in an active/active cluster environment, ensure that the two systems that host the WebSphere Message Broker Monitoring agent are correctly configured. Ensure that both systems fulfill the following requirements:

- Microsoft Windows 2003 Server is installed and includes Microsoft Cluster Server (MSCS), which is used to manage your cluster environment.
- You have used MSCS to configure both systems as cluster nodes.
- WebSphere Message Broker is installed and configured to run in a cluster environment. See your WebSphere Message Broker documentation for information about how to install WebSphere Message Broker in a cluster environment.
- WebSphere MQ is installed and configured to run in a cluster environment. See your WebSphere MQ documentation for information about how to install WebSphere MQ in a cluster environment.
- Message brokers and configuration managers that you want to monitor are created.
- The DB2 database is installed to run in a domain environment. See the DB2 documentation for information about how to install the DB2 software in a domain environment.
- The IBM Tivoli Monitoring framework is installed. This monitoring software must be installed separately on both cluster nodes. For instructions about how to install IBM Tivoli Monitoring in a cluster environment, see your IBM Tivoli Monitoring documentation.
- The WebSphere Message Broker Monitoring agent version 7.0 or higher version is installed. This agent must be installed separately on both cluster nodes. See *Installation and Setup Guide, SC14-7520* for installation instructions.

Also ensure that you have two separate logical drives in the cluster environment available for storing log and historical data collected from the agents. These drives are referenced as drives R and S in the following procedure.

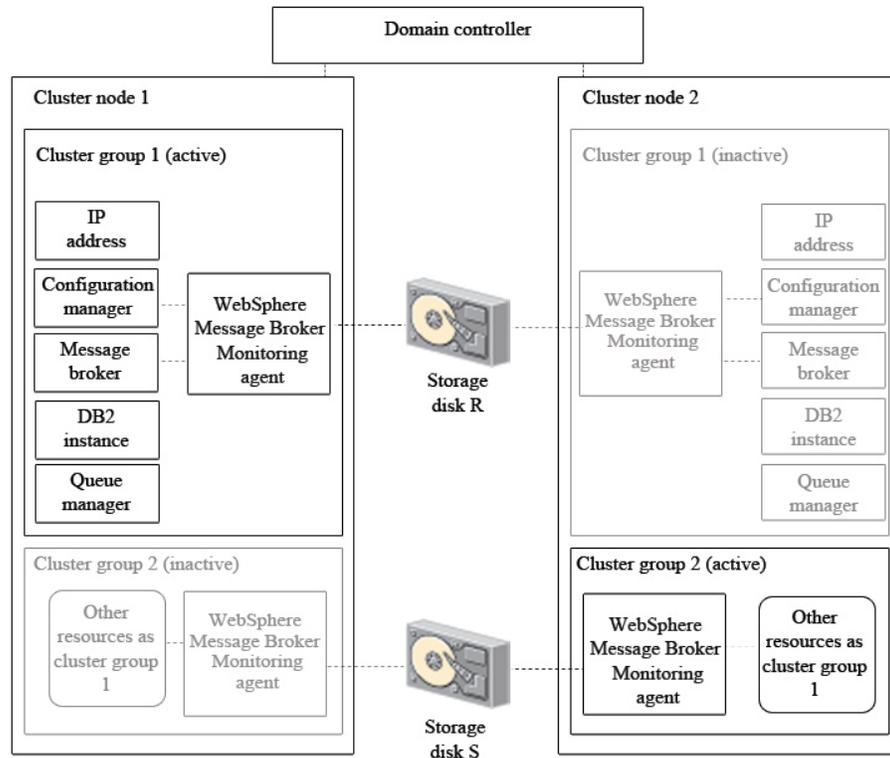


Figure 15. An example active/active cluster environment architecture with one cluster group active on each cluster node

An example of an active/active cluster environment is displayed in Figure 15. The environment consists of two cluster nodes on separate physical systems. Each cluster node hosts two cluster groups. The cluster groups hosted by each system are the same, so between them there are two identical copies of cluster group 1 and two identical copies of cluster group 2. Each cluster group contains the following resources:

- A message broker
- A configuration manager
- A DB2 instance
- An IP address
- A queue manager
- A WebSphere Message Broker Monitoring agent

Only one copy of each cluster group can be active simultaneously. For example, when cluster group 1 is active on cluster node 1 (as in Figure 15), the copy of cluster group 1 on cluster node 2 is inactive. In most environments with two cluster nodes and two cluster groups where both cluster nodes are running correctly, one cluster group is running on each cluster node, balancing the load between the two systems. If one of the nodes fails, the second cluster group on the node that is still active is started to continue the work of the cluster group that was active on the node that failed.

Information shared between different copies of the same agent, such as logs, is stored on a separate disk that can be accessed by all copies of the agent running on different cluster nodes. If the node that hosts the active agent fails and a copy of

the agent on the other node is started, shared information such as log files can still be read and written to the disk as if the same copy of the agent was still running. The agent is installed separately on each cluster node.

Configuring the WebSphere Message Broker Monitoring agent

To configure the WebSphere Message Broker Monitoring agent to run in a cluster environment, complete the following steps:

Important: To complete the following procedure, you must have two cluster groups on each cluster node because it is the most common scenario. If you have more than two cluster groups, create additional instances of the WebSphere Message Broker Monitoring agent to monitor the message brokers in each additional cluster group.

1. Create new instances of the WebSphere Message Broker Monitoring agent on both cluster nodes.
2. Set local variables on each cluster node.
3. Configure the Tivoli Enterprise Portal Server to list agents running in cluster groups by cluster name instead of host system name in Tivoli Enterprise Portal.
4. Use the Cluster Administrator ID to add a resource of the **Generic Service** type named KQI1 to cluster group 1.
5. Use the Cluster Administrator ID to add a resource of the **Physical Disk** type named R to cluster group 1.
6. Use the Cluster Administrator ID to add a resource of the **Generic Service** type named KQI2 to cluster group 2.
7. Use the Cluster Administrator ID to add a resource of the **Physical Disk** type named S to cluster group 2.
8. Use the Cluster Administrator ID to set the group owner of cluster group 1 to cluster node 1 and the group owner of cluster group 2 to cluster node 2.
9. Use the Cluster Administrator ID to start the broker, configuration manager, WebSphere Message Broker Monitoring agent, and other resources in each cluster group.

You have now completed the configuration of the WebSphere Message Broker Monitoring agent to monitor brokers in an active/active cluster environment.

Creating new instances of the WebSphere Message Broker Monitoring agent

Create new instances of the WebSphere Message Broker Monitoring agent on both cluster nodes by performing the following procedure:

1. Start Manage Tivoli Enterprise Monitoring Services.
2. Right-click the WebSphere Message Broker Monitoring agent, and then click **Create Instance** to create a new instance of the WebSphere Message Broker Monitoring agent to monitor the broker and configuration manager in cluster group 1.
3. Enter a name for the instance when prompted. In this procedure, use KQI1 for the instance name. Click **OK**.
4. Right-click the WebSphere Message Broker Monitoring agent, and then click **Create Instance** again to create a second new instance of the WebSphere Message Broker Monitoring agent to monitor the broker and configuration manager in cluster group 2.
5. Enter a name for the instance when prompted. In this procedure, use KQI2 for the instance name. Click **OK**.

6. Edit the KQI1 and KQI2 configuration files as follows:
 - a. Add the following attribute directly after the version attribute inside the KqiAgent tag, where *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance. Enter a string of up to four characters. The instance name must be different from all other agent instance names that run on the same system node. However, this value must be identical to the value of the *instance_name* used for the agent instance monitoring the failover broker by the same name on the failover node.


```
agentId="instance_name"
```
 - b. Add the following tags directly after the opening KqiAgent tag, where *broker_name* is the name of the message broker you want to monitor.


```
<MonitorBroker name="broker_name"> </MonitorBroker>
```
 - c. Optional: If the agent is only monitoring a single message broker, set the **discoveryInterval** parameter to a high value. This setting can reduce the number of unnecessary broker discover operations performed. For example, you can set the value to 2592000 seconds (30 days).

The resulting file looks similar to the following example:

```
<KqiAgent version="710"
  agentId="KQI"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="2592000"
  defaultStatisticInterval="60"
  defaultFlowEventInterval="15"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  holdTimeForQuery="180"
  defaultReplyQueueName="KQI.AGENT.REPLY.QUEUE"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultTakeActionAuthUsers="*">
  <MonitorBroker name="BK1"> </MonitorBroker>
</KqiAgent>
```

- d. Stop the primary WebSphere Message Broker Monitoring agent instance.

Remember: Do not use the primary WebSphere Message Broker Monitoring agent to monitor brokers in a cluster environment.

Setting local variables on each cluster

Set local variables on each cluster node by performing the following procedure:

1. Start Manage Tivoli Enterprise Monitoring Services.
2. Right-click the KQI1 agent instance that you created in the previous procedure, and then click **Advanced > Edit Variables**.
3. In the Override Local Variable Settings window, add the following variables and specify their values:
 - CTIRA_SIT_PATH
 - CTIRA_LOG_PATH
 - CTIRA_HIST_DIR

Remember: Each agent must have its own logical drive on which to store data. More than one agent cannot share a single drive.

If you specify the values as in the following example, a directory named R:\WMB\kqi\BK\logs is created to store the logs of the WebSphere Message Broker Monitoring agent, the R:\WMB\kqi\BK\logs\History directory is created

to store historical data of the agent, and the R:\WMB\kqi\BK\sitpath directory is created to store the situation data, where R is the letter assigned to the drive where log files and historical data collected by KQI1 agent are stored:

```
CTIRA_SIT_PATH=R:\WMB\kqi\BK\sitpath
CTIRA_LOG_PATH=R:\WMB\kqi\BK\logs
CTIRA_HIST_DIR=R:\WMB\kqi\BK\logs\History
```

4. To close the window, click **OK**.
5. Right-click the KQI2 agent instance that you created in the previous procedure, and then click **Advanced > Edit Variables**.
6. In the Override Local Variable Settings window, add the following variables and specify their values:
 - CTIRA_SIT_PATH
 - CTIRA_LOG_PATH
 - CTIRA_HIST_DIR

If you specify their values as in the following example, a directory named S:\WMB\kqi\BK\logs is created to store the logs of the WebSphere Message Broker Monitoring agent, S:\WMB\kqi\BK\logs\History is created to store its historical data, and S:\WMB\kqi\BK\sitpath is created to store its situation data, where S is the letter assigned to the drive where log files and historical data collected by KQI2 agent are stored. Make sure that the path names do not contain any spaces.

```
CTIRA_SIT_PATH=S:\WMB\kqi\BK\sitpath
CTIRA_LOG_PATH=S:\WMB\kqi\BK\logs
CTIRA_HIST_DIR=S:\WMB\kqi\BK\logs\History
```

7. To close the window, click **OK**.
8. Change the start mode of both KQI1 and KQI2 agents to manual startup.

Configuring the Tivoli Enterprise Portal to list agents running in the cluster groups

Use the following procedure to configure the Tivoli Enterprise Portal to list agents running in cluster groups by cluster name instead of host system name in the portal:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click the agent instance icon, and then click **Advanced > Edit Variables** from the displayed menu.
2. In the Override Local Variable Settings window, click **Add**.
3. Select CTIRA_HOSTNAME for the **Variable** field, set the value to the name of the MSCS cluster that you want to display on Tivoli Enterprise Portal, and click **OK**.

Important: Leave .TYPE=REG_EXPAND_SZ in the CTIRA_HOSTNAME variable when you change the value so that the registry is updated correctly.

4. Perform one of the following procedures, depending on which operating system the Tivoli Enterprise Portal Server is running.
 - If the Tivoli Enterprise Portal Server is running on a Windows system, perform the following procedure:
 - a. Stop the portal server if it is running.
 - b. In the Manage Tivoli Enterprise Monitoring Services window, right-click the Tivoli Enterprise Portal Server, and then click **Advanced > Edit Variables** from the displayed menu. The Tivoli Enterprise Portal Server Override Local Variable Setting window is displayed.
 - c. Click **Add**. The Add Environment Setting Override window is displayed.

- d. In the **Variable** menu, look for the **KFW_TOPOLOGY_CLUSTER_LIST** parameter. If the parameter exists, append `AFF_MQSI_AGENT AFF_MQSI_BROKER` to any existing values, separated by a space. If it does not already exist, create it and set the value to `AFF_MQSI_AGENT AFF_MQSI_BROKER`.
- e. Reconfigure and recycle the portal server.
- If the Tivoli Enterprise Portal Server is running on a UNIX or Linux system, perform the following procedure:
 - a. Stop the portal server if it is running.
 - b. Navigate to the `ITM_HOME/config` directory, where `ITM_HOME` is the IBM Tivoli Monitoring installation directory.
 - c. Open the `cq.ini` file in a text editor.
 - d. Look for the **KFW_TOPOLOGY_CLUSTER_LIST** parameter. If it already exists, add `AFF_MQSI_AGENT AFF_MQSI_BROKER` to any existing values, separated by a space. If it does not already exist, add the following line to the file:
`KFW_TOPOLOGY_CLUSTER_LIST=AFF_MQSI_AGENT AFF_MQSI_BROKER`
 - e. Save and close the file.
 - f. Reconfigure and recycle the portal server.

Active/passive clustering

This section describes how you can configure the WebSphere Message Broker Monitoring agent to run in an active/passive cluster environment.

Prerequisites

Before you begin configuring the WebSphere Message Broker Monitoring agent to run in an active/passive cluster environment, ensure that the two systems that host the WebSphere Message Broker Monitoring agent are correctly configured. Ensure that both systems fulfill the following requirements:

- Microsoft Windows 2003 Server is installed that includes Microsoft Cluster Server (MSCS), which is used to manage your cluster environment.
- You have used MSCS to configure both systems as cluster nodes.
- WebSphere Message Broker is installed and configured to run in a cluster environment. See your WebSphere Message Broker documentation for information about how to install it in a cluster environment.
- WebSphere MQ is installed and configured to run in a cluster environment. See your WebSphere MQ documentation for information about how to install it in a cluster environment.
- The DB2 database is installed to run in a cluster environment. See the DB2 documentation for information about how to install the DB2 software in a cluster environment.
- Message brokers and configuration managers that are to be monitored are created.
- The IBM Tivoli Monitoring framework is installed. This monitoring software must be installed separately on both cluster nodes. For instructions about how to install IBM Tivoli Monitoring in a cluster environment, see the IBM Tivoli Monitoring documentation.
- The WebSphere Message Broker Monitoring agent version 7.0 or higher version is installed. This agent must be installed separately on both cluster nodes. See *Installation and Setup Guide, SC14-7520* for installation instructions.

Also ensure that you have a separate logical drive in the cluster environment available for storing log and historical data collected from the agents. The drive is referenced as drive R in the following procedure.

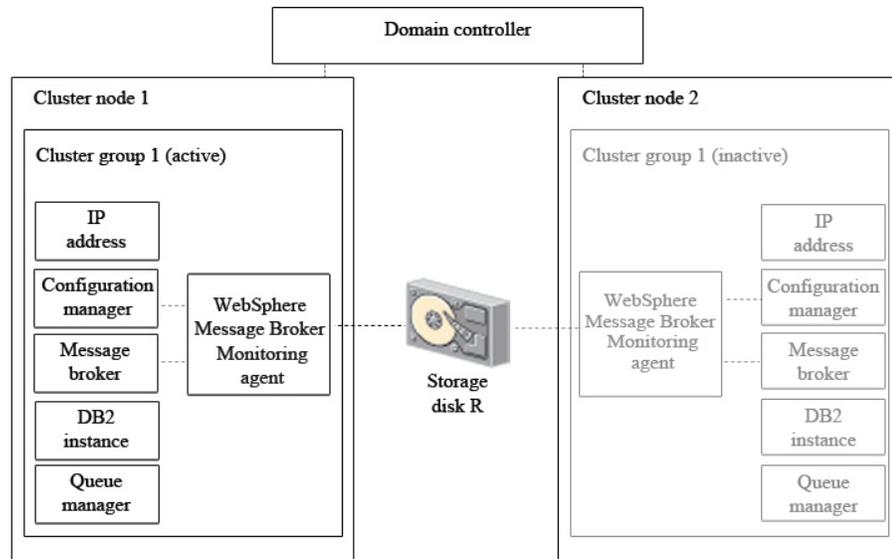


Figure 16. An example cluster environment architecture with a cluster group active on one cluster node and inactive on the other node

An example of an active/passive cluster environment is displayed in Figure 16. The environment consists of two cluster nodes on separate physical systems. Each cluster node hosts one cluster group (There is no limit to the number of cluster groups that can be hosted by a cluster node). The cluster groups hosted by each system are the same, so between them there are two identical copies of cluster group 1. Each cluster group contains the following components:

- A message broker
- A configuration manager
- A DB2 instance
- An IP address
- A queue manager
- A WebSphere Message Broker Monitoring agent

Only cluster groups on one cluster node are active at one time. For example, if cluster group 1 is active on cluster node 1 (as in Figure 16), the copy of cluster group 1 on cluster node 2 is inactive. In an active/passive cluster environment with two cluster nodes, only cluster groups on the active cluster node are running. If the active node fails, the cluster groups on the other node are started to continue the work of the cluster groups that were active on the node that failed.

Information shared between different copies of the same agent, such as logs, is stored on a separate disk that can be accessed by all copies of the agent running on different cluster nodes. If the node that hosts the active agent fails and a copy of the agent on the other node is started, shared information such as log files can still be read and written to the disk as if the same copy of the agent was still running. The agent is installed separately on each cluster node. Shared disks store logs and historical information that must be accessed by different copies of the same agent.

Configuring the WebSphere Message Broker Monitoring agent

To configure the WebSphere Message Broker Monitoring agent to run in an active/passive cluster environment, perform the following procedure:

Important: To complete the following procedure, you must one cluster group on each cluster node. If you have more than one cluster group, create additional instances of the WebSphere Message Broker Monitoring agent to monitor the resources in each additional cluster group.

1. Create a new instance of the WebSphere Message Broker Monitoring agent on both cluster nodes.
2. Set local variables on each cluster node.
3. Configure the Tivoli Enterprise Portal Server to list agents running in cluster groups by cluster name instead of host system name in Tivoli Enterprise Portal.
4. Use the Cluster Administrator ID to add a resource of type **Generic Service** named KQI to cluster group 1.
5. Use the Cluster Administrator ID to add a resource of type **Physical Disk** named R to cluster group 1.
6. Use the Cluster Administrator ID to set the group owner of cluster group 1 to cluster node 1.
7. Use the Cluster Administrator ID to start the broker, configuration manager, the WebSphere Message Broker Monitoring agent, and other resources in cluster group 1 on cluster node 1 or cluster node 2.

Now the configuration of the WebSphere Message Broker Monitoring agent to monitor brokers in an active/passive cluster environment is completed.

Creating new instances of the WebSphere Message Broker Monitoring

Remember: Do not use the primary WebSphere Message Broker Monitoring agent to monitor brokers in a cluster environment.

Create a new instance of the WebSphere Message Broker Monitoring agent by performing the following procedure on both cluster nodes:

1. Start Manage Tivoli Enterprise Monitoring Services.
2. Right-click the WebSphere Message Broker Monitoring agent, and then click **Create Instance** to create a new instance of the WebSphere Message Broker Monitoring agent to monitor the broker and the configuration manager in cluster group 1.
3. Enter a name for the instance when prompted. In this procedure, it is presumed you entered the name KQI. Click **OK**.
4. Edit the KQI agent configuration file as follows:
 - a. Add the following attribute directly after the version attribute inside the KqiAgent tag:

```
agentId="instance_name"
```

where *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance. Enter a string of up to four characters. The instance name must be different from all other agent instance names that run on the same system node. However, this value must be identical to the value of the *instance_name* used for the agent instance monitoring the failover broker by the same name on the failover node.

- b. Add the following tags directly after the opening KqiAgent tag:

```
<MonitorBroker name="broker_name"> </MonitorBroker>
```

where *broker_name* is the name of the message broker that you want to monitor.

- c. Optional: If the agent is only monitoring a single message broker, set the **discoveryInterval** parameter to a high value. This setting can reduce the number of unnecessary operations that are performed to discover brokers. For example, you can set the value to 2592000 seconds (30 days).

The resulting file looks similar to the following example:

```
<KqiAgent version="710"  
  agentId="KQI"  
  defaultRetainBrokerEvents="10"  
  defaultRetainFlowEvents="10"  
  retainProductEvents="10"  
  discoveryInterval="2592000"  
  defaultStatisticInterval="60"  
  defaultFlowEventInterval="15"  
  defaultHistoricalAccountingType="Archive"  
  defaultRetainRecentSnapshotSamples="15"  
  defaultRetainRecentArchiveSamples="5"  
  defaultRetainRecentPubSubSamples="15"  
  holdTimeForQuery="180"  
  defaultReplyQueueName="KQI.AGENT.REPLY.QUEUE"  
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"  
  defaultTakeActionAuthUsers="*">  
  <MonitorBroker name="BK1"> </MonitorBroker>  
</KqiAgent>
```

5. Stop the primary WebSphere Message Broker Monitoring agent instance.

Setting local variables on each cluster

Set local variables by performing the following procedure on each cluster node:

1. Start Manage Tivoli Enterprise Monitoring Services.
2. Right-click the KQI agent instance that you create in the previous procedure, and then click **Advanced > Edit Variables**.
3. In the Override Local Variable Settings window, add the following variables and specify their values:
 - CTIRA_SIT_PATH
 - CTIRA_LOG_PATH
 - CTIRA_HIST_DIR

If you specify their values as in the following example, a directory named R:\WMB\kqi\BK\logs is created to store the logs of the WebSphere Message Broker Monitoring agent and R:\WMB\kqi\BK\logs\History is created to store its historical data. However, you need to manually create the directory specified for CTIRA_SIT_PATH on the shared disk, which is R:\WMB\kqi\BK\sitpath in this example.

```
CTIRA_SIT_PATH=R:\WMB\kqi\BK\sitpath  
CTIRA_LOG_PATH=R:\WMB\kqi\BK\logs  
CTIRA_HIST_DIR=R:\WMB\kqi\BK\logs\History
```

where R is the letter assigned to the drive where log files and historical data collected by WebSphere Message Broker Monitoring agent are stored.

Remember:

- Make sure that the path names do not contain any spaces.

- Each agent must have its own logical drive on which to store data. More than one agent cannot share a single drive.
4. To close the window, click **OK**.
 5. Change the start mode of both KQI agents to manual startup.

Configuring the Tivoli Enterprise Portal to list agents running in the cluster groups

Use the following procedure to configure the Tivoli Enterprise Portal to list agents running in cluster groups by cluster name instead of host system name in the portal:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click the agent instance icon, and then click **Advanced > Edit Variables** from the displayed menu.
2. In the Override Local Variable Settings window, click **Add**.
3. Select CTIRA_HOSTNAME for the **Variable** field, set the value to the name of the MSCS cluster that you want to display on Tivoli Enterprise Portal, and click **OK**.

Important: Leave `.TYPE=REG_EXPAND_SZ` in the CTIRA_HOSTNAME variable when you change the value so that the registry is updated correctly.

4. Perform one of the following procedures, depending on which operating system the Tivoli Enterprise Portal Server is running.
 - If the Tivoli Enterprise Portal Server is running on a Windows system, perform the following procedure:
 - a. Stop the portal server if it is running.
 - b. In the Manage Tivoli Enterprise Monitoring Services window, right-click the Tivoli Enterprise Portal Server, and then click **Advanced > Edit Variables** from the displayed menu. The Tivoli Enterprise Portal Server Override Local Variable Setting window is displayed.
 - c. Click **Add**. The Add Environment Setting Override window is displayed.
 - d. In the **Variable** menu, look for the **KFW_TOPOLOGY_CLUSTER_LIST** parameter. If the parameter exists, append `AFF_MQSI_AGENT AFF_MQSI_BROKER` to any existing values, separated by a space. If it does not already exist, create it and set the value to `AFF_MQSI_AGENT AFF_MQSI_BROKER`.
 - e. Reconfigure and recycle the portal server.
 - If the Tivoli Enterprise Portal Server is running on a UNIX or Linux system, perform the following procedure:
 - a. Stop the portal server if it is running.
 - b. Navigate to the `ITM_HOME/config` directory, where `ITM_HOME` is the IBM Tivoli Monitoring installation directory.
 - c. Open the `cq.ini` file in a text editor.
 - d. Look for the **KFW_TOPOLOGY_CLUSTER_LIST** parameter. If it already exists, add `AFF_MQSI_AGENT AFF_MQSI_BROKER` to any existing values, separated by a space. If it does not already exist, add the following line to the file:
`KFW_TOPOLOGY_CLUSTER_LIST=AFF_MQSI_AGENT AFF_MQSI_BROKER`
 - e. Save and close the file.
 - f. Reconfigure and recycle the portal server.

Known limitations

Occasionally, the status of a broker and the broker configuration manager associated with it might be different in the Broker Information workspace in Tivoli Enterprise Portal. The information displayed in the portal might also be different from that displayed in the Cluster Administrator. However, the information displayed in Tivoli Enterprise Portal is correct.

This situation occurs because of the MSCS cluster failing to completely stop the broker. If this happens, manually stop the component that is not running (either the broker or configuration manager) and then restart it again if necessary. Run the following command to stop the component:

```
mqsisstop <component_name>
```

Chapter 9. Configuring in a cluster environment on AIX systems

This section contains instructions for configuring the WebSphere Message Broker Monitoring agent to run in a cluster environment on AIX systems using High Availability Cluster Multi Processing (HACMP™). For information about how to configure hardware such as redundant power supplies, redundant disk controllers, disk mirroring, or multiple network or adapter configurations, see the HACMP documentation.

The WebSphere Message Broker Monitoring agent can be configured to run in either an active/active or active/passive environment. Before you begin configuring the WebSphere Message Broker Monitoring agent, ensure that your environment fulfills the requirements stated in “Prerequisites.”

For instructions about how to install and configure the WebSphere Message Broker Monitoring agent for use in an HACMP cluster environment, see “Configuring the WebSphere Message Broker Monitoring agent” on page 130

Prerequisites

The WebSphere Message Broker Monitoring agent can be configured to run in either an active/active or active/passive environment. Before you begin configuring the WebSphere Message Broker Monitoring agent, be sure to review the following requirements for either environment that the two systems forming the cluster nodes must fulfill:

- “Active/active clustering”
- “Active/passive clustering” on page 129

Active/active clustering

Before you begin configuring the WebSphere Message Broker Monitoring agent to run in an HACMP active/active cluster environment, ensure that the two systems that form the cluster nodes in the environment are correctly configured. Both systems must fulfill the following requirements:

- The HACMP software is installed and your HACMP cluster environment is correctly configured.
- Both cluster nodes have access to a minimum of two shared disks, on which historical information shared between copies of the WebSphere Message Broker Monitoring agent running on different cluster nodes is stored. You must have a separate shared disk available in your cluster environment for each instance of the agent. If you want to have more than two agents running on each cluster node, increase the number of shared disks accordingly.
- WebSphere MQ is installed and configured to run in an HACMP cluster environment. See your WebSphere MQ documentation for information about how to install WebSphere MQ in a cluster environment.
- The message brokers that you want to monitor have been created on both cluster nodes within the HACMP cluster environment. Ensure that failover occurs correctly. See your message broker product documentation for more information.

- The DB2 database is configured to run in an HACMP environment. See the DB2 documentation for information about how to install the DB2 software in an HACMP environment.

An example of a cluster environment is displayed in Figure 17. The environment consists of two cluster nodes running on separate physical systems. Each cluster node hosts two cluster groups. The cluster groups hosted by each system are the same, so between them there are two identical copies of cluster group 1 and two identical copies of cluster group 2. Each cluster group contains one or more message brokers and an instance of the WebSphere Message Broker Monitoring agent to monitor each message broker.

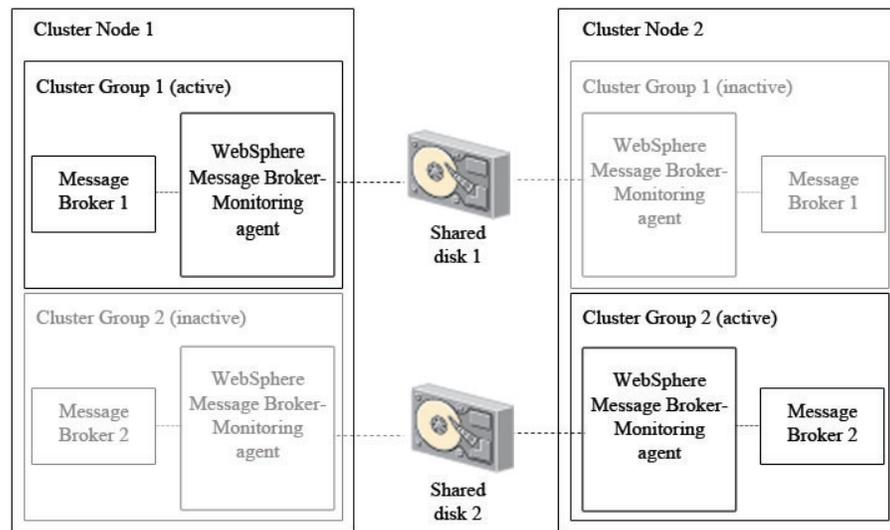


Figure 17. An example active-active cluster environment architecture with one cluster group active on each cluster node

Only one copy of each cluster group can be active simultaneously. For example, when cluster group 1 is active on cluster node 1 (as in Figure 17), the copy of cluster group 1 on cluster node 2 is inactive. In most environments with two cluster nodes and two cluster groups where both cluster nodes are running correctly, only one cluster group runs on each cluster node, balancing the load between the two systems. When one of the nodes fails, the second cluster group on the node that is still active is started to continue the work of the cluster group that was active on the node that failed.

Information shared between different copies of the same agent, such as historical data files, is stored on a separate disk that can be accessed by all copies of that agent running on different cluster nodes. In active/active clustering, at least two instances of the agent run on each cluster node, each requiring a separate shared disk to store shared information. If the node that hosts the active agent fails and a copy of the agent on the other node is started, shared information such as historical data files can still be read and written to the disk as if the same copy of the agent was still running.

To install and configure the WebSphere Message Broker Monitoring agent, repeat the procedure in “Configuring the WebSphere Message Broker Monitoring agent” on page 130 for each instance of the WebSphere Message Broker Monitoring agent in your environment.

Important: You must repeat this procedure for different copies of the same agent instance running on different cluster nodes.

Active/passive clustering

Before you begin configuring the WebSphere Message Broker Monitoring agent to run in an HACMP active/passive cluster environment, ensure that the two systems that form the cluster nodes in the environment are correctly configured. Both systems must fulfill the following requirements:

- The HACMP software is installed and your HACMP cluster environment is correctly configured.
- Both cluster nodes have access to a shared disk, on which historical information shared between copies of the WebSphere Message Broker Monitoring agent running on different cluster nodes is stored.
- WebSphere MQ is installed and configured to run in an HACMP cluster environment. See your WebSphere MQ documentation for information about how to install WebSphere MQ in a cluster environment.
- The message brokers that you want to monitor have been created on both cluster nodes within the HACMP cluster environment. Ensure that failover occurs correctly. See your message broker product documentation for more information.
- The DB2 database is configured to run in an HACMP environment. See your DB2 documentation for information about how to install the DB2 software in an HACMP environment.

An example of an active/passive cluster environment is displayed in Figure 18. The environment consists of two cluster nodes running on separate physical systems. The cluster groups hosted by each system are the same, so between them there are two identical copies of cluster group 1. Each cluster group contains one or more queue managers and an instance of the WebSphere Message Broker Monitoring agent to monitor each message broker.

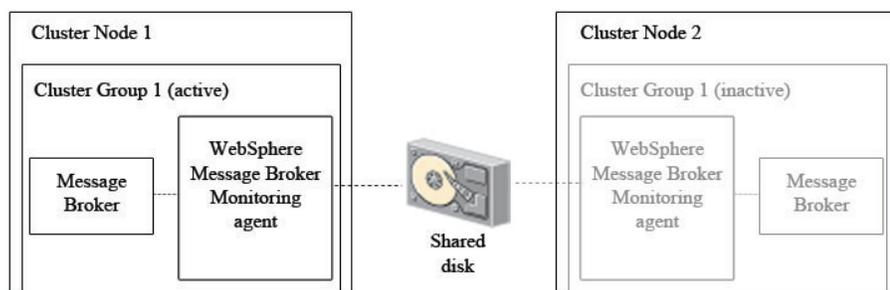


Figure 18. An example active-passive cluster environment architecture

Only cluster groups on one cluster node are active at one time. For example, when cluster group 1 is active on cluster node 1 (as in Figure 18), the copy of cluster group 1 on cluster node 2 is inactive. In an active/passive cluster environment

with two cluster nodes, only cluster groups on the active cluster node are running. If the active node fails, the cluster groups on the other node are started to continue the work of the cluster groups that were active on the node that failed.

Information shared between different copies of the same agent, such as historical data files, is stored on a separate disk that can be accessed by all copies of the agent running on different cluster nodes. If the node that hosts the active agent fails and a copy of the agent on the other node is started, shared information such as historical data files can still be read and written to the disk as if the same copy of the agent was still running. The agent is installed separately on each cluster node.

Remember: Using a shared disk is preferred over using a Network File System (NFS) mounted file system because results are unpredictable when the NFS mounted file system is not available.

To install and configure the WebSphere Message Broker Monitoring agent, repeat the procedure in “Configuring the WebSphere Message Broker Monitoring agent” for each instance of the WebSphere Message Broker Monitoring agent in your environment.

Important: You must repeat this procedure for different copies of the same agent instance running on different cluster nodes.

Configuring the WebSphere Message Broker Monitoring agent

Remember: WebSphere Message Broker supportpac IC91 must have been installed when installing WebSphere Message Broker product in the HACMP cluster environment. See your WebSphere Message Broker product documentation for further information.

To install and configure the WebSphere Message Broker Monitoring agent for use in an HACMP cluster environment, perform the following steps:

1. Install the WebSphere Message Broker Monitoring agent on the cluster node on which you want the agent to run. For installation instructions, see *Installation and Setup Guide, SC14-7520*.
2. Create new instances of the WebSphere Message Broker Monitoring agent for each message broker.
3. Set local variables in the `qi.ini` configuration file.
4. Create the directories where historical and situation data files are stored.
5. Configure the Tivoli Enterprise Portal Server to list agents running in cluster groups by the cluster name instead of the host name in the Tivoli Enterprise Portal.
6. Create a file containing the startup script used to start the agent.
7. Create a file containing the shutdown script used to stop the agent.
8. Set the scripts that are used to start and stop the agent in HACMP.

Important: You must repeat this procedure for different copies of the same agent instance that are running on different cluster nodes.

The WebSphere Message Broker Monitoring agent is now configured to operate in an HACMP cluster environment.

Creating new instances of the WebSphere Message Broker Monitoring agent for each message broker

Create new instances of the WebSphere Message Broker Monitoring agent for each message broker that you want to monitor by performing the following procedure:

1. Navigate to the *ITM_HOME*/config directory, where *ITM_HOME* is the directory where the IBM Tivoli Monitoring is installed. The default directory is /opt/IBM/ITM.
2. Create a new configuration file for each instance of the WebSphere Message Broker Monitoring agent by copying the default *kqi.xml* configuration file to the *hostname_qi_agent_instance_ID.xml* file, where *hostname* is the host name of cluster node, *agent_instance_ID* is the WebSphere Message Broker Monitoring agent ID.

Remember: The *agent_instance_ID* is string of up to four characters. It must be different for each agent instance that runs on the same cluster node. However, it must be identical to the value for the agent instance running on the failover cluster node that corresponds to this instance.

3. Edit each of the newly created configuration files as follows:
 - a. Add the following attribute directly after the version attribute inside the KqiAgent tag:

```
agentId="agent_instance_ID"
```

where *agent_instance_ID* is the ID of the WebSphere Message Broker Monitoring agent instance. This must be the same as that specified in the filename in the previous step.

- b. Add the following tags directly after the opening KqiAgent tag:

```
<MonitorBroker name="broker_name">
</MonitorBroker>
```

where *broker_name* is the name of the message broker that you want to monitor.

- c. Set the **discoveryInterval** parameter to a high value, because the agent is only monitoring a single message broker. This setting can reduce the number of unnecessary operations that are performed to discover brokers. For example, you can set the value to 2592000 seconds (30 days). The resulting file looks similar to the following example:

```
<KqiAgent version="710"
  agentId="MQHA"
  defaultRetainBrokerEvents="10"
  defaultRetainFlowEvents="10"
  retainProductEvents="10"
  discoveryInterval="2592000"
  defaultStatisticInterval="60"
  defaultFlowEventInterval="15"
  defaultHistoricalAccountingType="Archive"
  defaultRetainRecentSnapshotSamples="15"
  defaultRetainRecentArchiveSamples="5"
  defaultRetainRecentPubSubSamples="15"
  holdTimeForQuery="180"
  defaultReplyQueueName="KQI.AGENT.REPLY.QUEUE"
  defaultReplyQueueModel="SYSTEM.BROKER.MODEL.QUEUE"
  defaultCollectNodeData="YES"
  defaultTakeActionAuthUsers="*"
  <MonitorBroker name="HABK">
  </MonitorBroker>
</KqiAgent>
```

Setting local variables in the agent configuration file

Set local variables by performing the following procedure:

1. Navigate to the *ITM_HOME/config* directory, where *ITM_HOME* is the directory where IBM Tivoli Monitoring is installed.
2. Open the *qi.ini* file.
3. Locate the `$CTIRA_SIT_PATH` variable and change it to the following code:
`CTIRA_SIT_PATH=$CANDLEHOME/$BINARCH$/PRODUCTCODE$/sit$INSTANCE_HISTORY$`

Remember: Using the `$INSTANCE_HISTORY$` variable in the *qi.ini* file distinguishes the instances of the WebSphere Message Broker Monitoring agent on the cluster node.

4. Save and close the *qi.ini* file.

Creating the directories for historical and situation data files

Create the directories where historical and situation data files are stored by performing the following procedure on each cluster node:

1. Create the *ITM_HOME/aix513/qi/sit* directory for storing situation data, where *ITM_HOME* is the directory in which IBM Tivoli Monitoring is installed.

Remember: The user ID that is used to run the agent must have write access to this directory.

2. Create separate directories for storing historical and situation data from each agent instance on the shared disk as follows, where *disk_name* is the name of the shared disk on which historical data is stored, and *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance.
 - For storing historical data: *disk_name/kqi/hist/instance_name*
 - For storing situation data: *disk_name/kqi/sit/instance_name*

Remember: The user ID that is used to run the agent must have write access to these directories.

3. Create a link to the shared disk on which historical data is stored by running the following command, where *disk_name* is the name of the shared disk, *ITM_HOME* is the directory in which IBM Tivoli Monitoring is installed, and *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance.

```
ln -sf disk_name/kqi/hist/instance_name ITM_HOME/aix513/qi/hist/instance_name
```

Remember: The symbolic link definition is used to equate the *qi.ini* specification to a directory on the shared disk on which historical and situation data is stored.

4. Create a link to the shared disk on which situation data is stored by running the following command, where *disk_name* is the name of the shared disk, *ITM_HOME* is the directory in which IBM Tivoli Monitoring is installed, and *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance:

```
ln -sf disk_name/kqi/sit/instance_name ITM_HOME/aix513/qi/sit/instance_name
```

Remember: The symbolic link definition is used to equate the *qi.ini* specification to a directory on the shared disk on which historical and situation data is stored.

Configuring the Tivoli Enterprise Portal to list agents running in the cluster groups

Configure the Tivoli Enterprise Portal Server to list agents that are running in the cluster groups by the cluster name instead of the host name in the Tivoli Enterprise Portal. Perform the following steps:

1. Navigate to the *ITM_HOME/config* directory.
2. Open the *qi.ini* agent configuration file.
3. Add the following line, *CTIRA_HOSTNAME=cluster_name*, where *cluster_name* is the name of the HACMP cluster that is displayed in Tivoli Enterprise Portal. This value must be identical to the value of the *CTIRA_HOSTNAME* attribute of the agent running on the failover node.
4. Perform one of the following procedures, depending on which operating system the Tivoli Enterprise Portal Server is running on.
 - If the Tivoli Enterprise Portal Server is running on a Windows system, perform the following procedure:
 - a. Stop the Tivoli Enterprise Portal Server if it is running.
 - b. In the Manage Tivoli Enterprise Monitoring Services, right-click the Tivoli Enterprise Portal Server, and then click **Advanced > Edit Variables** from the displayed menu. The Tivoli Enterprise Portal Server Override Local Variable Setting window is displayed.
 - c. Click **Add**. The Add Environment Setting Override window is displayed.
 - d. In the **Variable** menu look for the **KFW_TOPOLOGY_CLUSTER_LIST** parameter. If it exists, append *AFF_MQSI_AGENT AFF_MQSI_BROKER* to any existing values, separated by a space. If it does not already exist, create it and set its value to *AFF_MQSI_AGENT AFF_MQSI_BROKER*.
 - e. Reconfigure and recycle the portal server.
 - If the Tivoli Enterprise Portal Server is running on a UNIX or Linux system, perform the following procedure:
 - a. Stop the Tivoli Enterprise Portal Server if it is running.
 - b. Navigate to the *ITM_HOME/config/* directory, where *ITM_HOME* is the IBM Tivoli Monitoring installation directory.
 - c. Open the *cq.ini* file in a text editor.
 - d. Look for the **KFW_TOPOLOGY_CLUSTER_LIST** parameter. If it already exists, add *AFF_MQSI_AGENT AFF_MQSI_BROKER* to any existing values, separated by a space. If it does not already exist, add the following line to the file:
KFW_TOPOLOGY_CLUSTER_LIST=AFF_MQSI_AGENT AFF_MQSI_BROKER
 - e. Save and close the file.
 - f. Reconfigure and recycle the portal server.

Creating a file that is used to start the agent

Important: When writing a startup script, ensure that the DB2 database is started first, followed by the monitored message broker, and finally the WebSphere Message Broker Monitoring agent.

Create a file containing the startup script that is used to start the agent by performing the following steps:

1. Create a new text file.
2. Enter the following lines to start the DB2 instance, where *db2_instance* is the name of the DB2 database instance.

```
su - db2_instance -c ". db2_instance/sqlllib/db2profile; db2start"
```

3. Enter the following lines to start the queue manager that is used by the message broker, where *MC91_install* is the directory in which WebSphere MQ supportpac MC91 is installed, and *QM_name* is the name of the queue manager used by the message broker.

```
MC91_install/bin/hamqm_start QM_name
```

4. Enter the following lines to start the monitored message broker, where *IC91_install* is the directory in which WebSphere Message Broker supportpac IC91 is installed, *broker_name* is the name of the monitored message broker

```
IC91_install/hamqsi_start_broker broker_name
```

5. Enter the following lines to start the WebSphere Message Broker Monitoring agent, where *ITM_HOME* is the directory in which IBM Tivoli Monitoring is installed, and *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance.

```
ITM_HOME/bin/itmcmd agent -o instance_name start qi
```

6. Save the file as *kqi_start.sh*.

Creating a file that is used to stop the agent

Important: When writing a shutdown script, ensure that the WebSphere Message Broker Monitoring agent is stopped first, followed by the monitored message broker, and finally the DB2 database.

Create a file containing the shutdown script used to stop the agent by performing the following steps:

1. Create a new text file.
2. Enter the following lines to stop the WebSphere Message Broker Monitoring agent, where *ITM_HOME* is the directory in which IBM Tivoli Monitoring is installed, and *instance_name* is the name of the WebSphere Message Broker Monitoring agent instance.

```
ITM_HOME/bin/itmcmd agent -o instance_name stop qi
```

3. Enter the following lines to stop the monitored message broker, where *IC91_install* is the directory in which WebSphere Message Broker supportpac IC91 is installed, *broker_name* is the name of the monitored message broker

```
IC91_install/bin/hamqsi_stop_broker broker_name 5
```

4. Enter the following lines to stop the queue manager that is used by the message broker, where *MC91_install* is the directory in which WebSphere MQ supportpac MC91 is installed, and *QM_name* is the name of the queue manager used by the message broker.

```
MC91_install/bin/hamqm_start QM_name
```

5. Enter the following lines to stop the DB2 instance, where *db2_instance* is the name of the DB2 database instance.

```
su - db2_instance -c ". db2_instance/sqlllib/db2profile; db2stop"
```

6. Save the file as *kqi_stop.sh*.

Setting the scripts that are used to start and stop the agent in an HACMP environment

Perform the following procedure to set the scripts used to start and stop the agent in an HACMP environment:

1. Open the cluster group in the HACMP cluster software.
2. Under Application Server set the start script as *kqi_start.sh*.

3. Under Application Server set the stop script as `kqi_stop.sh`.

Chapter 10. Reference

This section presents the reference topics for WebSphere Message Broker Monitoring agent.

Attributes

This section lists the attribute groups that are monitored by the WebSphere Message Broker Monitoring agent.

After you enable historical data collection for the eligible attribute groups, you can view historical data of these attribute groups. For instructions on how to enable and view historical data, see Chapter 6, “Collecting historical data,” on page 91.

For WebSphere Message Broker V7.0 or later

If the broker to be monitored is WebSphere Message Broker V7.0 or later, the monitoring agent can monitor the attribute groups that are listed in this section.

Accounting Message Flow Statistics attributes

The Accounting Message Flow Statistics attributes show the message flow accounting and statistical data gathered by the brokers (version 7 and later). The data might be gathered as snapshots, collected over short intervals, or as archival data, collected over longer intervals. Use the Message Flow Accounting attributes to create situations for detecting abnormal message flow accounting data, such as high Average CPU time and high Average Input Message Size.

Accounting Origin User-assigned identifier associated with a set of message flows which share a common origin.

Aggregation Date & Time The aggregation date and time of the collection interval. This attribute can be used to filter data for statistics. The value is the same as End Date & Time.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Avg CPU Microseconds Average CPU time in microseconds spent processing input messages. Valid format is an integer. This number is negative if overflow occurs.

Avg Elapsed Microseconds Average elapsed time in microseconds spent processing input messages. Valid format is an integer. This number is negative if overflow occurs.

Avg Input Message Size Average size of input messages in bytes. Valid format is an integer. This number is negative if overflow occurs.

Broker Name of the broker in alphanumeric format. This attribute is encoded in UTF8.

Broker UUID String associated with a broker that uniquely identifies it in alphanumeric format. This attribute is provided to assist in problem determination when reviewing trace and event log data. Valid format is an alphanumeric string.

End Date & Time The end date and time of the collection interval. This attribute is designed for logging and reporting data collection times rather than for creating situations. To specify a time of day for monitoring, use attributes from the Universal Time or Local Time groups.

Execution Group Name of the execution group which the message flow belongs to. Valid format is an alphanumeric string. This attribute is encoded in UTF8.

Execution Group UUID String that uniquely identifies the execution group. This identifier is provided to assist in problem determination when reviewing trace and event log data. Valid format is an alphanumeric string.

Input Byte Rate Rate in bytes per second, to two decimal places, at which the message flow processed input message bytes during the sampling period.

Input Msg Rate Rate in messages per second, to two decimal places, at which the message flow processed input messages during the sampling period.

Input Msg Wait CPU Microseconds Total CPU time in microseconds spent waiting for input messages. Valid format is an integer.

Input Msg Wait Elapsed Microseconds Total elapsed time in microseconds spent waiting for input messages. Valid format is an integer.

Library Name of the library that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Max CPU Microseconds Maximum CPU time in microseconds spent processing an input message. Valid format is an integer.

Max Elapsed Microseconds Maximum elapsed time in microseconds spent processing an input message, as an integer value.

Max Input Msg Size Maximum size of input messages in bytes. Valid format is an integer.

Message Flow Name of the message flow. This attribute is encoded in UTF8.

Min CPU Microseconds Minimum CPU time in microseconds spent processing an input message, as an integer value.

Min Elapsed Microseconds Minimum elapsed time in microseconds spent processing an input message, as an integer value.

Min Input Msg Size Minimum message size in bytes as an integer value.

Node Statistics Count Number of node statistics rows with this message flow sample.

Origin Node Name of the managed system that is the source of the data for the current row.

Query Type Type of query used in queries to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0

- Recent = 0
- Historical (not used, but set in historical data) = 1

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy
- StatsSettingsModified
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts (in different tables) as part of the same message flow accounting statistic sample.

Start Date & Time Start date and time of the sample interval.

Threads in Pool Number of threads in the pool.

Thread Stat Count Number of thread statistics rows associated with this message flow sample.

Times Max Threads Reached Number of times the maximum number of threads is reached.

Total Aggregate Wait Timeouts Number of transaction timeouts that occur when processing a message (for AggregateReply nodes only).

Total Backouts Number of transaction backouts that occur when processing a message.

Total Commits Number of transaction commits that occur when processing a message.

Total CPU Microseconds Total CPU time in microseconds spent processing input messages, as an integer value.

Total Elapsed Microseconds Total elapsed time in microseconds spent processing input messages, as an integer value.

Total Input Msgs Total number of messages that are processed.

Total Input Msg Size Total size of input messages in bytes, as an integer value.

Total MQ Errors Number of MQGET errors for MQInput nodes or Web Services errors for HTTPInput nodes.

Total Msg Errors Number of messages that contain errors.

Total Processing Errors Number of errors that occur when processing a message.

Accounting Node Statistics attributes

The Accounting Node Statistics attributes provide statistics pertaining to nodes used in message flows of WebSphere Message Broker (version 7 and higher). The data might be collected as snapshots (collected over short intervals) or as archival data (collected over longer intervals). Use the Accounting Node Statistics attributes to create situations for detecting concerned values pertaining to nodes in message flows.

Accounting Origin User-assigned identifier associated with a set of message flows sharing a common origin.

Aggregation Date & Time The aggregation date and time of the collection interval. This attribute can be used to filter data for statistics. The value is the same as End Date & Time.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Avg CPU Microseconds Average CPU time in microseconds spent processing input messages, as an integer. This number is negative if an overflow occurs.

Avg Elapsed Microseconds Average elapsed time in microseconds spent processing input messages as an integer. This number is negative if an overflow occurs.

Broker Name of the broker in alphanumeric format. This attribute is encoded in UTF8.

End Date & Time The end date and time of the collection interval.

This attribute is designed for logging and reporting data collection times rather than for creating situations. To specify a time of day for monitoring, use attributes from the Universal Time or Local Time groups.

Execution Group Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string. This attribute is encoded in UTF8.

Input Terminals Number of input terminals.

Invocations Total number of messages processed by this node.

Invocation Rate Rate per second, to two decimal places, at which the node is invoked during the sampling period.

Library Name of the library that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Max CPU Microseconds Maximum CPU time in microseconds spent processing an input message. The valid format is an integer value.

Max Elapsed Microseconds Maximum elapsed time in microseconds spent processing an input message as an integer value.

Message Flow Name of the message flow. This attribute is encoded in UTF8.

Min CPU Microseconds Minimum CPU time in microseconds spent processing an input message as an integer value.

Min Elapsed Microseconds Minimum elapsed time in microseconds spent processing an input message as an integer value.

Node Label Name of the node. This attribute is encoded in UTF8.

Node Type The type of the node.

Origin Node Name of the managed system that is the source of the data for the current row.

Output Terminals Number of output terminals.

Queue Manager Name The name of the queue manager associated with the node. This attribute only contains data if the queue manager name is specified in the node attributes, such as for MQInput and MQOutput nodes.

Queue Name The name of the queue associated with the node. This attribute only contains data if the queue name is specified in the node attributes, such as for MQInput and MQOutput nodes.

Query Type Type of query used in queries to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy
- StatsSettingsMethod
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts (in different tables) as part of the same message flow accounting statistic sample.

Start Date & Time Start date and time of the sampling interval.

Total CPU Microseconds Total CPU time in microseconds spent processing input messages, as an integer value.

Total Elapsed Microseconds Total elapsed time in microseconds spent processing input messages, as an integer value.

Accounting Terminal Statistics attributes

The Accounting Terminal Statistics attributes provide statistics pertaining to terminals on nodes that are used in message flows of WebSphere Message Broker

(version 7.0 or later). The data might be collected as snapshots (collected over short intervals) or as archival data (collected over longer intervals). Use the Accounting Terminal Statistics attributes to create situations for detecting concerned values of terminals.

Accounting Origin User-assigned identifier associated with a set of message flows that share a common origin.

Aggregation Date & Time The aggregation date and time of the collection interval. This attribute can be used to filter data for statistics. The value is the same as End Date & Time.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Broker Name of the broker in alphanumeric format. This attribute is encoded in UTF8.

End Date & Time The end date and time of the collection interval.

Execution Group Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string. This attribute is encoded in UTF8.

Invocations Total number of messages processed by this node.

Invocation Rate Rate per second, to two decimal places, at which the terminal is invoked during the sampling period.

Library Name of the library that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Message Flow Name of the message flow. This attribute is encoded in UTF8.

Node Label External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. The node label is encoded in UTF8.

Origin Node Name of the managed system, in this case, the broker subnode name.

Query Type Type of query used to get the data required. This attribute is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy

- StatsSettingsMethod
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts (in different tables) as part of the same message flow accounting statistic sample.

Start Date & Time Start date and time of the sampling interval.

Terminal Label Name of the terminal. This attribute is encoded in UTF8.

Terminal Type Type of terminal, either Input or Output.

Accounting Thread Statistics attributes

The Accounting Thread Statistics attributes provide the statistics gathered by WebSphere Message Broker (version 7 or later) for threads used by message flows. The data might be collected as snapshots (collected over short intervals) or as archival data (collected over longer intervals). Use the Accounting Thread Statistics attributes to create situations for detecting concerned values in threads.

Accounting Origin User-assigned identifier associated with a set of message flows that share a common origin.

Aggregation Date & Time The aggregation date and time of the collection interval. This attribute can be used to filter data for statistics. The value is the same as End Date & Time.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Avg CPU Microseconds Average CPU time, in microseconds, which is spent processing input messages, as an integer. This number is negative if overflow occurs.

Avg Elapsed Microseconds Average elapsed time, in microseconds, spent processing input messages as an integer. This number is negative if overflow occurs.

Avg Input Msg Size Average size, in bytes, of input messages as an integer value. This number is negative if overflow occurs.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

End Date & Time The end date and time of the collection interval.

This attribute is designed for logging and reporting data collection times rather than for creating situations. To specify a time of day for monitoring, use attributes from the Universal Time or Local Time groups.

Execution Group Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string. This attribute is encoded in UTF8.

| **Input Byte Rate** Rate in bytes per second, to two decimal places, at which the
| thread processed input message bytes during the sampling period.

| **Input Msg Rate** Rate in messages per second, to two decimal places, at which the
| thread processed input messages during the sampling period.

| **Input Msg Wait CPU Microseconds** Total CPU time in microseconds spent waiting
| for input messages as an integer value.

| **Input Msg Wait Elapsed Microseconds** Total elapsed time in microseconds waiting
| for input messages in integer format.

| **Library** Name of the library that contains the message flow. The valid format is an
| alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

| **Max Input Msg Size** Maximum message size in bytes in integer format.

| **Message Flow** Name of the message flow. This attribute is encoded in UTF8.

| **Min Input Msg Size** Minimum message size in bytes as an integer value.

| **Origin Node** Name of the managed system. In this case, it is the broker subnode
| name.

| **Query Type** Type of query used to get the data required. This attribute is not
| displayed by default. It has the following possible values:

- | • Current = 0
- | • Recent = 1
- | • Historical (not used, but set in historical data) = 2

| **Record Code** Reason for the output. This attribute has the following possible
| values:

- | • MajorInterval
- | • Snapshot
- | • Shutdown
- | • ReDeploy
- | • StatsSettingsMethod
- | • Stop

| **Record Type** Type of output, either Archive or Snapshot.

| **Sample Handle** Handle that identifies all parts as part of the same message flow
| accounting statistic sample.

| **Start Date & Time** Start date and time of the sampling interval.

| **Thread Number** Relative thread number in pool.

| **Total CPU Microseconds** Total CPU time in microseconds spent processing input
| messages, as an integer value.

| **Total Elapsed Microseconds** Total elapsed time in microseconds spent processing
| input messages, as an integer value.

Total Input Msgs Total number of messages processed. The valid format is an integer.

Total Input Msg Size Total size in bytes of input messages, as an integer value.

Broker Status attributes

The Broker Status attributes provide the activity status of the broker, name of the associated queue manager of the broker, and indicate whether the agent can connect to the queue manager of the broker. Use the Broker Status attributes to create situations for detecting changes in broker status.

ASID On z/OS systems, this attribute indicates the address space identifier (ASID) of the broker job in hexadecimal notation. This attribute is not applicable on distributed systems.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Broker Status State of the broker. This attribute can take the following values:

- n/a
- Started
- Stopped

Component Type Type of the component. This attribute can take the following values:

- n/a
- MessageBroker
- EventBroker
- ConfigurationManager
- UserNameServer
- ToolkitStudio

Configuration Delay Timeout Maximum time in seconds to allow a broker to process a deployed message.

Configuration Timeout Maximum time in seconds to allow a broker to process a deployed message of minimum size.

Converters Path to search for International Components for Unicode (ICU) converters or converter packages.

Execution Groups Number of execution groups that are running on the broker.

FAD Level The level of format and data of current component. If two versions of the component have the same FAD level, all the database tables, command-line arguments, and some other aspects must be compatible.

Http Listener Port Port that the broker HTTP listener uses.

Http Listener Trace Level Trace level of broker HTTP listener. This attribute can take the following values:

- none - no trace

- normal - normal trace
- debug - detailed debug trace

Http Listener Trace Size Trace size of the broker HTTP listener.

Jobname Taskname On z/OS systems, this attribute indicates the name of the job; on distributed systems, it indicates the name of the task.

Lil Path Where LIL (loadable implementation libraries) files are loaded from. This attribute is encoded in UTF8. It is shown as blank if there is no value on the system for a given broker registry string attribute.

Migration Needed Whether migration is requested from an existing Publish/Subscribe broker when the current broker is being created. This attribute can take the following values:

- n/a
- Yes (indicates that migration has been requested)
- No (indicates that migration has not been requested)

Origin Node Name of the managed system.

Platform Name of the host platform on which the broker is running.

Platform Version Version of the host platform on which the broker is running.

Process ID Process identifier for the administrative agent within the broker.

QMgr Connect Status State of the connection between the monitoring agent and the queue manager. This attribute has the following possible values:

- n/a
- Connected
- NotConnected

Queue Manager Name of the queue manager associated with the broker. Several WebSphere Business Integration components require a queue manager to be associated with them as all communication is accomplished using WebSphere MQ.

Release Version and release of the broker.

Start Date & Time On z/OS and UNIX systems, this attribute shows the job start date and time for the broker; this attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Started Msg Flows Number of message flows within a broker that have a status of started.

Statistics Archive Interval Indicates length of the interval (in minutes) for which archive statistics are produced if turned on for the broker.

Stopped Msg Flows Number of message flows within a broker that have a status of stopped.

System Name Name of the host system on which the broker is running.

Total Msg Flows Number of message flows in a broker.

Trusted QMgr Whether the association between this broker and its queue manager is set as a trusted relationship. Trusted status causes the broker to connect to its queue manager using fast-path binding. Otherwise, typical connections are used. This attribute has the following possible values:

- n/a
- Yes
- No

UUID The UUID string associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Broker Status Events attributes

The Broker Status Events attributes gather broker events as the events occur as pure events for situation use. Some events are retained for regular report queries as they are set in the agent parameter file. Use the Broker Status Events attributes to create situations for detecting concerned broker events.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing within WebSphere Business Integration brokers is performed at the broker level. This attribute is encoded in UTF8.

Event The action that triggers the event. This attribute has the following possible values:

- Unknown = 0
- Create = 1
- Delete = 2
- Change = 3
- MultipleUpdates = 4
- Start = 5
- Stop = 6
- InsideFlow-Unknown = 20
- InsideFlow-Create = 21
- InsideFlow-Delete = 22
- InsideFlow-Change = 23
- InsideFlow-Updates = 24

Event Date & Time The put date and time for the publication of the current event, which does not necessarily reflect the event time for a retained publication, such as a Status Start event for the broker.

Event Type Type of event. This attribute can take the following values:

- Configuration
- Status

Event Qualifier 1 The first qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Event Qualifier 2 The second qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Origin Node The name of the managed system that is the data source for the current row.

Qualifier 1 Description A short description of the item that Qualifier 1 refers to. This attribute can take the following values:

- n/a = 0
- ExecutionGroup = 1
- Broker = 2
- Neighbor = 3
- AllNeighbors = 4
- Topic = 5
- AllTopics = 6
- MultipleTopics = 12

Qualifier 2 Description A short description of the item that Qualifier 2 refers to. This attribute can take the following values:

- n/a = 0
- MessageFlow = 7
- MultipleMessageFlows = 8
- AllMessageFlows = 9
- ExpiryTimeStamp = 10
- StateCloned = 11
- Application = 12
- MultipleApplications = 13
- Library = 14
- MultipleLibraries = 15

Broker Topology attributes

The Broker Topology attribute group contains the attributes that are used in drawing high-level objects in the broker topology view. The broker topology attributes are related to brokers, execution groups, and message flows. The information in this table can be used only in workspace queries, and cannot be used as conditions in situations. The data in this attribute group is intended only for drawing the topology view in the Tivoli Enterprise Portal.

From Object Name The name of the parent object in the object hierarchy. This value is a UTF-8 string of up to 256 alphanumeric characters. Possible values are listed in Table 14.

Table 14. From Object Name values

Type of Object	From Object Name value
Broker	The name of the host on which the broker runs or no value
Execution group	Name of the broker to which the execution group belongs
Message flow	Name of the execution group to which the message flow belongs

Object UUID The UUID of the object. This value is a UTF-8 string of up to 36 alphanumeric characters.

Object Name The name of the object. This value is a UTF-8 string of up to 256 alphanumeric characters, and is the name of a broker, execution group or message flow.

Object Type The type of the object. This attribute can take the following values:

- kqi.BrokerNode
- kqi.ExecutionGroupNode
- kqi.MessageFlowNode.

Object Status The status of the object. Possible values are listed in Table 15.

Table 15. Object Status values

Value	Meaning	Notes
0	Stopped	The status of an execution group with no user-defined message flows is Stopped. The status of ConfigurationMessageFlow and PubSubControlMsgFlow system message flows is not considered when determining the status of an execution group.
1	Started	The status of ConfigurationMessageFlow and PubSubControlMsgFlow message flows is always started and is not used when determining this value.
2	Partially started	This value applies only to execution groups and indicates that only a part of the execution group's message flows are started. The status of ConfigurationMessageFlow and PubSubControlMsgFlow system message flows is not used when determining the status of an execution group.

Origin Node The name of the managed system.

To Object UUID The UUID of the next object in the object hierarchy. This value is a UTF-8 string of up to 36 alphanumeric characters.

Components attributes

The Components attributes provide a list of broker product components and their state at agent level. Use the Components attributes to create situations for detecting the state change of components.

ASID On z/OS systems, it indicates the address space identifier (ASID) of the component job in hexadecimal notation; this attribute is not applicable on distributed platforms.

Component (Deprecated) Name of the broker product component. This name is either one of the following constant names, or the name given to a broker component created on the monitored host system:

- ConfigMgr
- UserNameServer
- ControlCenter

Component Name of the broker product component. This attribute is encoded in UTF8. This name is either one of the following constant names, or the name given to a broker component created on the monitored host system:

- ConfigMgr
- UserNameServer
- ControlCenter

Component Type Type of the component. This attribute has the following possible values:

- N/A
- MessageBroker
- EventBroker
- ConfigurationManager
- UserNameServer
- ToolkitStudio

Remember: ToolkitStudio indicates either Message Brokers Toolkit V5 or Control Center V2.1.

Jobname Taskname On z/OS systems, it indicates the name of the job; on distributed systems, it indicates the name of the task.

Origin Node Name of the managed system.

Platform Name of the host platform on which the component is running.

Platform Version Version of the host platform on which the component is running.

Process ID Process identifier for the administrative agent within the broker or other component.

Queue Manager Name of the queue manager associated with the broker or other component. Several components require a queue manager to be associated with them because all communication is accomplished using WebSphere MQ.

Release Version and release of the broker product components.

Start Date & Time On z/OS and UNIX systems, it shows the job start date and time for the component; this attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Status State of the component. This attribute has the following possible values:

- n/a
- Started
- Stopped
- Installed

System Name Name of the host system on which the component is running.

User Name Server Queue QMgr Name of the queue manager associated with the User Name Server that has been defined to serve access control requests for the

message broker. Note that if access control is not used, this field might be blank because no User Name Server is serving the broker.

Execution Group Status attributes

The Execution Group Status attribute group shows the execution groups that are defined for the broker, and the definition information for each execution group.

Broker Identifies the broker that is the data source for the current row. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Event Log File Mode Indicates how often log entries are saved to the event log. This attribute has the following possible values:

- Safe (for frequent writing)
- Fast (for infrequent writing)

Event Log File Size Maximum size of the event log file in bytes. For Windows NT systems, this value is 0, because the event log of Windows NT operating system is used instead. Event records within the file display significant broker operations such as start or stop.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Started Msg Flows Number of message flows within an execution group that have a status of started.

Status State of the execution group. This attribute has the following possible values:

- n/a
- Started
- Stopped

Stopped Msg Flows Number of message flows within an execution group that have a status of stopped.

Total Msg Flows Number of message flows in an execution group.

Trace Level The level of trace that is currently active. This attribute has the following possible values:

- None
- Normal
- Debug

Trace Log File Mode Indicates how often log entries are saved to the trace log. This attribute has the following possible values:

- Safe (for frequent writing)
- Fast (for infrequent writing)

Trace Log File Size Maximum size of the trace log file in bytes. Trace records display data that is used by IBM technical-support representatives to identify problems with the product.

User Trace Level Level of user trace that is currently active. This attribute has the following possible values:

- None
- Normal
- Debug

User Trace Log File Mode Indicates how often log entries are saved to the user trace log. This attribute has the following possible values:

- Safe (for frequent writing)
- Fast (for infrequent writing)

User Trace Log File Size Maximum size of the user trace log file in bytes. User trace records display the data that covers the overall execution of a message flow for such items as nodes executed.

UUID The UUID string associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

File Resource Statistics attributes

After resource statistics collection is started, use the File Resource Statistics attributes to get the statistics information for the local file system of any file actions done by any type of file node.

Aggregation Date & Time The aggregation date and time of the collection interval. The value is the same as End Date & Time.

Broker The name of the monitored broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Bytes Read The total number of bytes read by either a file input node, FTE input node, or a file read node.

Bytes Written The total number of bytes written by a file output node or FTE output node.

End Date & Time The end date and time of the collection interval.

Execution Group The name of an execution group. This attribute is encoded in UTF8.

Files Created The number of files successfully created by a file output node.

Files Read The number of files successfully read by any file node.

Origin Node The managed system name of the monitored broker.

Records Read The number of records read by either a file input node, FTE input node, or a file read node.

Records Written The number of records written to files by a file output node or FTE output node.

Resource Identifier The identifier of the file resource.

Resource Type The type of the resource. For File resource, this attribute is always File.

Start Date & Time The start date and time of the sampling interval.

JDBC Connection Pools Resource Statistics attributes

After resource statistics collection is started, use the JDBC Connection Pools Resource Statistics attributes to review statistics information for each JDBC Provider configurable service.

Actual Pool Size A snapshot of the number of connections that are currently in the pool when the statistics are reported.

Aggregation Date & Time The aggregation date and time of the collection interval. The value is the same as End Date & Time.

Broker The name of the monitored broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Cumulative Delayed Requests The number of times that a request for a connection cannot be satisfied immediately, because the number of allocated connections has reached the maximum pool size and there are no connections currently available.

Cumulative Requests The number of requests that are received by the connection pool during this accounting period.

Cumulative Timed Out Requests The number of requests for connections that cannot be satisfied within 15 seconds.

End Date & Time The end date and time of the collection interval.

Execution Group The name of an execution group. This attribute is encoded in UTF8.

JDBC Provider The name of the JDBC Providers configurable service that is using connection pooling. If there are no connection pools started, this value is none.

Max Delay Milliseconds The maximum time that a caller has to wait for a connection to be allocated, in milliseconds.

Max Pool Size The maximum size of the connection pool.

Origin Node The managed system name of the monitored broker.

Resource Identifier The identifier of the JDBC Provider resource.

Resource Type The type of the resource. For JDBC Provider resource, this attribute is always JDBCConnectionPools.

Start Date & Time The start date and time of the sampling interval.

JVM Resource Statistics attributes

Each execution group starts its own Java Virtual Machine (JVM). After resource statistics collection is started, use the JVM Resource Statistics attributes to review how much memory is in use by the JVM resource, and how often garbage collection might occur within the execution group. Statistics are collected for the following JVM resources:

- Heap memory
- Non-heap memory
- Garbage collection

Aggregation Date & Time The aggregation date and time of the collection interval. The value is the same as End Date & Time.

Broker The name of the monitored broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Committed Memory MB The amount of memory that has been allocated to the JVM by the operating system.

Cumulative GC Collections The total number of garbage collections that have occurred for this instance of the JVM. Its value might be undefined.

Cumulative GC Seconds The accumulated garbage collection elapsed time in seconds for this instance of the JVM. Its value might be undefined.

End Date & Time The end date and time of the collection interval.

Execution Group The name of an execution group. This attribute is encoded in UTF8.

Initial Memory MB The initial amount of memory that the JVM requests from the operating system for memory management during startup.

Max Memory MB The maximum amount of memory that can be used for memory management. Its value might be undefined.

Origin Node The managed system name of the monitored broker.

Resource Identifier The identifier of the resource.

Resource Type The type of the resource. For JVM resource, this attribute is always JVM.

Start Date & Time The start date and time of the sampling interval.

Used Memory MB The amount of memory that is currently in use.

Message Flow Status attributes

The Message Flow Status attributes display defined message flows by execution group and definition information for each message flow. Use the Message Flow Information attributes to create situations for detecting concerned status of message flow.

Additional Instances Number of additional instances of the message flow that the execution group should run to allow for more multi-processing of messages.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Commit Count Number of messages that must be processed for commit processing to occur.

Commit Interval Time interval at which commit processing should occur.

Coordinated Transaction Indicates whether the message flow represents a coordinated transaction for commit/backout processing. This attribute has the following possible values:

- Yes
- No

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Library Name of the library that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Processing Nodes Number of processing nodes within a message flow.

Query UUID The UUID string associated with the object to indicate what object is used in this query.

Stats Archive Accounting Origin Identification value that indicates what the archive collection data is associated with. User can set this value in node ESQL statement:

```
SET Environment.Broker.Accounting.Origin = "value";
```

Stats Archive Node Data Level Indicates the archive collection of node statistics data for current message flow. This attribute has the following possible values:

- n/a
- None - archive collection contains no node related data
- Basic - archive collection contains node related data
- Advanced - archive collection contains node related and terminal related data

| **Stats Archive Output Format** Indicates the output destination for the statistics
| archive reports. This attribute has the following possible values:

- | • n/a
- | • Usertrace - statistics archive writes "bip" messages to user trace, which can be
| post-processed in the normal way using the **mqsi readlog** and **mqsi formatlog**
| commands
- | • XML - statistics archive reports are generated as XML documents
- | • SMF - (z/OS only) statistics archive reports are output as SMF type 117 records

| **Stats Archive Status** Indicates whether message flow accounting and statistics
| archive level collection is active for current message flow. This attribute has the
| following possible values:

- | • n/a
- | • Active - archive collection is on
- | • Inactive - archive collection is off

| **Stats Archive Thread Data Level** Indicates whether the archive collection contains
| thread statistics for current message flow. This attribute has the following possible
| values:

- | • n/a
- | • None - archive collection contains no thread related data
- | • Basic - archive collection contains thread related data

| **Stats Snapshot Accounting Origin** Identification value that indicates what the
| snapshot collection data is associated with. User can set this value in node ESQL
| statement:

| SET Environment.Broker.Accounting.Origin = "value";

| **Stats Snapshot Node Data Level** Indicates the snapshot collection of node statistics
| for current message flow. This attribute has the following possible values:

- | • n/a
- | • None - snapshot collection contains no node related data
- | • Basic - snapshot collection contains node related data
- | • Advanced - snapshot collection contains node related and terminal related data

| **Stats Snapshot Output Format** Indicates the output destination for the statistics
| snapshot reports. This attribute has the following possible values:

- | • n/a
- | • Usertrace - statistics snapshot writes "bip" messages to user trace, which can be
| post processed in the normal way using the **mqsi readlog** and **mqsi formatlog**
| commands
- | • XML - statistics snapshot reports are generated as XML documents
- | • SMF - (z/OS only) statistics snapshot reports are produced as SMF type 117
| records

| **Stats Snapshot Status** Indicates whether message flow accounting and statistics
| snapshot level collection is active for the current message flow. This attribute has
| the following possible values:

- | • n/a
- | • Active - snapshot collection is on
- | • Inactive - snapshot collection is off

Stats Snapshot Thread Data Level Indicates whether the snapshot collection contains thread statistics for the current message flow. This attribute has the following possible values:

- n/a
- None - snapshot collection contains no thread related data
- Basic - snapshot collection contains thread related data

Status State of the message flow. This attribute has the following possible values:

- n/a
- Started
- Stopped

Trace Level Level of trace that is currently active. The attribute has the following possible values:

- None
- Normal
- Debug

User Trace Level Level of user trace that is currently active. This attribute has the following possible values:

- None
- Normal
- Debug

UUID UUID string that is associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Message Flow Node Topology attributes

The Message Flow Node Topology attribute group contains attributes used in drawing low-level objects in the message flow topology view. The information in this table can only be used in workspace queries, and cannot be used as conditions in situations. The data in this attribute group is intended only for drawing the topology view in the Tivoli Enterprise Portal.

Attribute 1 The first attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Attribute 2 The second attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Connection Type The type of connection from this node to the next node in the hierarchy. This value is a UTF-8 string of up to 256 alphanumeric characters.

Execution Group The name of the execution group to which the message flow belongs. This value is a UTF-8 string of up to 256 alphanumeric characters.

Message Flow The name of the message flow. This value is a UTF-8 string of up to 256 alphanumeric characters.

Message Flow UUID UUID string that is associated with the message flow to uniquely identify the message flow. This attribute is available for brokers of version 8 or later.

Node Label The label of the node. This value is a UTF-8 string of up to 256 alphanumeric characters.

Node Name The name of the node. This value is a UTF-8 string of up to 256 alphanumeric characters.

Node Type The type of the node. This value is the same as UNODE_NAME for predefined nodes, and kqi.UserDefinedNode for user-defined nodes. This value is a UTF-8 string of up to 64 alphanumeric characters.

Node UUID The UUID of the node. Node UUID values might be much longer than the UUID values of other objects. This value is a UTF-8 string of up to 256 alphanumeric characters.

Origin Node The managed system name of the monitored broker.

To Node UUID The UUID of the next node in the hierarchy. Node UUID values might be much longer than the UUID values of other objects. This value is a UTF-8 string of up to 256 alphanumeric characters.

To Terminal The terminal identifier of the next node in the hierarchy. This value is a UTF-8 string of up to 256 alphanumeric characters.

Message Processing Nodes attributes

The Message Processing Nodes attributes present detailed information about the given message flow processing node. Use the Message Processing Nodes attributes to create situations for detecting concerned status of a message flow.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Attribute 1 The first attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Attribute 2 The second attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Attribute 1 Description Short description of which item Attribute 1 is referring to. This attribute has the following possible values:

- n/a
- QueueName
- MessageType
- Type
- ComputeExpression
- Statement
- FilterExpression
- TargetFormat

- ImplicitStreamNaming
- MessageNumber
- Destination
- LabelName
- GenQueueName
- Mode
- MQIpdpEnable
- DebugNodeID
- TraceFile
- AggregateName
- FolderName
- URLSpecifier
- IgnoreTransportFailures
- Port
- MappingExpression
- StylesheetName
- JavaClass
- UniqueIdentifier
- InitialContextFactory
- distributionMode
- eventCoordination
- dataSourceName
- inputDirectory
- outputDirectory
- selectedPort
- asyncRequestCorrelator
- createEnvelope
- envelopeDestination
- name
- adapterComponent
- fromAddress
- asyncResponseCorrelator
- connectionDetails
- hostname
- configurableService
- scriptName

Attribute 2 Description Short description of which item Attribute 2 is referring to. This attribute has the following possible values:

- n/a
- QueueManagerName
- MessageSet
- EventMessage
- DataSource
- SubscriptionPoint

- MessageText
- FilePath
- CollectQueueTime
- Set
- GenQMName
- MQIpdpPort
- OriginalFlowUuid
- MapName
- Trace
- SubFlowName
- TimeoutInterval
- UnsolicitedTimeout
- TimeoutForClient
- TimeoutForServer
- GenerateDefaultHttpHeaders
- Authentication
- StylesheetPath
- RequestPersistence
- OperationMode
- LocationJNDIBindings
- persistenceMode
- sqlQuery
- filenamePattern
- outputFilename
- wsdlFileName
- envelopeLocation
- pathMode
- namespace
- defaultMethod
- toAddresses
- messageSetProperty
- timeoutWaitingForData
- portNumber
- timeoutSendingData
- useEvaluateMethod

Broker Identifier of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Library Name of the library that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Message Flow Name of the message flow. Message flows are designed by the user within the Control Center to define how messages are to be processed. A message flow can consist of one or more message processing nodes. The message flow name is encoded in UTF8.

Message Flow UUID UUID string that is associated with the message flow to uniquely identify the message flow. This attribute is available for brokers of version 8 or later.

Node Label External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. The node label is encoded in UTF8.

Node Name Actual long name of the message processing node as it is defined within the Control Center or Message Brokers Toolkit. This name differs from the label name which the user provides for an instance of the node when including it within a message flow. This attribute is encoded in UTF8.

Origin Node The managed system name of the monitored broker.

UUID UUID string that is associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Monitor Node Base Statistics attributes

The Monitor Node Base Statistics attributes display the CandleMonitor nodes by execution group and message flow, and show statistics at the lowest collected level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Avg Flow Time The current average time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor node regardless of the node type.

Current Avg Msg Size The current average size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor node regardless of the node type.

Current Avg Msg Time The sum of Current Avg Queue Time and Current Avg Flow Time attribute values to give the average time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Avg Queue Time The current average time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is non-zero only for the nodes of input type, and is taken from

the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Current Msg Byte Rate The current rate of message bytes going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is determined for every CandleMonitor node regardless of the node type.

Current Msg Count The current number of messages going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is determined for every CandleMonitor node regardless of the node type.

Current Msg Rate The current rate of messages going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is determined for every CandleMonitor node regardless of the node type.

Execution Group Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

First Msg Date & Time Date and time of the first message that goes through this CandleMonitor node after monitoring begins or statistics are reset.

Last Msg Date & Time Date and time of the last message that goes through this CandleMonitor node.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Monitor Type The type attribute associated with this CandleMonitor node. This attribute has the following possible values:

- n/a
- Input
- Output
- subFlowInput
- subFlowOutput
- Other

Origin Node Name of the node that identifies the data source for the current row. The displayed name indicates either an agent node or broker subnode.

Overall Avg Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the entire monitoring

period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor node regardless of the node type.

Overall Avg Msg Size Overall average size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the average determined within the CandleMonitor node regardless of the node type.

Overall Avg Msg Time The sum of Overall Avg Queue Time and Overall Avg Flow Time attribute values to give the overall average time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Avg Queue Time Overall average time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is non-zero only for the nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Overall Max Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor node regardless of the node type.

Overall Max Msg Size Overall maximum size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximum determined within the CandleMonitor node regardless of the node type.

Overall Max Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is non-zero only for nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Overall Min Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor node regardless of the node type.

Overall Min Msg Size Overall minimum size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor node regardless of the node type.

Overall Min Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This

value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is non-zero only for the nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Overall Msg Byte Rate Overall rate of message bytes going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is determined by every CandleMonitor node regardless of the node type.

Overall Msg Count Overall number of messages going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is determined by every CandleMonitor node regardless of the node type.

Overall Msg Rate Overall rate of messages going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is determined by every CandleMonitor node regardless of the node type.

Processing Node Label The external name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. This attribute is encoded in UTF8.

Sample Date & Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

Sub-Flow Name Name given to the subflow by the message flow developer using the Control Center or Message Broker Toolkit. The name is assigned by setting the subFlowName attribute for a CandleMonitor node when placing it into a message flow. This attribute is encoded in UTF8.

Monitor Node Broker Statistics attributes

The Monitor Node Broker Statistics attributes provide message flow statistics summarized at the broker level. Use the Monitor Node Broker Statistics attributes to create situations for detecting the long message processing time in message flows at the broker level.

ASID On z/OS systems, this attribute indicates the address space identifier (ASID) of the broker job in hexadecimal notation. This attribute is not applicable on distributed platforms.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Avg Flow Time Current average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

| **Current Avg Msg Size** Current average size of messages (including headers) going
| through monitored message flows. This value reflects the activity over the latest
| collection interval for the monitoring agent. This value is taken from the averages
| determined within the CandleMonitor nodes that are placed in the message flows
| with a **type** attribute of input.

| **Current Avg Msg Time** Sum of Current Average Queue Time and Current Average
| Flow Time to give the average time in seconds (up to millisecond accuracy) that it
| takes to process a message over the latest collection interval.

| **Current Avg Queue Time** Current average time in seconds (up to millisecond
| accuracy) that a message remains in the input queue for a monitored message flow.
| This value reflects the activity over the latest collection interval for the monitoring
| agent. This value is taken from the averages determined within the CandleMonitor
| nodes that are placed in the message flows with a **type** attribute of input. Note
| that the value includes any processing time taken by the MQInput node.

| **Current Interval** The length of the sample interval (length of time between
| samples) in seconds. This attribute is the interval over which the delta calculations
| for other attributes that begin with "Current" are made - that is, the interval that
| they apply to.

| **Current Msg Byte Input Rate** Current rate of message bytes that are entered into
| monitored message flows. This value reflects the activity over the latest collection
| interval for the monitoring agent. Input message byte rates are determined by the
| CandleMonitor nodes that are placed in message flows with a **type** attribute of
| input.

| **Current Msg Byte Output Rate** Current rate of message bytes that are produced
| from monitored message flows. This value reflects the activity over the latest
| collection interval for the monitoring agent. Output message byte rates are
| determined by the CandleMonitor nodes that are placed in message flows with a
| **type** attribute of output.

| **Current Msg Input Count** Current number of messages that are entered into
| monitored message flows. This value reflects the activity over the latest collection
| interval for the monitoring agent. Input messages are counted by the
| CandleMonitor nodes that are placed in message flows with a **type** attribute of
| input.

| **Current Msg Input Rate** Current rate of messages that are entered into monitored
| message flows. This value reflects the activity over the latest collection interval for
| the monitoring agent. Input message rates are determined by the CandleMonitor
| nodes that are placed in message flows with a **type** attribute of input.

| **Current Msg Output Count** Current number of messages that are produced from
| monitored message flows. This value reflects activity over the latest collection
| interval for the monitoring agent. Output messages are counted by the
| CandleMonitor nodes that are placed in message flows with a **type** attribute of
| output.

| **Current Msg Output Rate** Current rate of messages that are produced from
| monitored message flows. This value reflects activity over the latest collection
| interval for the monitoring agent. Output message rates are determined by the
| CandleMonitor nodes that are placed in message flows with a **type** attribute of
| output.

First Msg Date & Time Date and time of the first message that goes through a monitored message flow after monitoring begins or statistics are reset.

Jobname Taskname On z/OS systems, this attribute indicates the name of the job; on distributed platforms, this attribute indicates the name of the task.

Last Msg Date & Time Date and time of the last message that goes through a monitored message flow.

Monitored Exec Grps Number of execution groups that have a CandleMonitor node included in one or more of their message flows.

Monitored Msg Flows Number of message flows in the broker that include a CandleMonitor node.

Origin Node The managed system name of the monitored broker.

Overall Avg Flow Time The overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Size The overall average size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Time The sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Avg Queue Time The overall average number of seconds (up to millisecond accuracy) that a message remained in a monitored message flow input queue since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Max Flow Time The overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Msg Size The overall maximum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Queue Time The overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Msg Byte Input Rate The overall rate of message bytes input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Byte Output Rate The overall rate of message bytes output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Input Count The overall number of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Input Rate The overall rate of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Output Count The overall number of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Output Rate The overall rate of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Min Flow Time The overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Msg Size The overall minimum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last

message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Queue Time The overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Platform Name of the host platform on which the component is running.

Platform Version Version of the host platform on which the component is running.

Process ID Process identifier for the administrative agent within the broker.

Sample Date & Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Start Date & Time On z/OS and UNIX systems, it shows the job start date and time for the component; this attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Started Msg Flows Number of message flows within the broker that have a status of started.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not for display.

Status State of the broker. This attribute has the following possible values:

- n/a
- Started
- Stopped

System Name Name of the host system on which the component is running.

Monitor Node Events attributes

The Monitor Node Events attributes gather message flow events based on parameters set by the user of the CandleMonitor node. These events are gathered as they occur, with some events being retained for regular report queries as set within the agent parameters file. Use the Monitor Node Events attributes to create situations for detecting concerned message flow events.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Correl ID Correlation identifier for the message being processed when the message flow event occurs. The correlation identifier is obtained from the WebSphere MQ MQMD.

Event Date & Time Time when the current event occurs.

| **Event Message** Message that describes the event. This attribute is encoded in
| UTF8.

| **Execution Group** Name of an execution group. An execution group is a process
| that runs within the broker and provides the execution environment in which
| message flows run. Users associate message flows with execution groups during
| message flow assignment in the Control Center or Message Brokers Toolkit. This
| attribute is encoded in UTF8.

| **Message Flow** Name of the message flow. Message flows are designed by the user
| to define how messages are to be processed. A message flow can consist of one or
| more message processing nodes. This attribute is encoded in UTF8.

| **Monitor Type** Type attribute associated with the CandleMonitor node. This
| attribute has the following possible values:

- | • n/a
- | • Input
- | • Output
- | • Other
- | • subFlowInput
- | • subFlowOutput

| **Msg ID** Message identifier that uniquely identifies which message is being
| processed when the message flow event occurs. The message identifier is obtained
| from the WebSphere MQ MQMD.

| **Node Label** External name given to an instance of the message processing node
| when it is included in a message flow. This name is set by the user to identify this
| message processing node. This attribute is encoded in UTF8.

| **Origin Node** Name of the managed system of the monitored broker.

| **Put Appl Name** Name of the put application from the WebSphere MQ MQMD for
| the message being processed. This is the name of the application that originally
| put the message as determined by WebSphere MQ.

| **Put Appl Type** Type of the put application from the WebSphere MQ MQMD for
| the message being processed. This is the type of the application that originally put
| the message as determined by WebSphere MQ. This attribute has the following
| possible values:

- | • Unknown
- | • NoContext
- | • CICS
- | • OS390
- | • IMS
- | • OS2
- | • DOS
- | • UNIX
- | • QMGR
- | • OS400
- | • WINDOWS

- CICS_VSE
- WINDOWS_NT
- VMS
- NSK
- VOS
- IMS_BRIDGE
- XCF
- CICS_BRIDGE
- NOTES_AGENT
- TPF
- USER
- BROKER
- JAVA
- DQM
- CHANNEL_INITIATOR
- WLM
- BATCH
- RRS_BATCH
- SIB
- SYSTEM
- USER_FIRST
- USER_LAST

Put Date & Time Put date and time from the WebSphere MQ MQMD for the message being processed. This is the date and time that the message is originally put as determined by WebSphere MQ.

Thread ID Thread identifier that uniquely identifies which thread processor is active when the message flow event occurs.

Monitor Node Execution Group Statistics attributes

The Monitor Node Execution Group Statistics attributes provide message flow statistics summarized at the execution group level. Use the Monitor Node Execution Group Statistics attributes to create situations for detecting long message processing time in message flows at execution group level.

ASID On z/OS systems, it indicates the address space identifier (ASID) of the execution group job in hexadecimal notation. This attribute is not applicable on distributed systems.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Avg Flow Time Current average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with the **type** attribute of input.

Current Avg Msg Size Current average size of messages (including headers) going through monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with the **type** attribute of input.

Current Avg Msg Time Sum of Current Average Queue Time and Current Average Flow Time to give the average time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Avg Queue Time Current average time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with the **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Current Msg Byte Input Rate Current rate of message bytes input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a type attribute of input.

Current Msg Byte Output Rate Current rate of message bytes output from monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a type attribute of output.

Current Msg Input Count Current number of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input messages are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Msg Input Rate Current rate of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a type attribute of input.

Current Msg Output Count Current number of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output messages are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Current Msg Output Rate Current rate of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output message rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which

message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

First Msg Date & Time Date and time of the first message that goes through a monitored message flow after monitoring begins or statistics are reset.

Jobname Taskname On z/OS systems, it indicates the name of the job; on distributed systems, it indicates the name of the task.

Last Msg Date & Time Date and time of the last message that goes through a monitored message flow.

Monitored Msg Flows Number of message flows in an execution group that include a CandleMonitor node.

Origin Node Name of the managed system.

Overall Avg Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input.

Overall Avg Msg Size Overall average size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Time Sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average total time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Avg Queue Time Overall average time in seconds (up to millisecond accuracy) that a message remained in a monitored message flow input queue since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Max Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input.

Overall Max Msg Size Overall maximum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last

message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Min Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Msg Size Overall minimum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Msg Byte Input Rate Overall rate of message bytes input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Byte Output Rate Overall rate of message bytes output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Input Count Overall number of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Input Rate Overall rate of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input

message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Output Count Overall number of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Output Rate Overall rate of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Platform Name of the host platform on which the execution group is running.

Platform Version Version of the host platform on which the execution group is running.

Process ID Process identifier for the execution group process.

Sample Date & Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Start Date & Time On z/OS and UNIX systems, it shows the job start date and time for the execution group. This attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Started Msg Flows Number of message flows within an execution group that have a status of started.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

Status State of the execution group. This attribute has the following possible values:

- n/a
- Started
- Stopped

System Name Name of the host system where the execution group is running.

Monitor Node Message Flow Statistics attributes

The Monitor Node Message Flow Statistics attributes provide message flow statistics summarized at the message flow level. Use the Message Flow Statistics attributes to create situations for detecting long processing time in message flows at the message flow level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

| **Current Avg Flow Time** Current average flow time in seconds (up to millisecond accuracy) that it takes to process a message in the message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the average determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

| **Current Avg Msg Size** Current average size of messages (including headers) going through monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

| **Current Avg Msg Time** Sum of Current Average Queue Time and Current Average Flow Time to give the average total time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

| **Current Avg Queue Time** Current average time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

| **Current Interval** Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

| **Current Msg Byte Input Rate** Current rate of message bytes input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

| **Current Msg Byte Output Rate** Current rate of message bytes output from monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

| **Current Msg Input Count** Current number of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

| **Current Msg Input Rate** Current rate of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

| **Current Msg Output Count** Current number of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

| **Current Msg Output Rate** Current rate of messages output from monitored message flows. This value reflects activity over the latest collection interval for the

monitoring agent. Output message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Execution Group Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

First Msg Date & Time Date and time of the first message that goes through a monitored message flow after monitoring begins or statistics are reset.

Last Msg Date & Time Date and time of the last message that goes through a monitored message flow.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Monitor Nodes Number of CandleMonitor nodes within a message flow.

Origin Node Name of the managed system that is the source of the data for the current row.

Overall Avg Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Size Overall average size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Time Sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average total time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Avg Queue Time Overall average time in seconds (up to millisecond accuracy) that a message remained in a monitored message flow input queue since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Max Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Msg Size Overall maximum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Min Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Msg Size Overall minimum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Msg Byte Input Rate Overall rate of message bytes input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Byte Output Rate Overall rate of message bytes output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Input Count Overall number of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

| **Overall Msg Input Rate** Overall rate of messages input to monitored message
| flows since the reset of statistics. This value reflects the activity over the entire
| monitoring period given by the first and last message date and times. Input
| message rates are determined by the CandleMonitor nodes that are placed in
| message flows with a **type** attribute of input.

| **Overall Msg Output Count** Overall number of messages output from monitored
| message flows since the reset of statistics. This value reflects the activity over the
| entire monitoring period given by the first and last message date and times.
| Output messages are counted by the CandleMonitor nodes that are placed in
| message flows with a **type** attribute of output.

| **Overall Msg Output Rate** Overall rate of messages output from monitored
| message flows since the reset of statistics. This value reflects the activity over the
| entire monitoring period given by the first and last message date and times.
| Output messages are counted by the CandleMonitor nodes that are placed in
| message flows with a **type** attribute of output.

| **Sample Date & Time** Date and time of the latest sample taken of CandleMonitor
| node data for producing the displayed statistics.

| **Statistics User** Encrypted logon ID for user statistics queries. This attribute is not
| displayed.

| **Status** State of the message flow. This attribute has the following possible values:

- | • n/a
- | • Started
- | • Stopped

| **Monitor Node Sub-Flow Statistics attributes**

| The Monitor Node Sub-Flow Statistics attribute group provides data for only that
| section of the message flow that is the part of a subflow. Monitor Node Sub-Flow
| Statistics attributes are always provided with respect to the particular message
| flow within an execution group to which the subflow is a part. Use the Monitor
| Node Sub-Flow Statistics attributes to create situations for detecting concerned
| values in subflows.

| **Broker** Name of the broker. The WebSphere Message Broker Monitoring agent
| targets specific brokers for monitoring because all message processing is performed
| at the broker level. This attribute is encoded in UTF8.

| **Current Avg Flow Time** Current average in seconds (up to millisecond accuracy)
| that it takes to process a message in the subflow. This value reflects the activity
| over the latest collection interval for the monitoring agent. This value is taken from
| the average determined within the CandleMonitor nodes that are placed in the
| message flows to delineate the subflows.

| **Current Avg Msg Size** Current average size of messages (including headers) going
| through monitored subflows. This value reflects the activity over the latest
| collection interval for the monitoring agent. This value is taken from the averages
| determined within the CandleMonitor nodes that are placed in the message flows
| to delineate the subflows.

| **Current Avg Msg Time** Sum of Current Avg Queue Time and Current Avg Flow
| Time attribute values to give the average total time in seconds (up to millisecond
| accuracy) that it takes to process a message over the latest collection interval.

| **Current Avg Queue Time** Current average time in seconds (up to millisecond
| accuracy) that a message remains in the input queue for a monitored subflow. This
| value reflects the activity over the latest collection interval for the monitoring
| agent. This value is taken from the averages determined within the CandleMonitor
| nodes that are placed in the message flows with a **type** attribute of input only.
| Note that the value includes any processing time taken by the MQInput node.

| **Current Interval** Length of time in seconds of the latest collection interval that the
| displayed statistics labeled Current reflect.

| **Current Msg Byte Input Rate** Current rate of message bytes input to monitored
| subflows. This value reflects the activity over the latest collection interval for the
| monitoring agent. Input message byte rates are determined by the CandleMonitor
| nodes that are placed in message flows with a **type** attribute of subFlowInput or
| input.

| **Current Msg Byte Output Rate** Current rate of message bytes output from
| monitored subflows. This value reflects the activity over the latest collection
| interval for the monitoring agent. Output message byte rates are determined by the
| CandleMonitor nodes that are placed in message flows with a **type** attribute of
| subFlowOutput or output.

| **Current Msg Input Count** Current number of messages input to monitored
| subflows. This value reflects the activity over the latest collection interval for the
| monitoring agent. Input messages are counted by CandleMonitor nodes that are
| placed in message flows with a **type** attribute of subFlowInput or input.

| **Current Msg Input Rate** Current rate of messages input to monitored subflows.
| This value reflects the activity over the latest collection interval for the monitoring
| agent. Input message rates are determined by CandleMonitor nodes that are placed
| in message flows with a **type** attribute of subFlowInput or input.

| **Current Msg Output Count** Current number of messages output from monitored
| subflows. This value reflects the activity over the latest collection interval for the
| monitoring agent. Output messages are counted by CandleMonitor nodes that are
| placed in message flows with a **type** attribute of subFlowOutput or output.

| **Current Msg Output Rate** Current rate of messages output from monitored
| subflows. This value reflects the activity over the latest collection interval for the
| monitoring agent. Output message rates are determined by CandleMonitor nodes
| that are placed in message flows with a **type** attribute of subFlowOutput or
| output.

| **Execution Group** Name of an execution group. An execution group is a process
| that runs in the broker and provides the execution environment in which message
| flows run. Users associate message flows with execution groups during message
| flow assignment in the Control Center or Message Brokers Toolkit. The execution
| group name is encoded in UTF8.

| **First Msg Date & Time** Date and time of the first message that goes through a
| monitored subflow after monitoring begins or statistics are reset.

Last Msg Date & Time Date and time of the last message that goes through a monitored subflow.

Message Flow Name of a message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Monitor Nodes Number of CandleMonitor nodes within a subflow.

Origin Node Name of the managed system.

Overall Avg Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Avg Msg Size Overall average size of messages (including headers) going through monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Avg Msg Time Sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average total time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Avg Queue Time Overall average number of seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Overall Max Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Max Msg Size Overall maximum size of messages (including headers) going through monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Max Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the

message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Overall Min Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Min Msg Size Overall minimum size of messages (including headers) going through monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Min Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Overall Msg Byte Input Rate Overall rate of message bytes input to monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Overall Msg Byte Output Rate Overall rate of message bytes output from monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Overall Msg Input Count Overall number of messages input to monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Overall Msg Input Rate Overall rate of messages input to monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Overall Msg Output Count Overall number of messages output from monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a type attribute of subFlowOutput or output.

Overall Msg Output Rate Overall rate of messages output from monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Sample Date & Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Sub-Flow Name Name given to the subflow by the message flow developer. The name is assigned by setting the subFlowName attribute for a CandleMonitor node when placing it into a message flow. This attribute is encoded in UTF8.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

ODBC Resource Statistics attributes

After resource statistics collection is started, use the ODBC Resource Statistics attributes to get the statistics information that is collected for ODBC DSN resources. Statistics are reported for each ODBC DSN that has been accessed since the execution group starts.

Active Connections The number of connections currently open to this DSN.

Aggregation Date & Time The aggregation date and time of the collection interval. The value is the same as End Date & Time.

Broker The name of the monitored broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Closed Connections The number of connections to this DSN that have ever been open, but are now closed. This number includes connections closed due to an error, forced closed by the DBMS or closed by broker because it was no longer required (for example, thread idle for 60 seconds).

Connection Errors The number of times that a connection to this DSN has been detected to have a connection error (which might have caused the error to be closed and therefore also contributed to closed connections).

End Date & Time The end date and time of the collection interval.

Execute Failures The total number of times that any statement has failed against this DSN.

Execute Successes The total number of times any statement has been executed successfully against this DSN.

Execution Group The name of an execution group. This attribute is encoded in UTF8.

Origin Node The managed system name of the monitored broker.

Resource Identifier The identifier of the ODBC DSN resource.

Resource Type The type of the resource. For ODBC DSN resource, this attribute is always ODBC.

Start Date & Time The start date and time of the sampling interval.

Parsers Resource Statistics attributes

All message flows in an execution group create parsers to parse and write input and output messages. After resource statistics collection is started, use the Parsers Resource Statistics attributes to see how much resource is being consumed by the message trees and bit streams that these parsers own. The statistics are summarized by message flow name and parser names used within the message flow.

Aggregation Date & Time The aggregation date and time of the collection interval. The value is the same as End Date & Time.

Approximate Memory KB The approximate amount of user data related memory used for the named message flow parser type.

Broker The name of the monitored broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

End Date & Time The end date and time of the collection interval.

Execution Group The name of an execution group. This attribute is encoded in UTF8.

Failed Reads The number of failed parses that have occurred for the named message flow parser type.

Failed Writes The number of failed writes that have occurred for the named message flow parser type.

Fields The number of message fields associated with the named message flow parser type. These fields are retained by the parser and are used for constructing the message trees.

Max KB Read The largest bit stream parsed by the parser type for the named message flow.

Max KB Written The largest bit stream written by the parser type for the named message flow.

Origin Node The managed system name of the monitored broker.

Reads The number of successful parses that have been performed by the named message flow parser type.

Resource Identifier The identifier of the message flow parser.

Resource Type The type of the resource. For Parsers resource, this attribute is always Parsers.

Start Date & Time The start date and time of the sampling interval.

Threads The number of message flow threads that have contributed to the statistics for a message flow's parser type accumulation.

Used Memory MB The amount of memory that is currently in use.

Writes The number of successful writes that have been performed by the named message flow parser type.

Processing Node Attributes

The Processing Node Attributes list all attribute values for a message processing node.

Application Name of the application that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Attribute Name Name of an attribute applicable to this message processing node. Attributes are defined for message processing nodes by the designer of the node to provide user customization of node activity. This attribute is encoded in UTF8.

Attribute Value Value of the named attribute applicable to this message processing node. The attribute value is defined within the Control Center or Message Brokers Toolkit when configuring a node within a message flow. It is encoded in UTF8.

Broker Identifier of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Library Name of the library that contains the message flow. The valid format is an alphanumeric string of up to 256 characters. This attribute is encoded in UTF8.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Message Flow UUID UUID string that is associated with the message flow to uniquely identify the message flow. This attribute is available for brokers of version 8 or later.

Node Label External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. The node label is encoded in UTF8.

Node Name Actual long name of the message processing node as it is defined within the Control Center or Message Brokers Toolkit. This name differs from the label name which the user provides for an instance of the node when including it within a message flow. This attribute is encoded in UTF8.

Origin Node The managed system name of the monitored broker.

Product Events attributes

The Product Events attributes display agent-level product events that affect the agent's ability to collect broker data. These events are gathered when they occur, with some events being retained for regular report queries, as set in the agent parameters file. Use the Product Events attributes to create situations for detecting concerned product events.

Remember: Do not use the situations based on Product Events attributes as a mechanism for detecting the status of a broker. Instead, the `Broker_Information.Broker_Status` attribute should be used for this purpose, as demonstrated in the product-provided situations.

Event The action that produced the product event. This attribute has the following possible values:

- `Collector_Initialization_Failed`
- `Collector_Cycle_Processing_Failed`
- `No_Brokers_Discovered_By_Agent`
- `Agent_Shutdown`
- `Queue_Manager_Connection_Failed`
- `Required_Queue_Unavailable`
- `Broker_Request_Message_Put_Failed`
- `No_Storage--Message_Not_Processed`
- `Broker_Product_Not_Installed`
- `No_Storage--Events_Not_Retained`
- `No_Storage--User_Data_Request_Affected`
- `Broker_Not_Started`
- `No_Storage--Collected_Data_Lost`
- `Agent_Broker_Data_Requests_Failed`
- `Broker_Not_Responding`
- `Broker_Not_Deployed--Data_Unavailable`
- `Broker_Events_Subscription_Failed`
- `No_Storage--Agent_Must_Shutdown`
- `Reply_Queue_Message_Get_Failed`
- `CandleMonitor_Data_Retrieval_Failed`
- `CandleMonitor_Node_Not_Deployed`
- `Agent_Environment_Data_Missing`
- `Agent_Userid_Group_Missing`
- `ConfigMgr_Qmgr_Name_Missing`
- `Table_Query_Missing_Input_Data`
- `UserId_Not_Authorized_For_Command`
- `No_Storage--Persistent_Data_Lost`
- `Message_Length_More_Than_Maximum_Value`
- `Java_Sub-process_Error`
- `Java_Sub-process_Timeout`
- `Java_Sub-process_Operation_Failed`
- `Java_Sub-process_Operation`

- Java_Sub-process-Manager_Operation_Failed
- Java_Sub-process-Manager_Operation

Event Date & Time Date and time when the current event occurs.

Event Qualifier 1 (Deprecated) The first qualifying information that pertains to the event. This information varies according to the event type.

Event Qualifier 1 The first qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Event Qualifier 2 (Deprecated) The second qualifying information that pertains to the event. This information varies according to the event type.

Event Qualifier 2 The second qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Reason Reason code associated with the problem that produced the product event. This attribute can take the following values:

- n/a
- Alias_Base_Q_Type_Error
- Backed_Out
- Buffer_Error
- Buffer_Length_Error
- Connection_Broken
- Data_Length_Error
- Dynamic_Q_Name_Error
- Expiry_Error
- Feedback_Error
- Get_Inhibited
- Handle_Not_Available
- Hconn_Error
- Hobj_Error
- Syncpoint_Limit_Reached
- Max_Conns_Limit_Reached
- MD_Error
- Missing_Reply_To_Q
- Msg_Type_Error
- Msg_Too_Big_For_Q
- No_Msg_Available
- No_Msg_Under_Cursor
- Not_Authorized
- Not_Open_For_Browse
- Not_Open_For_Input
- Not_Open_For_Output
- Object_Changed

- | • Object_In_Use
- | • Object_Type_Error
- | • OD_Error
- | • Option_Not_Valid_For_Type
- | • Options_Error
- | • Persistence_Error
- | • Persistence_Not_Allowed
- | • Priority_Exceeds_Maximum
- | • Priority_Error
- | • Put_Inhibited
- | • Q_Deleted
- | • Q_Full
- | • Q_Space_Not_Available
- | • Q_Type_Error
- | • Q_Mgr_Name_Error
- | • Q_Mgr_Not_Available
- | • Report_Options_Error
- | • Second_Mark_Not_Allowed
- | • Security_Error
- | • Signal_Outstanding
- | • Signal_Request_Accepted
- | • Storage_Not_Available
- | • Syncpoint_Not_Available
- | • Truncated_Msg_Accepted
- | • Truncated_Msg_Failed
- | • Unknown_Alias_Base_Q
- | • Unknown_Object_Name
- | • Unknown_Object_Q_Mgr
- | • Unknown_Remote_Q_Mgr
- | • Wait_Interval_Error
- | • Xmit_Q_Type_Error
- | • Xmit_Q_Usage_Error
- | • Not_Open_For_Pass_All
- | • Not_Open_For_Pass_Ident
- | • Not_Open_For_Set_All
- | • Not_Open_For_Set_Ident
- | • Context_Handle_Error
- | • Context_Not_Available
- | • Signal1_Error
- | • Object_Already_Exists
- | • Object_Damaged
- | • Resource_Problem
- | • Another_Q_Mgr_Connected
- | • Unknown_Report_Option
- | • Storage_Class_Error

- | • COD_Not_Valid_For_XCF_Q
- | • Suppressed_By_Exit
- | • Format_Error
- | • Source_CCSID_Error
- | • Source_Integer_Enc_Error
- | • Source_Decimal_Enc_Error
- | • Source_Float_Enc_Error
- | • Target_CCSID_Error
- | • Target_Integer_Enc_Error
- | • Target_Decimal_Enc_Error
- | • Target_Float_Enc_Error
- | • Not_Converted
- | • Converted_Msg_Too_Big
- | • Adapter_Storage_Shortage
- | • Adapter_Conn_Load_Error
- | • Adapter_Serv_Load_Error
- | • Adapter_Defs_Error
- | • Adapter_Defs_Load_Error
- | • Adapter_Conv_Load_Error
- | • DH_Error
- | • Multiple_Reasons
- | • Open_Failed
- | • CNO_Error
- | • CICS_Wait_Failed
- | • Object_Name_Error
- | • Object_Q_Mgr_Name_Error
- | • Recs_Present_Error
- | • Object_Records_Error
- | • Response_Records_Error
- | • ASID_Mismatch
- | • PMO_Record_Flags_Error
- | • Put_Msg_Records_Error
- | • Conn_Id_In_User
- | • Q_Mgr_Quiescing
- | • Q_Mgr_Stopping
- | • Duplicate_Recov_Coord
- | • PMO_Error
- | • API_Exit_Load_Error
- | • Remote_Q_Name_Error
- | • Inconsistent_Persistence
- | • GMO_Error
- | • Pageset_Full
- | • Pageset_Error
- | • Name_Not_Valid_For_Type
- | • Unexpected_Error

- | • Unknown_Xmit_Q
- | • Unknown_Def_Xmit_Q
- | • Def_Xmit_Q_Type_Error
- | • Def_Xmit_Q_Usage_Error
- | • Name_In_Use
- | • Connection_Quiescing
- | • Connection_Stopping
- | • Adapter_Not_Available
- | • No_Msg_Locked
- | • Connection_Not_Authorized
- | • Call_In_Progress
- | • Incomplete_Group
- | • Incomplete_Msg
- | • Inconsistent_CCSIDs
- | • Inconsistent_Encodings
- | • Inconsistent_UOW
- | • Invalid_Msg_Under_Cursor
- | • Match_Options_Error
- | • MDE_Error
- | • Msg_Flags_Error
- | • Msg_Seq_Number_Error
- | • Offset_Error
- | • Original_Length_Error
- | • Segment_Length_Zero
- | • UOW_Not_Available
- | • Wrong_GMO_Version
- | • Wrong_MD_Version
- | • Group_Id_Error
- | • Inconsistent_Browse
- | • RFH_Command_Error
- | • RFH_Parm_Error
- | • RFH_Duplicate_Parm
- | • RFH_Parm_Missing
- | • Topic_Error
- | • Not_Registered
- | • Q_Mgr_Name_Error
- | • Q_Name_Error
- | • Incorrect_Q
- | • Correl_ID_Error
- | • Not_Authorized
- | • Reg_Options_Error
- | • Filter_Error
- | • Wrong_User
- | • Take_Action_Auth_Users_Parameters
- | • Object_Unusable

- | • Thread_Create_Error
- | • Associated_Object_Error
- | • Unknown_Failure
- | • Storage_Depleted
- | • XML_Parse_Init_Failure
- | • XML_Parse_Error
- | • XML_Message_Type_Error
- | • Object_Construction_Error
- | • QMgr_Connection_Thread_Unavailable
- | • QMgr_Connection_Object_Missing
- | • QMgr_Pipe_Communication_Error
- | • Task_Ending_Cannot_Process
- | • Task_Request_Bad
- | • Too_Many_Get_Signal_Requests
- | • Task_Attach_Failed
- | • Message_Flow_Definitions
- | • Dynamic_Subscription_Engine_Data
- | • Message_Flow_Statistics
- | • Broker_Events
- | • Message_Flow_Events
- | • Product_Events
- | • Execution_Group_Information
- | • Message_Flow_Information
- | • Message_Processing_Node_Information
- | • Neighbors
- | • Subscriptions
- | • Retained_Publications
- | • ACL_Entries
- | • Execution_Group_Statistics
- | • Message_Flow_Statistics
- | • CandleMonitor_Node_Statistics
- | • Collected_String_Lost
- | • CandleMonitor_Data_Lost
- | • User_Command
- | • Message_Processing_Node_Attributes
- | • Sub-Flow_Statistics
- | • User_Statistics
- | • Message_Flow_Accounting
- | • PubSub_Statistics
- | • Message_Flow_CMP
- | • ATTRIBUTE_PARTIAL
- | • FAILED_FOUND_CHILD
- | • FAILED_CREATE_CHILD
- | • FAILED_GET_CHILD
- | • FAILED_BROKER_CREATE

- FAILED_BROKER_GET_INSTANCE

SOAP Input Resource Statistics attributes

The SOAPInput and SOAPReply nodes send and receive SOAP messages. After resource statistics collection is started, use the SOAP Input Resource Statistics attributes to review how many inbound messages the SOAPInput node receives, how many replies the SOAPReply node sends, and how many of those calls are successful or result in SOAP Faults, on a per-operation basis. Statistics for the SOAP nodes are collected with both HTTP and JMS transport. You can review the name of the applied policy set if one is defined.

Aggregation Date & Time The aggregation date and time of the collection interval. The value is the same as End Date & Time.

Broker The name of the monitored broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

End Date & Time The end date and time of the collection interval.

Execution Group The name of an execution group. This attribute is encoded in UTF8.

Fault Replies Sent The number of SOAP Fault replies sent to the client. These can be user-defined faults or broker exceptions.

Inbound Msgs Faulted before Flow The number of messages that faulted before reaching the flow. This includes input messages that are sent down the Failure terminal.

Inbound Msgs Made Flow The number of messages that made the flow without faulting.

Name Indicates the names of the message flow, node, and operation. The value is either "Summary", Flow.Node.Operation, or [Undeployed].Flow.Node.Operation, where Flow is the name of your message flow, Node is the node name and Operation is the name of the operation.

Origin Node The managed system name of the monitored broker.

Policy Set Applied The name of the policy set if one is defined.

Resource Identifier The identifier of the SOAP node resource.

Resource Type The type of the resource. For SOAP nodes, this value is either SOAPInput or SOAPReply.

Start Date & Time The start date and time of the sampling interval.

Successful Replies Sent The number of successful replies, without SOAP Fault, sent to the client.

Total Inbound Msgs The total number of SOAP messages that are received from the client. This value is equal to the sum of **Inbound Msgs Made Flow** and **Inbound Msgs Faulted before Flow** values.

| **Total Replies Sent** The total number of SOAP replies that are sent back to the
| client. This value is equal to the sum of **Successful Replies Sent** and **Fault**
| **Replies Sent** values.

For WebSphere Message Broker V6 (all releases)

If the broker to be monitored is WebSphere Message Broker V6 (all releases), the monitoring agent can monitor the attribute groups that are listed in this section.

ACL Entries attributes

The ACL Entries attributes provide Access Control List entries that define access to topics for publishing and subscription. Use the ACL Entries attributes to create situations for detecting the change of certain user access to certain topics.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Persistent Whether the principal is allowed to set message persistence. This attribute has the following possible values:

- Yes
- No
- Inherit (inherit from the nearest ancestor in the topic tree)

Principal (Deprecated) Name of a user or group taken from the User Name Server that is used to control access to topics.

Principal Name of a user or group taken from the User Name Server that is used to control access to topics. This attribute is encoded in UTF8.

Publish Whether the principal is allowed to publish messages for the given topic. This attribute has the following possible values:

- Allow
- Deny
- Inherit (inherit from the nearest ancestor in the topic tree)

Subscribe Whether the principal is allowed to subscribe to the given topic. This attribute has the following possible values:

- Allow
- Deny
- Inherit (inherit from the nearest ancestor in the topic tree).

Topic (Deprecated) A descriptive name of published data to which one can subscribe. Topic names for subscriptions and ACL entries can contain wildcard characters (as defined by the topic name rules of the broker product), so that more than one topic can be identified.

Topic A descriptive name of published data to which one can subscribe. Topic names for subscriptions and ACL entries can contain wildcard characters (as defined by the topic name rules of the broker product), so that more than one topic can be identified. This attribute is encoded in UTF8.

Broker Events attributes

The Broker Events attributes gather broker events as the events occur for situation use, with some events being retained for regular report queries as set in the agent parameter file. Use the Broker Events attributes to create situations for detecting concerned broker events.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing within WebSphere Business Integration brokers is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing within WebSphere Business Integration brokers is performed at the broker level. This attribute is encoded in UTF8.

Event The action that triggers the event. This attribute has the following possible values:

- Unknown
- Create
- Delete
- Change
- MultipleUpdates
- Start
- Stop
- ACL-Unknown
- ACL-Create
- ACL-Delete
- ACL-Change
- ACL-Updates
- InsideFlow-Unknown
- InsideFlow-Create
- InsideFlow-Delete
- InsideFlow-Change
- InsideFlow-Updates

Event Date and Time The put date and time for the publication of the current event, which does not necessarily reflect the event time for a retained publication, such as a Status Start event for the broker.

Event Qualifier 1 (Deprecated) The first qualifying information that pertains to the event. This information varies according to the event type.

Event Qualifier 1 The first qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Event Qualifier 2 (Deprecated) The second qualifying information that pertains to the event. This information varies according to the event type.

Event Qualifier 2 The second qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Event Type Type of event. This attribute can take the following values:

- Configuration
- Expiry/Publication
- Expiry/Subscription
- Neighbor
- Security
- Status
- Subscription
- Topic
- ClonedSubscriptionState
- PubSubConfig

Origin Node The name of the managed system that is the data source for the current row.

Qualifier 1 Description A short description of the item that Qualifier 1 refers to. This attribute can take the following values:

- n/a
- ExecutionGroup
- Broker
- Neighbor
- AllNeighbors
- Topic
- AllTopics
- MultipleTopics

Qualifier 2 Description A short description of the item that Qualifier 2 refers to. This attribute can take the following values:

- n/a
- MessageFlow
- MultipleMessageFlows
- AllMessageFlows
- ExpiryTimeStamp
- StateCloned

Broker Information attributes

The Broker Information attributes provide the activity status of the broker, name of the associated queue manager of the broker, and indicate whether the agent can connect to the queue manager of the broker. Use the Broker Information attributes to create situations for detecting changes in broker status.

ASID On z/OS systems, this attribute indicates the address space identifier (ASID) of the broker job in hexadecimal notation. This attribute is not applicable on distributed systems.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Broker Status State of the broker. This attribute can take the following values:

- n/a
- Started
- Stopped

Component Type Type of the component. This attribute can take the following values:

- n/a
- MessageBroker
- EventBroker
- ConfigurationManager
- UserNameServer
- ToolkitStudio

Configuration Delay Timeout Maximum time in seconds to allow a broker to process a deployed message.

Configuration Manager QMgr Name of the queue manager for the configuration manager of the broker. It is shown as blank if the Configuration Manager queue manager name that is supposed to be in broker information cannot be determined. This name is determined by analyzing broker subscription data. If that analysis fails to provide the name, the Configuration Manager QMgr value is shown as blank.

Configuration Timeout Maximum time in seconds to allow a broker to process a deployed message of minimum size.

Converters Path to search for International Components for Unicode (ICU) converters or converter packages.

Data Source Name (Deprecated) Data source name of the database in which broker tables are created, as specified during broker creation; on z/OS systems, this name is specified as the DB2 location name during broker customization.

Data Source Name Data source name of the database in which broker tables are created, as specified during broker creation; on z/OS systems, this name is specified as the DB2 location name during broker customization. This attribute is encoded in UTF8.

Execution Groups Number of execution groups that are running on the broker.

FAD Level The level of format and data of current component. If two versions of the component have the same FAD level, all the database tables, command-line arguments, and some other aspects must be compatible.

Http Listener Port Port that the broker HTTP listener uses.

Http Listener Trace Level Trace level of broker HTTP listener. This attribute can take the following values:

- none - no trace
- normal - normal trace
- debug - detailed debug trace

Http Listener Trace Size Trace size of the broker HTTP listener.

Jobname Taskname On z/OS systems, this attribute indicates the name of the job; on distributed systems, it indicates the name of the task.

Lil Path Where LIL (loadable implementation libraries) files are loaded from. This attribute is encoded in UTF8. It is shown as blank if there is no value on the system for a given broker registry string attribute.

Migration Needed Whether migration is requested from an existing Publish/Subscribe broker when the current broker is being created. This attribute can take the following values:

- n/a
- Yes (indicates that migration has been requested)
- No (indicates that migration has not been requested)

Origin Node Name of the managed system.

Platform Name of the host platform on which the broker is running.

Platform Version Version of the host platform on which the broker is running.

Process ID Process identifier for the administrative agent within the broker.

Pub Sub Access Control Whether publish/subscribe access control is to be enabled for this broker. This attribute can take the following values:

- Enabled - Publish/subscribe access control is on
- Disabled - Publish/subscribe access control is off
- n/a

Queue Manager Name of the queue manager associated with the broker. Several WebSphere Business Integration components require a queue manager to be associated with them as all communication is accomplished using WebSphere MQ.

QMgr Connect Status State of the connection between the monitoring agent and the queue manager. This attribute has the following possible values:

- n/a
- Connected
- NotConnected

Release Version and release of the broker.

Start Date & Time On z/OS and UNIX systems, this attribute shows the job start date and time for the broker; this attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Started Msg Flows Number of message flows within a broker that have a status of started.

Statistics Archive Interval Indicates length of the interval (in minutes) for which archive statistics are produced if turned on for the broker.

Stopped Msg Flows Number of message flows within a broker that have a status of stopped.

System Name Name of the host system on which the broker is running.

Total Msg Flows Number of message flows in a broker.

Trace Level Level of trace currently active. This attribute has the following possible values:

- None
- Normal
- Debug
- n/a

Trusted QMgr Whether the association between this broker and its queue manager is set as a trusted relationship. Trusted status causes the broker to connect to its queue manager using fast-path binding. Otherwise, typical connections are used. This attribute has the following possible values:

- n/a
- Yes
- No

User Name Server QMgr Name of the queue manager associated with the User Name Server that has been defined to serve access control requests for the message broker. Note that if access control is not used, this field might be blank, because no User Name Server is serving the broker.

UUID The UUID string associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Broker Statistics attributes

The Broker Statistics attributes provide message flow statistics summarized at the broker level. Use the Broker Statistics attributes to create situations for detecting the long message processing time in message flows at the broker level.

ASID On z/OS systems, this attribute indicates the address space identifier (ASID) of the broker job in hexadecimal notation. This attribute is not applicable on distributed platforms.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Avg Flow Time Current average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Current Avg Msg Size Current average size of messages (including headers) going through monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Current Avg Msg Time Sum of Current Average Queue Time and Current Average Flow Time to give the average time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Avg Queue Time Current average time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Current Interval The length of the sample interval (length of time between samples) in seconds. This attribute is the interval over which the delta calculations for other attributes that begin with "Current" are made - that is, the interval that they apply to.

Current Msg Byte Input Rate Current rate of message bytes that are entered into monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Msg Byte Output Rate Current rate of message bytes that are produced from monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Current Msg Input Count Current number of messages that are entered into monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Msg Input Rate Current rate of messages that are entered into monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Msg Output Count Current number of messages that are produced from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Current Msg Output Rate Current rate of messages that are produced from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

First Msg Date and Time Date and time of the first message that goes through a monitored message flow after monitoring begins or statistics are reset.

Jobname Taskname On z/OS systems, this attribute indicates the name of the job; on distributed platforms, this attribute indicates the name of the task.

Last Msg Date and Time Date and time of the last message that goes through a monitored message flow.

Monitored Execution Groups Number of execution groups that have a CandleMonitor node included in one or more of their message flows.

Monitored Msg Flows Number of message flows in the broker that include a CandleMonitor node.

Origin Node Name of the managed system that is the source of the data for the current row.

Overall Avg Flow Time The overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Size The overall average size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Avg Msg Time The sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Avg Queue Time The overall average number of seconds (up to millisecond accuracy) that a message remained in a monitored message flow input queue since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Max Flow Time The overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Msg Size The overall maximum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Max Queue Time The overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Min Flow Time The overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Msg Size The overall minimum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Min Queue Time The overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Msg Byte Input Rate The overall rate of message bytes input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Byte Output Rate The overall rate of message bytes output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Input Count The overall number of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Input Rate The overall rate of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Msg Output Count The overall number of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Msg Output Rate The overall rate of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Platform Name of the host platform on which the component is running.

Platform Version Version of the host platform on which the component is running.

Process ID Process identifier for the administrative agent within the broker.

Sample Date & Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Start Date & Time On z/OS and UNIX systems, it shows the job start date and time for the component; this attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Started Msg Flows Number of message flows within the broker that have a status of started.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not for display.

Status State of the broker. This attribute has the following possible values:

- n/a
- Started
- Stopped

System Name Name of the host system on which the component is running.

Broker Topology attributes

The Broker Topology attribute group contains the attributes that are used in drawing high-level objects in the broker topology view. The broker topology attributes are related to brokers, execution groups, and message flows. The

information in this table can be used only in workspace queries, and cannot be used as conditions in situations. The data in this attribute group is intended only for drawing the topology view in the Tivoli Enterprise Portal.

From Object Name The name of the parent object in the object hierarchy. This value is a UTF-8 string of up to 256 alphanumeric characters. Possible values are listed in Table 14 on page 148.

Table 16. From Object Name values

Type of Object	From Object Name value
Broker	The name of the host on which the broker runs or no value
Execution group	Name of the broker to which the execution group belongs
Message flow	Name of the execution group to which the message flow belongs

Object UUID The UUID of the object. This value is a UTF-8 string of up to 36 alphanumeric characters.

Object Name The name of the object. This value is a UTF-8 string of up to 256 alphanumeric characters, and is the name of a broker, execution group or message flow.

Object Type The type of the object. This attribute can take the following values:

- kqi.BrokerNode
- kqi.ExecutionGroupNode
- kqi.MessageFlowNode.

Object Status The status of the object. Possible values are listed in Table 15 on page 149.

Table 17. Object Status values

Value	Meaning	Notes
0	Stopped	The status of an execution group with no user-defined message flows is Stopped. The status of ConfigurationMessageFlow and PubSubControlMsgFlow system message flows is not considered when determining the status of an execution group.
1	Started	The status of ConfigurationMessageFlow and PubSubControlMsgFlow message flows is always started and is not used when determining this value.
2	Partially started	This value applies only to execution groups and indicates that only a part of the execution group's message flows are started. The status of ConfigurationMessageFlow and PubSubControlMsgFlow system message flows is not used when determining the status of an execution group.

Origin Node The name of the managed system.

To Object UUID The UUID of the next object in the object hierarchy. This value is a UTF-8 string of up to 36 alphanumeric characters.

CandleMonitor Node Statistics attributes

The CandleMonitor Node Statistics attributes display the CandleMonitor nodes by execution group and message flow, and show statistics at the lowest collected level.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Average Flow Time The current average time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor node regardless of the node type.

Current Average Message Size The current average size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor node regardless of the node type.

Current Average Message Time The sum of Current Average Queue Time and Current Average Flow Time to give the average time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Average Queue Time The current average time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is non-zero only for the nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Current Message Byte Rate The current rate of message bytes going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is determined for every CandleMonitor node regardless of the node type.

Current Message Count The current number of messages going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is determined for every CandleMonitor node regardless of the node type.

Current Message Rate The current rate of messages going through this CandleMonitor node. This value reflects the activity over the latest collection interval for the monitoring agent. This value is determined for every CandleMonitor node regardless of the node type.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in

which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

First Message Date and Time Date and time of the first message that goes through this CandleMonitor node after monitoring begins or statistics are reset.

Last Message Date and Time Date and time of the last message that goes through this CandleMonitor node.

Message Flow (Deprecated) Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Monitor Type The type attribute associated with this CandleMonitor node. This attribute has the following possible values:

- n/a
- Input
- Output
- subFlowInput
- subFlowOutput
- Other

Origin Node Name of the node that identifies the data source for the current row. The displayed name indicates either an agent node or broker subnode.

Overall Average Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor node regardless of the node type.

Overall Average Message Size Overall average size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the average determined within the CandleMonitor node regardless of the node type.

Overall Average Message Time The sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Average Queue Time Overall average time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This

value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is non-zero only for the nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Overall Maximum Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor node regardless of the node type.

Overall Maximum Message Size Overall maximum size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximum determined within the CandleMonitor node regardless of the node type.

Overall Maximum Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is non-zero only for nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Overall Message Byte Rate Overall rate of message bytes going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is determined by every CandleMonitor node regardless of the node type.

Overall Message Count Overall number of messages going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is determined by every CandleMonitor node regardless of the node type.

Overall Message Rate Overall rate of messages going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is determined by every CandleMonitor node regardless of the node type.

Overall Minimum Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message from the location of this node to the end of the message flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor node regardless of the node type.

Overall Minimum Message Size Overall minimum size of messages (including headers) going through this CandleMonitor node. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor node regardless of the node type.

Overall Minimum Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for this message

flow. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is non-zero only for the nodes of input type, and is taken from the averages determined within the CandleMonitor node. Note that the value includes any processing time taken by the MQInput node.

Processing Node Label (Deprecated) The external name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement.

Processing Node Label The external name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. This attribute is encoded in UTF8.

Sample Date and Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

Sub-Flow Name (Deprecated) Name given to the subflow by the message flow developer using the Control Center or Message Brokers Toolkit. The name is assigned by setting the subFlowName attribute for a CandleMonitor node when placing it into a message flow.

Sub-Flow Name Name given to the subflow by the message flow developer using the Control Center or Message Brokers Toolkit. The name is assigned by setting the subFlowName attribute for a CandleMonitor node when placing it into a message flow. This attribute is encoded in UTF8.

Components attributes

The Components attributes provide a list of broker product components and their state at agent level. Use the Components attributes to create situations for detecting the state change of components.

ASID On z/OS systems, it indicates the address space identifier (ASID) of the component job in hexadecimal notation; this attribute is not applicable on distributed platforms.

Component (Deprecated) Name of the broker product component. This name is either one of the following constant names, or the name given to a broker component created on the monitored host system:

- ConfigMgr
- UserNameServer
- ControlCenter

Component Name of the broker product component. This attribute is encoded in UTF8. This name is either one of the following constant names, or the name given to a broker component created on the monitored host system:

- ConfigMgr
- UserNameServer

- ControlCenter

Component Type Type of the component. This attribute has the following possible values:

- N/A
- MessageBroker
- EventBroker
- ConfigurationManager
- UserNameServer
- ToolkitStudio

Remember: ToolkitStudio indicates either Message Brokers Toolkit V5 or Control Center V2.1.

Jobname Taskname On z/OS systems, it indicates the name of the job; on distributed systems, it indicates the name of the task.

Origin Node Name of the managed system.

Platform Name of the host platform on which the component is running.

Platform Version Version of the host platform on which the component is running.

Process ID Process identifier for the administrative agent within the broker or other component.

Queue Manager Name of the queue manager associated with the broker or other component. Several components require a queue manager to be associated with them because all communication is accomplished using WebSphere MQ.

Release Version and release of the broker product components.

Start Date & Time On z/OS and UNIX systems, it shows the job start date and time for the component; this attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Status State of the component. This attribute has the following possible values:

- n/a
- Started
- Stopped
- Installed

System Name Name of the host system on which the component is running.

User Name Server Queue QMgr Name of the queue manager associated with the User Name Server that has been defined to serve access control requests for the message broker. Note that if access control is not used, this field might be blank because no User Name Server is serving the broker.

Execution Group Information attributes

The Execution Group Information Attribute group shows execution groups that are defined for the broker, and the definition information for each execution group.

Broker (Deprecated) Identifies the broker that is the data source for the current row. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Identifies the broker that is the data source for the current row. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Event Log File Mode Indicates how often log entries are saved to the event log. This attribute has the following possible values:

- Safe (for frequent writing)
- Fast (for infrequent writing)

Event Log File Size Maximum size of the event log file in bytes. For Windows NT systems, this value is 0, because the Windows NT Event Log is used instead. Event records within the file display significant broker operations such as start or stop.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Started Message Flows Number of message flows within an execution group that have a status of started.

Stopped Message Flows Number of message flows within an execution group that have a status of stopped.

Total Message Flows Number of message flows in an execution group.

Trace Filter Name of the filter that is used to determine which log entries are written to the trace log.

Trace Level The level of trace that is currently active. This attribute has the following possible values:

- None
- Normal
- Debug

Trace Log File Mode Indicates how often log entries are saved to the trace log. This attribute has the following possible values:

- Safe (for frequent writing)
- Fast (for infrequent writing)

Trace Log File Size Maximum size of the trace log file in bytes. Trace records display data that is used by IBM technical-support representatives to identify problems with the product.

User Trace Filter Name of the filter used to determine which log entries are written to the user trace log.

User Trace Level Level of user trace that is currently active. This attribute has the following possible values:

- None
- Normal
- Debug

User Trace Log File Mode Indicates how often log entries are saved to the user trace log. This attribute has the following possible values:

- Safe (for frequent writing)
- Fast (for infrequent writing)

User Trace Log File Size Maximum size of the user trace log file in bytes. User trace records display the data that covers the overall execution of a message flow for such items as nodes executed.

UUID The UUID string associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Execution Group Statistics attributes

The Execution Group Statistics attributes provide message flow statistics summarized at the execution group level. Use the Execution Group Statistics attributes to create situations for detecting long message processing time in message flows at execution group level.

Address Space Identifier On z/OS systems, it indicates the address space identifier (ASID) of the execution group job in hexadecimal notation. This attribute is not applicable on distributed systems.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Average Flow Time Current average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with the **type** attribute of input.

Current Average Message Size Current average size of messages (including headers) going through monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with the **type** attribute of input.

Current Average Message Time Sum of Current Average Queue Time and Current Average Flow Time to give the average time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Average Queue Time Current average time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with the **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Current Message Byte Input Rate Current rate of message bytes input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a type attribute of input.

Current Message Byte Output Rate Current rate of message bytes output from monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a type attribute of output.

Current Message Input Count Current number of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input messages are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Message Input Rate Current rate of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a type attribute of input.

Current Message Output Count Current number of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output messages are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Current Message Output Rate Current rate of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output message rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

First Message Date and Time Date and time of the first message that goes through a monitored message flow after monitoring begins or statistics are reset.

Jobname Taskname On z/OS systems, it indicates the name of the job; on distributed systems, it indicates the name of the task.

Last Message Date and Time Date and time of the last message that goes through a monitored message flow.

Monitored Message Flows Number of message flows in an execution group that include a CandleMonitor node.

Origin Node Name of the managed system.

Overall Average Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input.

Overall Average Message Size Overall average size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Average Message Time Sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average total time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Average Queue Time Overall average time in seconds (up to millisecond accuracy) that a message remained in a monitored message flow input queue since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Maximum Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input.

Overall Maximum Message Size Overall maximum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Maximum Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored

message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Message Byte Input Rate Overall rate of message bytes input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Message Byte Output Rate Overall rate of message bytes output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Message Input Count Overall number of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Message Input Rate Overall rate of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Message Output Count Overall number of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Message Output Rate Overall rate of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Minimum Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input.

Overall Minimum Message Size Overall minimum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Minimum Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within CandleMonitor nodes that are placed in the message flows with a type attribute of input. Note that the value includes any processing time taken by the MQInput node.

Platform Name of the host platform on which the execution group is running.

Platform Version Version of the host platform on which the execution group is running.

Process Identifier Process identifier for the execution group process.

Sample Date and Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Start Date and Time On z/OS and UNIX systems, it shows the job start date and time for the execution group. This attribute is not applicable on Windows systems. On UNIX systems, if the process is more than one day old, the time portion is displayed as 00:00:00.

Started Message Flows Number of message flows within an execution group that have a status of started.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

Status State of the execution group. This attribute has the following possible values:

- n/a
- Started
- Stopped

System Name Name of the host system on which the execution group is running.

Message Flow Accounting attributes

The Message Flow Accounting attributes show the message flow accounting and statistical data gathered by the brokers. The data might be gathered as snapshots, collected over short intervals, or as archival data, collected over longer intervals. Use the Message Flow Accounting attributes to create situations for detecting abnormal message flow accounting data, such as high Average CPU time and high Average Input Message Size.

Remember:

- Special care must be exercised in creating situations using time attributes that are measured in microseconds. For situations, the WebSphere Message Broker Monitoring agent supports only 32-bit signed integers; values exceeding 2,147,483,647 are returned as -1. Therefore, if you want to test whether maximum elapsed time is greater than 35 minutes (2,100,000,000 microseconds), create a situation as follows:

if Maximum Elapsed Microseconds > 2,100,000,000 OR Maximum Elapsed Microseconds < 0

Because values less than 0 also indicate a maximum elapsed time greater than 35 minutes. When this type of situation trips, the attribute giving the equivalent alphanumeric form of the time value can be displayed to determine the exact microsecond value encountered (Maximum Elapsed Time in this example).

- In workspaces, the largest number that can be displayed is 4,294,967,295. If this exact number is displayed, it means that the actual value is greater than or equal to this value.

Accounting Origin User-assigned identifier associated with a set of message flows which share a common origin.

Avg CPU Microseconds Average CPU time in microseconds spent processing input messages. Valid format is an integer. This number is negative if overflow occurs.

Avg CPU Time Average CPU time in microseconds spent processing input messages. The valid format is a string of form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Avg Elapsed Microseconds Average elapsed time in microseconds spent processing input messages. Valid format is an integer. This number is negative if overflow occurs.

Avg Elapsed Time Average elapsed time in microseconds spent processing input messages. The valid format is a string of form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Avg Input Msg Size Average size of input messages in bytes. Valid format is an integer. This number is negative if overflow occurs.

Avg Input Msg Size String Average size of input messages in bytes in string format.

Broker (Deprecated) Name of the broker in alphanumeric format.

Broker Name of the broker in alphanumeric format. This attribute is encoded in UTF8.

Broker UUID String associated with a broker that uniquely identifies it in alphanumeric format. This attribute is provided to assist in problem determination when reviewing trace and event log data. Valid format is an alphanumeric string.

End Date End date of the collection interval, in the format of YYYY-MM-DD. By default, this attribute is not displayed.

End Date & Time End date and time of the collection interval.

This attribute is designed for logging and reporting data collection times rather than for creating situations. To specify a time of day for monitoring, use attributes from the Universal Time or Local Time groups.

End Time End time of the collection interval, in the format HH:MM:SS:NNNNNN. By default, this attribute is not displayed. Valid format is an alphanumeric string.

Execution Group (Deprecated) Name of the execution group which the message flow belongs to. Valid format is an alphanumeric string.

Execution Group Name of the execution group which the message flow belongs to. Valid format is an alphanumeric string. This attribute is encoded in UTF8.

Execution Group UUID String that uniquely identifies the execution group. This identifier is provided to assist in problem determination when reviewing trace and event log data. Valid format is an alphanumeric string.

Input Byte Rate Rate in bytes per second, to two decimal places, at which the message flow processed input message bytes during the sampling period.

Input Msg Rate Rate in messages per second, to two decimal places, at which the message flow processed input messages during the sampling period.

Input Msg Wait CPU Microseconds Total CPU time in microseconds spent waiting for input messages. Valid format is an integer.

Input Msg Wait CPU Time Total CPU time in microseconds spent waiting for input messages. Valid format is ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Input Msg Wait Elapsed Microseconds Total elapsed time in microseconds spent waiting for input messages. Valid format is an integer.

Input Msg Wait Elapsed Time Total elapsed time in microseconds spent waiting for input messages, in alphanumeric format. The valid format is an alphanumeric string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Max CPU Microseconds Maximum CPU time in microseconds spent processing an input message, as an integer value.

Max CPU Time Maximum CPU time in microseconds spent processing an input message. The valid format is an alphanumeric string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Max Elapsed Microseconds Maximum elapsed time in microseconds spent processing an input message, as an integer value.

Max Elapsed Time Maximum elapsed time in microseconds spent processing an input message in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Max Input Msg Size Maximum size of input messages in bytes, as an integer value.

Max Input Msg Size String Maximum size of input messages in bytes in alphanumeric format.

Message Flow (Deprecated) Name of the message flow.

Message Flow Name of the message flow. This attribute is encoded in UTF8.

Min CPU Microseconds Minimum CPU time in microseconds spent processing an input message as an integer value.

Min CPU Time Minimum CPU time in microseconds spent processing an input message in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Min Elapsed Microseconds Minimum elapsed time in microseconds spent processing an input message as an integer value.

Min Elapsed Time Minimum elapsed time in microseconds spent processing an input message in alphanumeric format. The valid format is a string of form ssss:mmm:nnn , where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Min Input Msg Size Minimum message size in bytes as an integer value.

Min Input Msg Size String Minimum message size in bytes in alphanumeric format.

Node Statistics Count Number of node statistics rows with this message flow sample.

Number of Threads in Pool Number of threads in the pool.

Origin Node Name of the managed system that is the source of the data for the current row, in this case, it is the broker subnode name.

Query Type Type of query used in queries to get the data required. It is not displayed by default. This attribute has the following possible values:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy
- StatsSettingsModified
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts (in different tables) as part of the same message flow accounting statistic sample.

Start Date Start date of the sample interval in the format (YYYY-MM-DD). It is not displayed by default.

Start Date & Time Start date and time of the sample interval.

Start Time Start time of the sample interval in the format (HH:MM:SS:NNNNNN). It is not displayed by default.

Thread Statistics Number of thread statistics rows associated with this message flow sample.

Times Max Threads Reached Number of times the maximum number of threads is reached.

Total Aggregate Wait Timeouts Number of transaction timeouts that occur when processing a message (for AggregateReply nodes only).

Total Backouts Number of transaction backouts that occur when processing a message.

Total Commits Number of transaction commits that occur when processing a message.

Total CPU Microseconds Total CPU time in microseconds spent processing input messages, as an integer value.

Total CPU Time Total CPU time in microseconds spent processing input messages, in alphanumeric format. Valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Total Elapsed Microseconds Total elapsed time in microseconds spent processing input messages, as an integer value.

Total Elapsed Time Total elapsed time in microseconds spent processing input messages, in alphanumeric format. Valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds

- mmm = milliseconds
- nnn = microseconds

Total Input Msg Size Total size of input messages in bytes, as an integer value.

Total Input Msg Size String Total size of input messages in bytes in alphanumeric format.

Total Input Msgs Total number of messages that are processed.

Total MQ Errors Number of MQGET errors for MQInput nodes or Web Services errors for HTTPInput nodes.

Total Msg Errors Number of messages that contain errors.

Total Processing Errors Number of errors that occur when processing a message.

Message Flow Events attributes

The Message Flow Events attributes gather message flow events based on parameters set by the user of the CandleMonitor node. These events are gathered as they occur, with some events being retained for regular report queries as set within the agent parameters file. Use the Message Flow Events attributes to create situations for detecting concerned message flow events.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Correlation Identifier Correlation identifier for the message being processed when the message flow event occurs. The correlation identifier is obtained from the WebSphere MQ MQMD.

Event Date and Time Time when the current event occurs.

Event Message (Deprecated) Message that describes the event.

Event Message Message that describes the event. This attribute is encoded in UTF8.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Message Flow (Deprecated) Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Message Identifier Message identifier that uniquely identifies which message is being processed when the message flow event occurs. The message identifier is obtained from the WebSphere MQ MQMD.

Monitor Type Type attribute associated with the CandleMonitor node. This attribute has the following possible values:

- n/a
- Input
- Output
- subFlowInput
- subFlowOutput
- Other

Origin Node Name of the managed system.

Processing Node Label (Deprecated) External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node.

Processing Node Label External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This attribute is encoded in UTF8.

Put Application Name Name of the put application from the WebSphere MQ MQMD for the message being processed. This is the name of the application that originally put the message as determined by WebSphere MQ.

Put Application Type Type of the put application from the WebSphere MQ MQMD for the message being processed. This is the type of the application that originally put the message as determined by WebSphere MQ. This attribute has the following possible values:

- Unknown
- NoContext
- CICS
- OS390
- IMS
- OS2
- DOS
- UNIX
- QMGR
- OS400
- WINDOWS
- CICS_VSE

- WINDOWS_NT
- VMS
- NSK
- VOS
- OPEN_TP1
- VM
- IMS_BRIDGE
- XCF
- CICS_BRIDGE
- NOTES_AGENT
- TPF
- USER
- BROKER
- JAVA
- DQM
- CHANNEL_INITIATOR
- WLM
- BATCH
- RRS_BATCH
- SIB
- SYSTEM
- USER_FIRST
- USER_LAST

Put Date and Time Put date and time from the WebSphere MQ MQMD for the message being processed. This is the date and time that the message is originally put as determined by WebSphere MQ.

Thread Identifier Thread identifier that uniquely identifies which thread processor is active when the message flow event occurs.

Message Flow Information attributes

The Message Flow Information attributes display defined message flows by execution group and definition information for each message flow. Use the Message Flow Information attributes to create situations for detecting concerned status of message flow.

Additional Instances Number of additional instances of the message flow that the execution group should run to allow for more multi-processing of messages.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Commit Count Number of messages that must be processed for commit processing to occur.

Commit Interval Time interval at which commit processing should occur.

Coordinated Transaction Indicates whether the message flow represents a coordinated transaction for commit/backout processing. This attribute has the following possible values:

- Yes
- No

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Message Flow (Deprecated) Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Msg Flow Type Type of message flow. This attribute has the following possible values:

- n/a
- System (product internal message flow)
- User (user-defined in the Control Center or Message Brokers Toolkit)

Origin Node Name of the managed system.

Processing Nodes Number of processing nodes within a message flow.

Status State of the message flow. This attribute has the following possible values:

- n/a
- Started
- Stopped

Stats Archive Accounting Origin Identification value that indicates what the archive collection data is associated with. User can set this value in node ESQL statement:

```
SET Environment.Broker.Accounting.Origin = "value";
```

Stats Archive Node Data Level Indicates the archive collection of node statistics data for current message flow. This attribute has the following possible values:

- n/a
- None - archive collection contains no node related data
- Basic - archive collection contains node related data
- Advanced - archive collection contains node related and terminal related data

Stats Archive Output Format Indicates the output destination for the statistics archive reports. This attribute has the following possible values:

- n/a
- Usertrace - statistics archive writes "bip" messages to user trace, which can be post-processed in the normal way using the **mqsireadlog** and **mqsiformatlog** commands
- XML - statistics archive reports are generated as XML documents
- SMF - (z/OS only) statistics archive reports are output as SMF type 117 records

Stats Archive Reset Indicates whether archive collection of current message flow is reset. This attribute has the following possible values:

- n/a
- Yes - indicates archive collection is reset
- No - indicates archive collection is not reset

Stats Archive Status Indicates whether message flow accounting and statistics archive level collection is active for current message flow. This attribute has the following possible values:

- n/a
- Active - archive collection is on
- Inactive - archive collection is off

Stats Archive Thread Data Level Indicates whether the archive collection contains thread statistics for current message flow. This attribute has the following possible values:

- n/a
- None - archive collection contains no thread related data
- Basic - archive collection contains thread related data

Stats Snapshot Accounting Origin Identification value that indicates what the snapshot collection data is associated with. User can set this value in node ESQL statement:

```
SET Environment.Broker.Accounting.Origin = "value";
```

Stats Snapshot Node Data Level Indicates the snapshot collection of node statistics for current message flow. This attribute has the following possible values:

- n/a
- None - snapshot collection contains no node related data
- Basic - snapshot collection contains node related data
- Advanced - snapshot collection contains node related and terminal related data

Stats Snapshot Output Format Indicates the output destination for the statistics snapshot reports. This attribute has the following possible values:

- n/a
- Usertrace - statistics snapshot writes "bip" messages to user trace, which can be post processed in the normal way using the **mqsireadlog** and **mqsiformatlog** commands
- XML - statistics snapshot reports are generated as XML documents
- SMF - (z/OS only) statistics snapshot reports are produced as SMF type 117 records

Stats Snapshot Status Indicates whether message flow accounting and statistics snapshot level collection is active for the current message flow. This attribute has the following possible values:

- n/a
- Active - snapshot collection is on
- Inactive - snapshot collection is off

Stats Snapshot Thread Data Level Indicates the snapshot collection of thread statistics for the current message flow. This attribute has the following possible values:

- n/a
- None - snapshot collection contains no thread related data
- Basic - snapshot collection contains thread related data

Trace Filter Filter used to determine which log entries are written to the trace log.

Trace Level Level of trace that is currently active. The attribute has the following possible values:

- None
- Normal
- Debug

User Trace Filter Filter used to determine which log entries are written to the user trace log.

User Trace Level Level of user trace that is currently active. This attribute has the following possible values:

- None
- Normal
- Debug

UUID UUID string that is associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Message Flow Statistics attributes

The Message Flow Statistics attributes provide message flow statistics summarized at the message flow level. Use the Message Flow Statistics attributes to create situations for detecting long processing time in message flows at the message flow level.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Average Flow Time Current average flow time in seconds (up to millisecond accuracy) that it takes to process a message in the message flow. This value reflects the activity over the latest collection interval for the monitoring

agent. This value is taken from the average determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Current Average Message Size Current average size of messages (including headers) going through monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Current Average Message Time Sum of Current Average Queue Time and Current Average Flow Time to give the average total time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Average Queue Time Current average time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored message flow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Current Message Byte Input Rate Current rate of message bytes input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Message Byte Output Rate Current rate of message bytes output from monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Current Message Input Count Current number of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Message Input Rate Current rate of messages input to monitored message flows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Current Message Output Count Current number of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Current Message Output Rate Current rate of messages output from monitored message flows. This value reflects activity over the latest collection interval for the monitoring agent. Output message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

First Message Date and Time Date and time of the first message that goes through a monitored message flow after monitoring begins or statistics are reset.

Last Message Date and Time Date and time of the last message that goes through a monitored message flow.

Message Flow (Deprecated) Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Monitor Nodes Number of CandleMonitor nodes within a message flow.

Origin Node Name of the managed system that is the source of the data for the current row.

Overall Average Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Average Message Size Overall average size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Average Message Time Sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average total time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Average Queue Time Overall average time in seconds (up to millisecond accuracy) that a message remained in a monitored message flow input queue since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Maximum Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Maximum Message Size Overall maximum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Maximum Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Overall Message Byte Input Rate Overall rate of message bytes input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Message Byte Output Rate Overall rate of message bytes output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Message Input Count Overall number of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Message Input Rate Overall rate of messages input to monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of input.

Overall Message Output Count Overall number of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Message Output Rate Overall rate of messages output from monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times.

Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of output.

Overall Minimum Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Minimum Message Size Overall minimum size of messages (including headers) going through monitored message flows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input.

Overall Minimum Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored message flow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input. Note that the value includes any processing time taken by the MQInput node.

Sample Date and Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

Status State of the message flow. This attribute has the following possible values:

- n/a
- Started
- Stopped

Message Flow Topology attributes

The Message Flow Topology attribute group contains attributes used in drawing low-level objects in the message flow topology view. The information in this table can only be used in workspace queries, and cannot be used as conditions in situations. The data in this attribute group is intended only for drawing the topology view in the Tivoli Enterprise Portal.

Attribute 1 The first attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Attribute 2 The second attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Connection Type The type of connection from this node to the next node in the hierarchy. This value is a UTF-8 string of up to 256 alphanumeric characters.

Execution Group The name of the execution group to which the message flow belongs. This value is a UTF-8 string of up to 256 alphanumeric characters.

Message Flow The name of the message flow. This value is a UTF-8 string of up to 256 alphanumeric characters.

Node Label The label of the node. This value is a UTF-8 string of up to 256 alphanumeric characters.

Node Name The name of the node. This value is a UTF-8 string of up to 256 alphanumeric characters.

Node Type The type of the node. This value is the same as UNODE_NAME for predefined nodes, and kqi.UserDefinedNode for user-defined nodes. This value is a UTF-8 string of up to 64 alphanumeric characters.

Node UUID The UUID of the node. Node UUID values might be much longer than the UUID values of other objects. This value is a UTF-8 string of up to 256 alphanumeric characters.

Origin Node The managed system name of the monitored broker. This value is a UTF-8 string of up to 128 alphanumeric characters.

To Node UUID The UUID of the next node in the hierarchy. Node UUID values might be much longer than the UUID values of other objects. This value is a UTF-8 string of up to 256 alphanumeric characters.

To Terminal The terminal identifier of the next node in the hierarchy. This value is a UTF-8 string of up to 256 alphanumeric characters.

Message Processing Node attributes

The Message Processing Node attributes list all attribute values for a message processing node.

Attribute Name (Deprecated) Name of an attribute applicable to this message processing node. Attributes are defined for message processing nodes by the designer of the node to provide user customization of node activity.

Attribute Name Name of an attribute applicable to this message processing node. Attributes are defined for message processing nodes by the designer of the node to provide user customization of node activity. This attribute is encoded in UTF8.

Attribute Value (Deprecated) Value of the named attribute applicable to this message processing node. The attribute value is defined within the Control Center or Message Brokers Toolkit when configuring a node within a message flow.

Attribute Value Value of the named attribute applicable to this message processing node. The attribute value is defined within the Control Center or Message Brokers Toolkit when configuring a node within a message flow. It is encoded in UTF8.

Broker (Deprecated) Identifier of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Identifier of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Message Flow (Deprecated) Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of the message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Origin Node Name of the managed system (in this case, the broker subnode) that is the source of the data for the current row.

Processing Node Label (Deprecated) External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement.

Processing Node Label External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. The node label is encoded in UTF8.

Processing Node Name (Deprecated) Actual long name of the message processing node as it is defined within the Control Center or Message Brokers Toolkit. This name differs from the label name that the user provides for an instance of the node when including it within a message flow.

Processing Node Name Actual long name of the message processing node as it is defined within the Control Center or Message Brokers Toolkit. This name differs from the label name which the user provides for an instance of the node when including it within a message flow. This attribute is encoded in UTF8.

Message Processing Node Information attributes

The Message Processing Node Information attributes display defined Message Processing Nodes by their execution group and message flow.

Attribute 1 (Deprecated) The first attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is.

Attribute 1 The first attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Attribute 2 (Deprecated) The second attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is.

Attribute 2 The second attribute that provides information pertaining to this processing node. The value for this attribute varies depending on what the node is. This attribute is encoded in UTF8.

Attribute 1 Description Short description of which item Attribute 1 is referring to. This attribute has the following possible values:

- n/a
- QueueName
- MessageType
- Type
- ComputeExpression
- Statement
- FilterExpression
- TargetFormat
- ImplicitStreamNaming
- MessageNumber
- Destination
- LabelName
- GenQueueName
- Mode
- MQIpdpEnable
- DebugNodeID
- TraceFile
- AggregateName
- FolderName
- URLSpecifier
- IgnoreTransportFailures
- Port
- MappingExpression
- StylesheetName
- JavaClass
- UniqueIdentifier
- InitialContextFactory
- distributionMode
- eventCoordination
- dataSourceName
- inputDirectory
- outputDirectory
- selectedPort

- asyncRequestCorrelator
- createEnvelope
- envelopeDestination
- name
- adapterComponent
- fromAddress
- asyncResponseCorrelator
- connectionDetails
- hostname
- configurableService
- scriptName

Attribute 2 Description Short description of which item Attribute 2 is referring to. This attribute has the following possible values:

- n/a
- QueueManagerName
- MessageSet
- EventMessage
- DataSource
- SubscriptionPoint
- MessageText
- FilePath
- CollectQueueTime
- Set
- GenQMName
- MQIpdpPort
- OriginalFlowUuid
- MapName
- Trace
- SubFlowName
- TimeoutInterval
- UnsolicitedTimeout
- TimeoutForClient
- TimeoutForServer
- GenerateDefaultHttpHeaders
- Authentication
- StylesheetPath
- RequestPersistence
- OperationMode
- LocationJNDIBindings
- persistenceMode
- sqlQuery
- filenamePattern
- outputFilename
- wsdlFileName

- envelopeLocation
- pathMode
- namespace
- defaultMethod
- toAddresses
- messageSetProperty
- timeoutWaitingForData
- portNumber
- timeoutSendingData
- useEvaluateMethod

Broker (Deprecated) Identifier of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Identifier of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs within the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. This attribute is encoded in UTF8.

Message Flow (Deprecated) Name of the message flow. Message flows are designed by the user within the Control Center to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of the message flow. Message flows are designed by the user within the Control Center to define how messages are to be processed. A message flow can consist of one or more message processing nodes. The message flow name is encoded in UTF8.

Message Flow Type Type of message flow. This attribute has the following possible values:

- n/a
- System - product internal message flow
- User - user-defined in the Control Center or Message Brokers Toolkit

Origin Node Name of the managed system.

Processing Node Label (Deprecated) External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement.

Processing Node Label External name given to an instance of the message processing node when it is included in a message flow. This name is set by the user to identify this message processing node. This name can be set by the user so that the same message processing node type can have a different label name for each message flow placement. The node label is encoded in UTF8.

Processing Node Name (Deprecated) Actual long name of the message processing node as it is defined within the Control Center or Message Brokers Toolkit. This name differs from the label name that the user provides for an instance of the node when including it within a message flow.

Processing Node Name Actual long name of the message processing node as it is defined within the Control Center or Message Brokers Toolkit. This name differs from the label name that the user provides for an instance of the node when including it within a message flow. This attribute is encoded in UTF8.

UUID (Deprecated) UUID string that is associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

UUID UUID string that is associated with an object that uniquely identifies that object. This attribute is provided to assist in problem determination when reviewing trace and event log data.

Multicast Group Statistics attributes

The Multicast Group Statistics attributes display multicast group-level statistics. Use the Multicast Group Statistics attributes to create situations for detecting concerned values in multicast groups.

Broker Name of the broker that generated the statistics report. This attribute is encoded in UTF8.

Bytes Queued Total number of bytes of data that are currently queued by the broker for multicast delivery, for the specified multicast group.

Bytes Resent Total number of bytes that have been resent in multicast messages after the broker was started, for the specified multicast group.

Bytes Resent per Second Rate of bytes that are resent in multicast messages by broker over a specified interval for the specified multicast group.

Bytes Sent Total number of bytes in multicast messages that have been sent by the broker since it is started, for the multicast group named.

Bytes Sent per Second Rate of bytes in multicast messages that are sent by the broker over a specified interval for the specified multicast group.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Execution Group Name of the execution group running on the broker. The execution group name is encoded in UTF8.

Group Address Internet address of the multicast group.

Group Name Name of a multicast group. Multicast group name is encoded in UTF8.

Messages Sent Total number of multicast messages that have been sent by the broker after it was started, for the multicast group named.

Messages Sent per Second Rate of multicast messages that are sent by the broker over a specified interval for the multicast group named.

Origin Node Node name (Managed system name) of the broker.

Query Type Type of query used to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Time Stamp Timestamp for the statistics.

Multicast Summary Statistics attributes

The Multicast Summary Statistics attributes provide multicast summary statistics. There is one row for each execution group in the broker that generates these statistics. Use the Multicast Summary Statistics attributes to create situations for detecting concerned value of multicast groups at the execution group level.

Broker Name of the broker that generated the statistics report. This attribute is encoded in UTF8.

Bytes Queued Total number of bytes of data that are currently queued by the broker for multicast delivery.

Bytes Resent Total number of bytes that have been resent in multicast messages after the broker was started.

Bytes Resent per Second Rate of bytes that are resent in multicast messages by the broker over a specified interval.

Bytes Sent Total number of bytes in multicast messages that have been sent by the broker after it was started.

Bytes Sent per Second Rate of bytes in multicast messages that are sent by the broker over a specified interval.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Execution Group Name of the execution group running on the broker. The execution group name is encoded in UTF8.

Messages Sent Total number of multicast messages that have been sent by the broker after it was started.

Messages Sent per Second Rate of multicast messages that are sent by the broker over a specified interval.

Origin Node Node name (Managed system name) of the broker.

Query Type Type of query used to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Time Stamp Timestamp for the statistics data.

Multicast Topic Statistics attributes

The Multicast Topic Statistics attributes provide multicast topic-level statistics. There is one row per multicast topic per execution group in broker that generates these statistics. Use the Multicast Topic Statistics attributes to create situations for detecting concerned values of multicast topics.

Broker Name of the broker that generated the statistics report. The broker name is encoded in UTF8.

Bytes Queued Total number of bytes of data that are currently queued by the broker for multicast delivery, for the specified multicast topic.

Bytes Resent Total number of bytes that have been resent in multicast messages after the broker was started, for the specified multicast topic.

Bytes Resent per Second Rate of bytes that are resent in multicast messages by the broker over a specified interval for the specified multicast topic.

Bytes Sent Total number of bytes in multicast messages that have been sent by the broker after it was started, for the specified multicast topic.

Bytes Sent per Second Rate of bytes in multicast messages that are sent by the broker over a specified interval for the specified multicast topic.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Execution Group Name of the execution group running on the broker. The execution group name is encoded in UTF8.

Group Name Name of a multicast group. The multicast group name is encoded in UTF8.

Messages Sent Total number of multicast messages that have been sent by the broker after it was started, for the specified multicast topic.

Messages Sent per Second Rate of multicast messages that are sent by the broker over a specified interval for the specified multicast topic.

Origin Node Node name (managed system name) of the broker.

Query Type Type of query used to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1

- Historical (not used, but set in historical data) = 2

Time Stamp Timestamp for the statistics data.

Topic Name Name of the multicast topic. The topic name is encoded in UTF8.

Node Accounting attributes

The Node Accounting attributes provide statistics pertaining to nodes used in message flows. The data might be gathered as snapshots (collected over short intervals) or as archival data (collected over longer intervals). Use the Node Accounting attributes to create situations for detecting concerned values pertaining to nodes in message flows.

Accounting Origin User-assigned identifier associated with a set of message flows sharing a common origin.

Average CPU Microseconds Average CPU time in microseconds spent processing input messages, as an integer. This number is negative if an overflow occurs.

Average CPU Time Average CPU time in microseconds spent processing input messages. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Average Elapsed Microseconds Average elapsed time in microseconds spent processing input messages as an integer. This number is negative if an overflow occurs.

Average Elapsed Time Average elapsed time in microseconds spent processing input messages. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Broker (Deprecated) Name of the broker in alphanumeric format.

Broker Name of the broker in alphanumeric format. This attribute is encoded in UTF8.

Count of Invocations Total number of messages processed by this node.

End Date and Time The end date and time of the collection interval.

Execution Group (Deprecated) Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string.

Execution Group Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string. This attribute is encoded in UTF8.

Invocation Rate Rate per second, to two decimal places, at which the node is invoked during the sampling period.

Maximum CPU Microseconds Maximum CPU time in microseconds spent processing an input message. The valid format is an integer value.

Maximum CPU Time Maximum CPU time in microseconds spent processing an input message. The valid format is an alphanumeric string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Maximum Elapsed Microseconds Maximum elapsed time in microseconds spent processing an input message as an integer value.

Maximum Elapsed Time Maximum elapsed time in microseconds spent processing an input message, in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Message Flow (Deprecated) Name of the message flow.

Message Flow Name of the message flow. This attribute is encoded in UTF8.

Minimum CPU Microseconds Minimum CPU time in microseconds spent processing an input message as an integer value.

Minimum CPU Time Minimum CPU time in microseconds spent processing an input message in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Minimum Elapsed Microseconds Minimum elapsed time in microseconds spent processing an input message as an integer value.

Minimum Elapsed Time Minimum elapsed time in microseconds spent processing an input message in alphanumeric format. The valid format is a string of form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Number of Input Terminals Number of input terminals.

Number of Output Terminals Number of output terminals.

Origin Node Name of the managed system, in this case, the broker subnode name.

Processing Node Label (Deprecated) Name of the node.

Processing Node Label Name of the node. This attribute is encoded in UTF8.

Processing Node Type The type of the node.

Queue Manager Name The name of the queue manager associated with the node. This attribute only contains data if the queue manager name is specified in the node attributes, such as for MQInput and MQOutput nodes.

Queue Name The name of the queue associated with the node. This attribute only contains data if the queue name is specified in the node attributes, such as for MQInput and MQOutput nodes.

Query Type Type of query used in queries to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy
- StatsSettingsMethod
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts (in different tables) as part of the same message flow accounting statistic sample.

Start Date Start date of the sample interval in the format (YYYY-MM-DD). It is not displayed by default.

Start Date and Time Start date and time of the sampling interval.

Total CPU Microseconds Total CPU time in microseconds spent processing input messages, as an integer value.

Total CPU Time Total CPU time in microseconds spent processing input messages, in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Total Elapsed Microseconds Total elapsed time in microseconds spent processing input messages, as an integer value.

Total Elapsed Time Total elapsed time in microseconds spent processing input messages, in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Neighbors attributes

The Neighbors attributes display a list of neighbors to the broker in the publish and subscribe topology, and their connection attributes.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Collective UUID The UUID string that identifies the collective in the domain. The collective is defined in the publish and subscribe topology of the Control Center or Message Brokers Toolkit. It is blank when the neighbor broker is in the same collective as the monitored broker.

Connection Queue Name of the queue in the neighbor broker's queue manager that is used as the basis for publish and subscribe communications. Typically, this value is SYSTEM.BROKER.INTERBROKER.QUEUE.

Connection Queue Manager Name of the queue manager for the neighbor broker. This queue manager is used as the basis for publish and subscribe communications.

Neighbor UUID The UUID string that identifies a neighbor broker in the domain. A neighbor is a broker that is connected to the monitored broker in the publish and subscribe topology of the Control Center or Message Brokers Toolkit. The neighbor UUID can be matched with a broker UUID in the Broker Information table to find the broker that is the neighbor.

Origin Node Name of the managed system.

Product Events attributes

The Product Events attributes display agent-level product events that affect the agent's ability to collect broker data. These events are gathered when they occur, with some events being retained for regular report queries, as set in the agent parameters file. Use the Product Events attributes to create situations for detecting concerned product events.

Remember: Do not use the situations based on Product Events attributes as a mechanism for detecting the status of a broker. Instead, the `Broker_Information.Broker_Status` attribute should be used for this purpose, as demonstrated in the product-provided situations.

Event The action that produced the product event. This attribute has the following possible values:

- `Collector_Initialization_Failed`
- `Collector_Cycle_Processing_Failed`
- `No_Brokers_Discovered_By_Agent`
- `Agent_Shutdown`
- `Queue_Manager_Connection_Failed`
- `Required_Queue_Unavailable`
- `Broker_Request_Message_Put_Failed`

- No_Storage--Message_Not_Processed
- Broker_Product_Not_Installed
- No_Storage--Events_Not_Retained
- No_Storage--User_Data_Request_Affected
- Broker_Not_Started
- No_Storage--Collected_Data_Lost
- Agent_Broker_Data_Requests_Failed
- Broker_Not_Responding
- Broker_Not_Deployed--Data_Unavailable
- Broker_Events_Subscription_Failed
- No_Storage--Agent_Must_Shutdown
- Reply_Queue_Message_Get_Failed
- CandleMonitor_Data_Retrieval_Failed
- CandleMonitor_Node_Not_Deployed
- Agent_Environment_Data_Missing
- Agent_Userid_Group_Missing
- ConfigMgr_Qmgr_Name_Missing
- Table_Query_Missing_Input_Data
- UserId_Not_Authorized_For_Command
- No_Storage--Persistent_Data_Lost
- Message_Length_More_Than_Maximum_Value
- Java_Sub-process_Error
- Java_Sub-process_Timeout
- Java_Sub-process_Operation_Failed
- Java_Sub-process_Operation
- Java_Sub-process-Manager_Operation_Failed
- Java_Sub-process-Manager_Operation

Event Date & Time Date and time when the current event occurs.

Event Qualifier 1 (Deprecated) The first qualifying information that pertains to the event. This information varies according to the event type.

Event Qualifier 1 The first qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Event Qualifier 2 (Deprecated) The second qualifying information that pertains to the event. This information varies according to the event type.

Event Qualifier 2 The second qualifying information that pertains to the event. This information varies according to the event type. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Reason Reason code associated with the problem that produced the product event. This attribute can take the following values:

- n/a
- Alias_Base_Q_Type_Error

- Backed_Out
- Buffer_Error
- Buffer_Length_Error
- Connection_Broken
- Data_Length_Error
- Dynamic_Q_Name_Error
- Expiry_Error
- Feedback_Error
- Get_Inhibited
- Handle_Not_Available
- Hconn_Error
- Hobj_Error
- Syncpoint_Limit_Reached
- Max_Conns_Limit_Reached
- MD_Error
- Missing_Reply_To_Q
- Msg_Type_Error
- Msg_Too_Big_For_Q
- No_Msg_Available
- No_Msg_Under_Cursor
- Not_Authorized
- Not_Open_For_Browse
- Not_Open_For_Input
- Not_Open_For_Output
- Object_Changed
- Object_In_Use
- Object_Type_Error
- OD_Error
- Option_Not_Valid_For_Type
- Options_Error
- Persistence_Error
- Persistence_Not_Allowed
- Priority_Exceeds_Maximum
- Priority_Error
- Put_Inhibited
- Q_Deleted
- Q_Full
- Q_Space_Not_Available
- Q_Type_Error
- Q_Mgr_Name_Error
- Q_Mgr_Not_Available
- Report_Options_Error
- Second_Mark_Not_Allowed
- Security_Error
- Signal_Outstanding

- Signal_Request_Accepted
- Storage_Not_Available
- Syncpoint_Not_Available
- Truncated_Msg_Accepted
- Truncated_Msg_Failed
- Unknown_Alias_Base_Q
- Unknown_Object_Name
- Unknown_Object_Q_Mgr
- Unknown_Remote_Q_Mgr
- Wait_Interval_Error
- Xmit_Q_Type_Error
- Xmit_Q_Usage_Error
- Not_Open_For_Pass_All
- Not_Open_For_Pass_Ident
- Not_Open_For_Set_All
- Not_Open_For_Set_Ident
- Context_Handle_Error
- Context_Not_Available
- Signal1_Error
- Object_Already_Exists
- Object_Damaged
- Resource_Problem
- Another_Q_Mgr_Connected
- Unknown_Report_Option
- Storage_Class_Error
- COD_Not_Valid_For_XCF_Q
- Suppressed_By_Exit
- Format_Error
- Source_CCSID_Error
- Source_Integer_Enc_Error
- Source_Decimal_Enc_Error
- Source_Float_Enc_Error
- Target_CCSID_Error
- Target_Integer_Enc_Error
- Target_Decimal_Enc_Error
- Target_Float_Enc_Error
- Not_Converted
- Converted_Msg_Too_Big
- Adapter_Storage_Shortage
- Adapter_Conn_Load_Error
- Adapter_Serv_Load_Error
- Adapter_Defs_Error
- Adapter_Defs_Load_Error
- Adapter_Conv_Load_Error
- DH_Error

- Multiple_Reasons
- Open_Failed
- CNO_Error
- CICS_Wait_Failed
- Object_Name_Error
- Object_Q_Mgr_Name_Error
- Recs_Present_Error
- Object_Records_Error
- Response_Records_Error
- ASID_Mismatch
- PMO_Record_Flags_Error
- Put_Msg_Records_Error
- Conn_Id_In_User
- Q_Mgr_Quiescing
- Q_Mgr_Stopping
- Duplicate_Recov_Coord
- PMO_Error
- API_Exit_Load_Error
- Remote_Q_Name_Error
- Inconsistent_Persistence
- GMO_Error
- Pageset_Full
- Pageset_Error
- Name_Not_Valid_For_Type
- Unexpected_Error
- Unknown_Xmit_Q
- Unknown_Def_Xmit_Q
- Def_Xmit_Q_Type_Error
- Def_Xmit_Q_Usage_Error
- Name_In_Use
- Connection_Quiescing
- Connection_Stopping
- Adapter_Not_Available
- No_Msg_Locked
- Connection_Not_Authorized
- Call_In_Progress
- Incomplete_Group
- Incomplete_Msg
- Inconsistent_CCSIDs
- Inconsistent_Encodings
- Inconsistent_UOW
- Invalid_Msg_Under_Cursor
- Match_Options_Error
- MDE_Error
- Msg_Flags_Error

- Msg_Seq_Number_Error
- Offset_Error
- Original_Length_Error
- Segment_Length_Zero
- UOW_Not_Available
- Wrong_GMO_Version
- Wrong_MD_Version
- Group_Id_Error
- Inconsistent_Browse
- RFH_Command_Error
- RFH_Parm_Error
- RFH_Duplicate_Parm
- RFH_Parm_Missing
- Topic_Error
- Not_Registered
- Q_Mgr_Name_Error
- Q_Name_Error
- Incorrect_Q
- Correl_ID_Error
- Not_Authorized
- Reg_Options_Error
- Filter_Error
- Wrong_User
- Take_Action_Auth_Users_Parameters
- Object_Unusable
- Thread_Create_Error
- Associated_Object_Error
- Unknown_Failure
- Storage_Depleted
- XML_Parse_Init_Failure
- XML_Parse_Error
- XML_Message_Type_Error
- Object_Construction_Error
- QMgr_Connection_Thread_Unavailable
- QMgr_Connection_Object_Missing
- QMgr_Pipe_Communication_Error
- Task_Ending_Cannot_Process
- Task_Request_Bad
- Too_Many_Get_Signal_Requests
- Task_Attach_Failed
- Message_Flow_Definitions
- Dynamic_Subscription_Engine_Data
- Message_Flow_Statistics
- Broker_Events
- Message_Flow_Events

- Product_Events
- Execution_Group_Information
- Message_Flow_Information
- Message_Processing_Node_Information
- Neighbors
- Subscriptions
- Retained_Publications
- ACL_Entries
- Execution_Group_Statistics
- Message_Flow_Statistics
- CandleMonitor_Node_Statistics
- Collected_String_Lost
- CandleMonitor_Data_Lost
- User_Command
- Message_Processing_Node_Attributes
- Sub-Flow_Statistics
- User_Statistics
- Message_Flow_Accounting
- PubSub_Statistics
- Message_Flow_CMP
- ATTRIBUTE_PARTIAL
- FAILED_FOUND_CHILD
- FAILED_CREATE_CHILD
- FAILED_GET_CHILD
- FAILED_BROKER_CREATE
- FAILED_BROKER_GET_INSTANCE

Publish-Subscribe Statistics attributes

The Publish-Subscribe Statistics attributes provide a list of Publish and Subscribe statistics, which does not include multicast data. Use the Publish-Subscribe Statistics attributes to create situations for detecting concerned values in publishing and subscribing.

Broker Identifies the name of the broker that generated the statistics report. This attribute is encoded in UTF8.

Client Bytes Cut Through Total number of bytes of data that are sent immediately to clients, without being queued internally by the broker.

Client Bytes Cut Through per Second Rate of bytes per second that are sent immediately to clients over a specified interval without being queued internally by the broker.

Client Bytes Dropped Number of bytes that are dropped due to queue overflow, where the client is not subsequently disconnected from the broker.

Client Bytes Dropped per Second Rate of bytes per second that are dropped over a specified interval because of queue overflow, where the client is not subsequently disconnected from the broker.

Client Bytes Queued Number of bytes of data that are currently queued by the broker for delivery to clients.

Client Bytes Received Total number of bytes that are received by the broker from its clients.

Client Bytes Received per Second Rate of bytes per second that are received by the broker from its clients over a specified interval.

Client Bytes Sent Total number of bytes that the broker has delivered to its clients.

Client Bytes Sent per Second Rate of bytes per second that the broker has delivered to its clients over a specified interval.

Client Count Total number of clients that are connected to the broker.

Client Disconnect Bytes Dropped Number of bytes that are dropped due to queue overflow, where the client is subsequently disconnected from the broker.

Client Disconnect Bytes Dropped per Second Rate of bytes per second that are dropped over a specified interval because of queue overflow, where the client is subsequently disconnected from the broker.

Client Disconnect Messages Dropped Number of messages that are dropped due to queue overflow, where the client is subsequently disconnected from the broker.

Client Disconnect Messages Dropped per Second Rate of messages per second that are dropped over a specified interval because of queue overflow, where the client is subsequently disconnected from the broker.

Client Messages Dropped Number of messages that are dropped due to queue overflow, where the client is not subsequently disconnected from the broker.

Client Messages Dropped per Second Rate of messages per second that are dropped due to queue overflow, where the client is not subsequently disconnected from the broker, over a specified interval.

Client Messages Received Total number of messages that are received by the broker from its clients.

Client Messages Received per Second Rate of messages per second that are received by the broker from its clients over a specified interval.

Client Messages Sent Total number of messages that the broker has delivered to its clients.

Client Messages Sent per Second Rate of messages per second that the broker has delivered to its clients over a specified interval.

Current Interval Length of time in milliseconds of the latest collection interval that the displayed statistics labeled Current reflect.

Execution Group Name of the execution group running on the broker. This attribute is encoded in UTF8.

Neighbor Bytes Cut Through Total number of bytes of data that are sent immediately to neighboring brokers without being queued internally by the broker.

Neighbor Bytes Cut Through per Second Rate of bytes per second that are sent immediately to neighboring brokers over a specified interval, without being queued internally by the broker.

Neighbor Bytes Dropped Number of bytes that are dropped due to queue overflow, where the neighboring broker is not subsequently disconnected from the broker.

Neighbor Bytes Dropped per Second Rate of bytes per second that are dropped due to queue overflow, where the neighboring broker is not subsequently disconnected from the broker, over a specified interval.

Neighbor Bytes Queued Number of bytes of data that are currently queued by the broker for delivery to neighboring brokers.

Neighbor Bytes Received Total number of bytes that are received by the broker from its neighboring brokers.

Neighbor Bytes Received per Second Rate of bytes per second that are received by the broker from its neighboring brokers over a specified interval.

Neighbor Bytes Sent Total number of bytes that the broker has delivered to its neighboring brokers.

Neighbor Bytes Sent per Second Rate of bytes per second that the broker has delivered to its neighboring brokers over a specified interval.

Neighbor Count Total number of neighbor brokers that are connected to the broker.

Neighbor Disconnect Bytes Dropped Number of bytes that are dropped due to queue overflow, where the neighboring broker is subsequently disconnected from the broker.

Neighbor Disconnect Bytes Dropped per Second Rate of bytes per second that are dropped due to queue overflow, where the neighboring broker is subsequently disconnected from the broker, over a specified interval.

Neighbor Disconnect Messages Dropped Number of messages that are dropped due to queue overflow, where the neighboring broker is subsequently disconnected from the broker.

Neighbor Disconnect Messages Dropped per Second Rate of messages per second that are dropped due to queue overflow, where the neighboring broker is subsequently disconnected from the broker, over a specified interval.

Neighbor Messages Dropped Number of messages that are dropped due to queue overflow, where the neighboring broker is not subsequently disconnected from the broker.

Neighbor Messages Dropped per Second Rate of messages per second that are dropped due to queue overflow, where the neighboring broker is not subsequently disconnected from the broker, over a specified interval.

Neighbor Messages Received Displays the total number of messages that are received by the broker from its neighboring brokers.

Neighbor Messages Received per Second Rate of messages per second that are received by the broker from its neighboring brokers over a specified interval.

Neighbor Messages Sent Total number of messages that the broker has delivered to its neighboring brokers.

Neighbor Messages Sent per Second Rate of messages per second that the broker has delivered to its neighboring brokers over a specified interval.

Origin Node Node name (managed system name) of the broker.

Query Type Type of query used to get the data required. It is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Time Stamp Timestamp for the statistics data.

Subscription Count Number of subscriptions held by the broker.

Retained Publications attributes

The Retained Publications attributes provide a list of topics and subscription points that have retained publications.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Origin Node Name of the managed system.

Subscription Point (Deprecated) Subscription point name that refers to a publication node within a message flow. The publication node has its subscription point attribute set to this name. A blank column indicates that the default subscription point (publication node) is being used.

Subscription Point Subscription point name that refers to a publication node within a message flow. The publication node has its subscription point attribute set to this name. A blank column indicates that the default subscription point (publication node) is being used. This attribute is encoded in UTF8.

Topic (Deprecated) A descriptive name of published data that one can subscribe to. Topic names for retained publications are full names without wildcard characters and therefore identify only one topic.

Topic A descriptive name of published data that one can subscribe to. Topic names for retained publications are full names without wildcard characters and therefore identify only one topic. This attribute is encoded in UTF8.

Sub-Flow Statistics attributes

The Sub-Flow Statistics attribute group provides data for only that section of the message flow that is the part of a subflow. Sub-Flow Statistics attributes are always provided with respect to the particular message flow within an execution group to which the subflow is a part. Use the Sub-Flow Statistics attributes to create situations for detecting concerned values in subflows.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent targets specific brokers for monitoring because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent targets specific brokers for monitoring because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Current Average Flow Time Current average in seconds (up to millisecond accuracy) that it takes to process a message in the subflow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the average determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Current Average Message Size Current average size of messages (including headers) going through monitored subflows. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Current Average Message Time Sum of Current Average Queue Time and Current Average Flow Time to give the average total time in seconds (up to millisecond accuracy) that it takes to process a message over the latest collection interval.

Current Average Queue Time Current average time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored subflow. This value reflects the activity over the latest collection interval for the monitoring agent. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Current Interval Length of time in seconds of the latest collection interval that the displayed statistics labeled Current reflect.

Current Message Byte Input Rate Current rate of message bytes input to monitored subflows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Current Message Byte Output Rate Current rate of message bytes output from monitored subflows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message byte rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Current Message Input Count Current number of messages input to monitored subflows. This value reflects the activity over the latest collection interval for the

monitoring agent. Input messages are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Current Message Input Rate Current rate of messages input to monitored subflows. This value reflects the activity over the latest collection interval for the monitoring agent. Input message rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Current Message Output Count Current number of messages output from monitored subflows. This value reflects the activity over the latest collection interval for the monitoring agent. Output messages are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Current Message Output Rate Current rate of messages output from monitored subflows. This value reflects the activity over the latest collection interval for the monitoring agent. Output message rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Execution Group (Deprecated) Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit.

Execution Group Name of an execution group. An execution group is a process that runs in the broker and provides the execution environment in which message flows run. Users associate message flows with execution groups during message flow assignment in the Control Center or Message Brokers Toolkit. The execution group name is encoded in UTF8.

First Message Date and Time Date and time of the first message that goes through a monitored subflow after monitoring begins or statistics are reset.

Last Message Date and Time Date and time of the last message that goes through a monitored subflow.

Message Flow (Deprecated) Name of a message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes.

Message Flow Name of a message flow. Message flows are designed by the user to define how messages are to be processed. A message flow can consist of one or more message processing nodes. This attribute is encoded in UTF8.

Monitor Nodes Number of CandleMonitor nodes within a subflow.

Origin Node Name of the managed system.

Overall Average Flow Time Overall average time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Average Message Size Overall average size of messages (including headers) going through monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Average Message Time Sum of Overall Average Queue Time and Overall Average Flow Time to give the overall average total time in seconds (up to millisecond accuracy) taken to process a message since the last reset of statistics.

Overall Average Queue Time Overall average number of seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Overall Maximum Flow Time Overall maximum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Maximum Message Size Overall maximum size of messages (including headers) going through monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the maximums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Maximum Queue Time Overall maximum time in seconds (up to millisecond accuracy) that a message remains in the input queue for a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the averages determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Overall Message Byte Input Rate Overall rate of message bytes input to monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message byte rates are determined by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Overall Message Byte Output Rate Overall rate of message bytes output from monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output message bytes are counted by CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Overall Message Input Count Overall number of messages input to monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input

messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Overall Message Input Rate Overall rate of messages input to monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Input message rates are determined by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowInput or input.

Overall Message Output Count Overall number of messages output from monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Overall Message Output Rate Overall rate of messages output from monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. Output messages are counted by the CandleMonitor nodes that are placed in message flows with a **type** attribute of subFlowOutput or output.

Overall Minimum Flow Time Overall minimum time in seconds (up to millisecond accuracy) that it takes to process a message in a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Minimum Message Size Overall minimum size of messages (including headers) going through monitored subflows since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within the CandleMonitor nodes that are placed in the message flows to delineate the subflows.

Overall Minimum Queue Time Overall minimum time in seconds (up to millisecond accuracy) that a message remained in the input queue for a monitored subflow since the reset of statistics. This value reflects the activity over the entire monitoring period given by the first and last message date and times. This value is taken from the minimums determined within CandleMonitor nodes that are placed in the message flows with a **type** attribute of input only. Note that the value includes any processing time taken by the MQInput node.

Sample Date and Time Date and time of the latest sample taken of CandleMonitor node data for producing the displayed statistics.

Sub-Flow Name (Deprecated) Name given to the subflow by the message flow developer. The name is assigned by setting the subFlowName attribute for a CandleMonitor node when placing it into a message flow.

Sub-Flow Name Name given to the subflow by the message flow developer. The name is assigned by setting the subFlowName attribute for a CandleMonitor node when placing it into a message flow. This attribute is encoded in UTF8.

Statistics User Encrypted logon ID for user statistics queries. This attribute is not displayed.

Subscriptions attributes

The Subscriptions attributes display a list of subscriptions to published data topics.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

Client Identity (Deprecated) Identifier of the client destination for the subscription.

Client Identity Identifier of the client destination for the subscription. This attribute is encoded in UTF8.

Create Date and Time Date and time that the subscription is created.

Expiration Date and Time Date and time that the subscription is due to expire. When this field is blank, no expiration date is specified.

Filter (Deprecated) Content filter SQL expression that qualifies which publication is applicable to the subscription. The publication must match this filter to be sent to the subscribing application. A blank filter indicates that no message content filtering is taking place for the subscription.

Filter Content filter SQL expression that qualifies which publication is applicable to the subscription. The publication must match this filter to be sent to the subscribing application. A blank filter indicates that no message content filtering is taking place for the subscription. This attribute is encoded in UTF8.

Inform If Retained Whether the broker should inform the subscriber when a published message is a retained publication. This attribute has the following possible values:

- Yes
- No

Local Only Whether the subscription is local to the monitored broker. This attribute has the following possible values:

- Yes (the subscription is local and will not be sent to other brokers)
- No (the subscription is sent to other brokers in the publish and subscribe topology)

Origin Node Name of the managed system.

Persistent Whether publications are delivered as persistent messages. This attribute has the following possible values:

- Yes
- No
- AsPublish (persistence is set by the publisher)
- AsQueue (persistence is set by the queue definition)

Publish On Request Only Whether the subscriber wants to receive messages only on request for an update. This attribute has the following possible values:

- Yes
- No

Subscription Point (Deprecated) Subscription point name that refers to a publication node within a message flow. The publication node has its subscription point attribute set to this name. A blank column indicates that the default subscription point (publication node) is being used.

Subscription Point Subscription point name that refers to a publication node within a message flow. The publication node has its subscription point attribute set to this name. A blank column indicates that the default subscription point (publication node) is being used. This attribute is encoded in UTF8.

Topic (Deprecated) Descriptive name of published data that one can subscribe on. Topic names for subscriptions and ACL entries can contain wildcard characters (as defined by topic name rules) so that they can identify more than one topic.

Topic Descriptive name of published data that one can subscribe to. Topic names for subscriptions and ACL entries can contain wildcard characters (as defined by topic name rules) so that they can identify more than one topic. This attribute is encoded in UTF8.

User Identifier (Deprecated) User identifier of the subscribing application.

User Identifier User identifier of the subscribing application. This attribute is encoded in UTF8.

Terminal Accounting attributes

The Terminal Accounting attributes provide statistics pertaining to terminals on nodes used in message flows of brokers. The data might be gathered as snapshots (collected over short intervals) or as archival data (collected over longer intervals). Use the Terminal Accounting attributes to create situations for detecting concerned values of terminals.

Accounting Origin User-assigned identifier associated with a set of message flows that share a common origin.

Broker (Deprecated) Name of the broker in alphanumeric format.

Broker Name of the broker in alphanumeric format. This attribute is encoded in UTF8.

Count of Invocations Total number of messages processed by this node.

End Date and Time The end date and time of the collection interval.

Execution Group (Deprecated) Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string.

Execution Group Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string. This attribute is encoded in UTF8.

Invocation Rate Rate per second, to two decimal places, at which the terminal is invoked during the sampling period.

Message Flow (Deprecated) Name of the message flow.

Message Flow Name Name of the message flow. This attribute is encoded in UTF8.

Origin Node Name of the managed system, in this case, the broker subnode name.

Processing Node Label (Deprecated) Name of node (label).

Processing Node Label Name of node (label). This attribute is encoded in UTF8.

Query Type Type of query used to get the data required. This attribute is not displayed by default. This attribute has the following possible values:

- Current = 0
- Recent = 1
- Historical (not used, but set in historical data) = 2

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy
- StatsSettingsMethod
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts (in different tables) as part of the same message flow accounting statistic sample.

Start Date and Time Start date and time of the sampling interval.

Terminal Label (Deprecated) Name of the terminal.

Terminal Label Name of the terminal. This attribute is encoded in UTF8.

Terminal Type Type of terminal, either Input or Output.

Thread Accounting attributes

The Thread Accounting attributes provide the statistics gathered by WebSphere Business Integration brokers for threads used by message flows. The data might be gathered as snapshots (collected over short intervals) or as archival data (collected over longer intervals). Use the Thread Accounting attributes to create situations for detecting concerned values in threads.

Accounting Origin User-assigned identifier associated with a set of message flows that share a common origin.

Average CPU Microseconds Average CPU time, in microseconds, which is spent processing input messages, as an integer. This number is negative if overflow occurs.

Average CPU Time Average CPU time, in microseconds, spent processing input messages. The valid format is a string of the form sss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Average Elapsed Microseconds Average elapsed time, in microseconds, spent processing input messages as an integer. This number is negative if overflow occurs.

Average Elapsed Time Average elapsed time, in microseconds, spent processing input messages. The valid format is a string of form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Average Input Message Size Average size, in bytes, of input messages as an integer value. This number is negative if overflow occurs.

Average Input Message Size String Average size, in bytes, of input messages in character format in case of overflow.

Broker (Deprecated) Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level.

Broker Name of the broker. The WebSphere Message Broker Monitoring agent specifically targets brokers for monitoring, because all message processing is performed at the broker level. This attribute is encoded in UTF8.

End Date and Time The end date and time of the collection interval.

This attribute is designed for logging and reporting data collection times rather than for creating situations. To specify a time of day for monitoring, use attributes from the Universal Time or Local Time groups.

Execution Group (Deprecated) Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string.

Execution Group Name of the execution group that the message flow belongs to. The valid format is an alphanumeric string. This attribute is encoded in UTF8.

Input Byte Rate Rate in bytes per second, to two decimal places, at which the thread processed input message bytes during the sampling period.

Input Message Rate Rate in messages per second, to two decimal places, at which the thread processed input messages during the sampling period.

Input Message Wait CPU Microseconds Total CPU time in microseconds spent waiting for input messages as an integer value.

Input Message Wait CPU Time Total CPU time in microseconds spent waiting for input messages in alphanumeric format. The valid format is a string of form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds

- nnn = microseconds

Input Message Wait Elapsed Microseconds Total elapsed time in microseconds waiting for input messages in integer format.

Input Message Wait Elapsed Time Total elapsed time in microseconds spent waiting for input messages in alphanumeric format. The valid format is a string of the form ssss:mmm:nnn, where:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Maximum Input Message Size Maximum message size in bytes in integer format.

Maximum Input Message Size String Maximum message size in bytes in alphanumeric format.

Message Flow (Deprecated) Name of the message flow.

Message Flow Name of the message flow. This attribute is encoded in UTF8.

Minimum Input Message Size Minimum message size in bytes as an integer value.

Minimum Input Message Size String Minimum message size in bytes in alphanumeric format.

Origin Node Name of the managed system. In this case, it is the broker subnode name.

Query Type Type of query used to get the data required. This attribute is not displayed by default. It has the following possible values:

- ssss = seconds
- mmm = milliseconds
- nnn = microseconds

Record Code Reason for the output. This attribute has the following possible values:

- MajorInterval
- Snapshot
- Shutdown
- ReDeploy
- StatsSettingsMethod
- Stop

Record Type Type of output, either Archive or Snapshot.

Sample Handle Handle that identifies all parts as part of the same message flow accounting statistic sample.

Start Date and Time Start date and time of the sampling interval.

Thread Number Relative thread number in pool.

Total CPU Microseconds Total CPU time in microseconds spent processing input messages, as an integer value.

Total CPU Time Total CPU time in microseconds spent processing input messages, in alphanumeric format. The valid format is a string of the form `ssss:mmm:nnn`, where:

- `ssss` = seconds
- `mmm` = milliseconds
- `nnn` = microseconds

Total Elapsed Microseconds Total elapsed time in microseconds spent processing input messages, as an integer value.

Total Elapsed Time Total elapsed time in microseconds spent processing input messages, in alphanumeric format. The valid format is a string of the form `ssss:mmm:nnn`, where:

- `ssss` = seconds
- `mmm` = milliseconds
- `nnn` = microseconds

Total Input Message Size Total size in bytes of input messages, as an integer value.

Total Input Message Size String Total size in bytes of input messages in alphanumeric format.

Total Input Messages Total number of messages processed. The valid format is an integer.

CandleMonitor node

This section introduces the attributes and configuration variables related to the CandleMonitor node.

Attributes

You can set the attributes of the CandleMonitor node when you place it in a message flow. The attribute values that you set determine how the node operates.

type

The **type** attribute specifies the type of statistics that are gathered. Valid values are:

- `input`
Identifies the node as a main message flow input node that records the entry of messages into the message flow. For example, specify the **type** attribute to `input` when placing the CandleMonitor node immediately after the MQInput node.
- `output`
Identifies the node as a main message flow output node that records the exit of messages from the message flow. For example, specify the **type** attribute to `output` when placing the CandleMonitor node immediately in front of MQOutput, MQReply, or Publication node.
- `subFlowInput`
Identifies the node as a subflow input node
- `subFlowOutput`

Identifies the node as a subflow output node

- other

Identifies the node as being used for other purposes, such as temporary collection of statistics or message flow event generation. For example, specify the **type** attribute to other when placing the CandleMonitor node wherever necessary to debug your message flows.

collectQueueTime

The **collectQueueTime** attribute specifies if the CandleMonitor node collects queue times. Valid values are:

- no
Queue times are not collected for these messages.
- yes
Queue times are collected for these messages.

Queue times are calculated by using the *put date and time* (the time and date at which the message was put into the queue) of the messages in the queue. If the input queue to a message flow has messages with put date and times that do not reflect accurately when the message was put into the input queue, set the **collectQueueTime** attribute to no so that queue times are not collected for these messages. (See also “Monitoring the input or output of a message flow” on page 49.)

eventMessage

The **eventMessage** attribute is used to produce user-defined message flow events that can trigger Message Flow Events situations and reports. If this attribute is set to anything other than cleared (“”), an event is produced when a message enters the CandleMonitor node.

When a CandleMonitor node is placed for generating events, set the **type** attribute to other and the **eventMessage** attribute to the message text that appears in Tivoli Enterprise Portal workspaces.

By default this value is cleared (“”) and no events are produced.

subFlowName

The **subFlowName** attribute specifies an identifying name for the subflow. This attribute is required for CandleMonitor nodes which are of type subFlowInput and subFlowOutput; otherwise, these nodes cannot produce correct subflow statistics without the **subFlowName** attribute specified.

You can specify the **subFlowName** attribute for other types of nodes. The following table summarizes the effect of specifying the **subFlowName** attribute for each node type.

Table 18. The effect of providing a subFlowName attribute for each node type

Value of the type attribute	subFlowName attribute requirement	Effect
input	Optional	The node is a combination node; it marks not only the beginning of a message flow, but also the beginning of a subflow in the message flow.
output	Optional	The node is a combination node; it marks not only the end of a message flow, but also the end of a subflow within the message flow.

Table 18. The effect of providing a subFlowName attribute for each node type (continued)

Value of the type attribute	subFlowName attribute requirement	Effect
other	Optional	The node is a part of the subflow. This combination of settings has little effect. However, it causes the CandleMonitor node to be displayed in the list of CandleMonitor nodes for the named subflow. Statistics for this node are displayed in the Monitor Node Base Statistics workspace/ CandleMonitor Node Statistics workspace.
subFlowInput	Required	The subFlowInput type node marks the beginning of a monitored subflow and requires a name so that statistical data is correlated with the subflow. There can be multiple inputs to a subflow; for multiple inputs, use the same subFlowName for each node.
subFlowOutput	Required	The subFlowOutput type node marks the end of a monitored subflow and requires a name so that statistical data is correlated with the subflow. There can be multiple outputs for a subflow; for multiple outputs, use the same subFlowName for each node.

For more information about placing CandleMonitor nodes and required attribute values, see “Monitoring subflows” on page 50.

activateNode

The **activateNode** attribute controls the activation of instances of a CandleMonitor node. Valid values are:

- yes
The CandleMonitor node is active. This setting is appropriate for a test environment.
- no
The CandleMonitor node is not active.
- eventOnly
The CandleMonitor node is activated only if the eventMessage attribute is specified. The node does not produce statistics; it produces only message flow events. This setting is useful when you are interested in message flow events, but you do not want to gather statistics.
- inputOutputOnly
The CandleMonitor node is activated if the node type is input or output. This setting produces statistics (and events, if the eventMessage attribute is set).
- inputOutputAndEventOnly
The CandleMonitor node is activated if the node type is input or output, or if the eventMessage attribute is set. Statistics are produced; events are produced also if the eventMessage attribute is set. This setting is useful in a production environment when you want message flow statistics and event messages, but you do not want any of the additional details provided by other node types.

If you are monitoring WebSphere Business Integration broker 5.0 or later, you can configure the activateNode attribute. Set this attribute when you are configuring bar files, so the attribute can have a different setting for different bar files in test and production environments. In addition, with version 5.0 you can promote the attribute setting so that this attribute has the same setting for each CandleMonitor node in a message flow.

The set of possible values for the attribute supports specification of some general rules for when the node must be active; for example, it can be active for nodes of type input and output only. In general, when you are customizing a CandleMonitor node in a message flow, use the default value (yes) to facilitate node usage during testing. And then, when you are deploying the node to a production broker, configure the bar file setting for the **activateNode** attribute to the level that you want and promote the attribute to automatically change the settings of all nodes in the flow to that activation level.

You can specify a runtime override setting for the **activateNode** attribute, which allows different activation levels for different brokers (see “KQIActivateNode” on page 262).

Configuration variables

The following sections provide a detailed description of some configuration variables.

KQIMemorySize

The **KQIMemorySize** variable specifies the size (in bytes) of a shared memory segment created by the plug-in. The CandleMonitor node creates three of these shared memory segments for holding different types of data. The default value is 32768 bytes. Do not modify this variable unless instructed to do so by the IBM Software Support.

KQITempDirectory

The **KQITempDirectory** variable is not applicable on Windows systems.

The **KQITempDirectory** variable specifies the directory to be used for shared memory and mutexes. The broker and the agent must have the required authority to read, write and create files in this directory. Its default value is /tmp.

By default, temporary files used by the CandleMonitor node and related to mutexes and shared memory are stored in the /tmp directory. The user IDs used by the broker and the WebSphere Message Broker Monitoring agent must both have write access to this directory. If the user IDs do not have this access, you can use the **KQITempDirectory** variable in the `kqi.xml` file to specify a different directory for storing this information.

You must also use this variable to specify a different location if there is a possibility that the /tmp directory becomes full or when the /tmp directory is used by a large number of applications, because this might result in files being accidentally overwritten by other programs.

The only safe time to delete files with file names containing **kqi** from the directory specified in the **KQITempDirectory** variable is after you shut down all brokers and WebSphere Message Broker Monitoring agents on the system. The **KQITempDirectory** variable is used by all brokers and WebSphere Message Broker Monitoring agents on the system, so they must all be stopped before you can delete any files that have **kqi** in the name. Under typical circumstances, you do not need to delete the files because when all the brokers and agents are stopped, they are not present. However, in exceptional circumstances where the files still exist after all the components have stopped, you can delete the files manually. These files implement shared memory and mutex, and WebSphere Message Broker Monitoring agent might not behave normally if you delete them while the processes that use the files are active.

KQINodeTrace

The **KQINodeTrace** variable determines whether `kqipnode.lil` tracing is turned on. Valid values are Off and On. The default is Off. For tracing to occur, the broker normal tracing must be turned on for one or more of the applicable broker, execution group, message flow. Set this value to On only when instructed to do so by the IBM Software Support.

KQIActivateNode

Use this variable to override the CandleMonitor node **activateNode** attribute of any broker on the system. If a value other than the default `NoOverride` is specified, that value overrides the value set for the attribute for all CandleMonitor nodes deployed on a broker.

The possible values are:

- **NoOverride**
The value of the **activateNode** attribute that is deployed for each instance of the CandleMonitor node is honored; no override takes place. This is the default setting.
- **Yes**
Every instance of the CandleMonitor node is active.
- **No**
All instances of the CandleMonitor node are inactive. No statistics or message flow events are generated.
- **EventOnly**
An instance of the CandleMonitor node is active only if the **eventMessage** attribute is assigned. No statistics are produced; only message flow events are produced.
- **InputOutputOnly**
An instance of the CandleMonitor node is active only if the type is input or output. Statistics are generated (and events, if the **eventMessage** attribute is assigned).
- **InputOutputAndEventOnly**
Instances of the CandleMonitor node are active only if they are of type input or output, or if the **eventMessage** attribute is assigned. Statistics are produced, and message flow events are produced, if the **eventMessage** attribute is assigned.

KQIActivateNodeForBROKERNAME

Use this variable to override the CandleMonitor node **activateNode** attribute of any one broker on the system. If this variable is specified, the CandleMonitor node that is running in the *BROKERNAME* broker uses this setting instead of the setting specified in the **KQIActivateNode** attribute.

Important: This variable name does not exist by default; you must add it to the `kqipnode.cfg` file (UNIX, Linux, or z/OS systems) or type it in the variable field instead of selecting it from the list (Windows systems).

Take Action commands

You can issue the commands from the Take Action view, from the situation window when a situation becomes true, from the Navigator, or from a row in a table view.

The predefined Take Action commands can be divided into two categories, the commands for the agent-managed systems and the commands for the broker-managed systems. The names of commands for the agent-managed systems begin with the letters **QI**. The commands for the broker-managed systems have different names depending on the broker versions. The command names begin with the letters **WMB** for brokers of version 7.0 or later, and **QI** for brokers of previous versions.

The WebSphere Message Broker Monitoring agent provides the following Take Action commands for brokers of version 7.0 or later:

- Commands for agent-managed systems:
 - QI Change Trace Component
 - QI Start Component
 - QI Stop Component
 - QI Stop Component and its Queue Manager
- Commands for broker-managed systems:
 - WMB Change Broker
 - WMB Change Flow Stats
 - WMB Change Properties
 - WMB Change Trace Broker
 - WMB Create User Statistics
 - WMB Destroy User Statistics
 - WMB Refresh Broker Data
 - WMB Reset Statistics
 - WMB Sample User Statistics
 - WMB Start Broker
 - WMB Start Message Flow(s)
 - WMB Stop Broker
 - WMB Stop Broker and its Queue Manager
 - WMB Stop Message Flow(s)

The WebSphere Message Broker Monitoring agent provides the following Take Action commands for the brokers that are earlier than version 7.0:

- Commands for agent-managed systems:
 - QI Change Trace Component
 - QI Start Component
 - QI Stop Component
 - QI Stop Component and its Queue Manager
- Commands for broker-managed systems:
 - QI Change Broker
 - QI Change Flow Stats
 - QI Change Properties
 - QI Change Trace Broker
 - QI Create User Statistics
 - QI Delete Retained Publication
 - QI Destroy User Statistics
 - QI Refresh Broker Data

- QI Reset Statistics
- QI Sample User Statistics
- QI Start Broker
- QI Start Message Flow(s)
- QI Stop Broker
- QI Stop Broker and its Queue Manager
- QI Stop Message Flow(s)

Commands for agent managed systems

The Take Action commands for agent managed systems are described in the section.

Remember: The name of the component in the commands can be a broker name, a configuration manager name, or the name of a user name server, unless otherwise noted.

QI Start Component command

This command starts a broker product component.

Identify the component that you want to start by specifying its name in the **Value** column of Edit Argument window. The destination of this command must be an agent-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiStart 'ComponentName'
```

On UNIX, Linux, and Windows systems: This command also starts the queue manager that is associated with the component if it is not already active. You can also use this command to start the configuration manager (component = ConfigMgr).

On z/OS systems: You cannot use this command to start a queue manager or user name server. See "Command restrictions on z/OS systems" on page 270 for more information.

QI Stop Component command

This command stops a broker product component.

Identify the component that you want to stop by specifying its name in the **Value** column of the Edit Argument window. The destination of this command must be an agent-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiStop 'ComponentName'
```

On UNIX, Linux, and Windows systems: You can also use this command to stop the configuration manager (component = ConfigMgr).

On z/OS systems: you cannot use this command to stop a user name server.

QI Stop Component and its Queue Manager command

This command stops a broker product component and the queue manager that is associated with it.

Identify the component that you want to stop by specifying its name in the **Value** column of Edit Argument window. The destination of this command must be the agent-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiStop -q 'ComponentName'
```

On UNIX, Linux, and Windows systems: This command stops the broker product component and the queue manager that is associated with it. You can also use this command to stop the configuration manager (component=ConfigMgr).

On z/OS systems: This command does not stop a queue manager; see “Command restrictions on z/OS systems” on page 270. This command has the same effect as the **QI Stop Component** command; it stops only an broker product component.

QI Change Trace Component command

This command changes the trace characteristics of broker product components.

On UNIX, Linux, and Windows systems: You can also use this command to change trace characteristics for the configuration manager (component=ConfigMgr).

Identify the broker whose trace characteristics you want to change by specifying its name in the top field of the Edit Argument window. If the broker name contains spaces, the entire string must be enclosed in one set of double quotation marks (for example, "aaaaaa bbbb cc"). Specify the new trace parameters in the lower field of the Edit Argument window. The format of the argument must use the same format as used by the **mqsichangetrace** command. For change trace commands targeted to z/OS systems brokers, follow the brokers documentation precisely for the **changetrace** command parameters.

Remember: On z/OS systems, any lowercase label names must be in one set of double quotation marks, and commas must separate parameters. For example, on z/OS systems, supply this argument to turn on normal trace for the default execution group:

```
u=yes,e="default",l=normal
```

On other operating systems, the following argument has the same result:

```
-u -e default -l normal
```

The destination of this command must be an agent-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiChangeTrace 'ComponentName' 'Parms'
```

You can refer to user's guide of the WebSphere Message Broker for the command details.

Commands for broker-managed systems

The Take Action commands for broker managed systems are as follows.

WMB Start Broker/QI Start Broker command

This command starts a broker.

Specify the name of the broker that you want to start in the Edit Argument window. The destination of this command must be a broker-managed system.

Remember: On UNIX and Linux systems, this command can only be used to start a broker after the database used by the broker has been sourced.

The syntax of the command that is sent to the agent is:

```
QI:mqsiStart 'BrokerName'
```

On z/OS systems: The product does not currently support the **f broker, startcomponent** command form of starting the broker. The **s broker** command form is used.

WMB Stop Broker/QI Stop Broker command

This command stops a broker.

Specify the name of the broker that you want to stop in the Edit Argument window. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiStop 'BrokerName'
```

On z/OS systems: Using the **f broker, stopcomponent** command to stop the broker is not currently supported. The **p broker** command is used instead.

WMB Stop Broker/QI Stop Broker and its Queue Manager command

On UNIX, Linux, and Windows systems, this command stops the broker and its associated queue manager.

On z/OS systems, this command does not stop the associated queue manager (see "Command restrictions on z/OS systems" on page 270). This command has the same effect as the **WMB Stop Broker/QI Stop Broker** command; it only stops the broker.

Specify the name of the broker that you want to stop in the Edit Argument window. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiStop -q 'BrokerName'
```

WMB Change Trace Broker/QI Change Trace Broker command

This command changes the trace characteristics of an IBM broker product component.

Identify the broker whose trace characteristics you want to change by specifying its name in the top field of the Edit Argument window. If the broker name contains spaces, the entire string must be enclosed in one set of double quotation marks (for example, "aaaaaa bbbb cc"). Specify the new trace parameters in the lower field of the Edit Argument window. The format of the argument must use the same format as used by the **mqsichangetrace** command. For change trace commands targeted to z/OS systems brokers, follow the brokers documentation precisely for the **changetrace** command parameters.

Remember: On z/OS systems, any lowercase label names must be in one set of double quotation marks, and commas must separate parameters. For example, on z/OS systems, supply this argument to turn on normal trace for the default execution group:

```
u=yes,e="default",l=normal
```

On other operating systems, the following argument has the same result:

```
-u -e default -l normal
```

The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:mqsiChangeTrace 'BrokerName' 'Parms'
```

You can refer to WebSphere Message Broker documentation for the command details.

WMB Change Broker/QI Change Broker command

Use this command to change the archiving interval of a broker in the Tivoli Enterprise Portal. You can also modify other broker properties by using this command.

Identify the broker by specifying its name in the top field of the Edit Argument window. If the broker name contains spaces, the entire string must be enclosed within two sets of double quotation marks (for example, ""aaaaaa bbbb cc""). Specify the new trace parameters in the lower field of the Edit Argument window. The format of the argument must use the same format as used by IBM **mqsi changebroker** command, using two sets of double quotation marks for any double quotation marks used. For change broker commands targeted to z/OS systems brokers, follow the brokers documentation precisely for the **changebroker** command parameters, except use two sets of double quotation marks instead of single quotation marks. Remember that on z/OS systems any lowercase label names must be in quotation marks, and commas must separate parameters.

The syntax of the command that is sent to the agent is:

```
QI:mqsiChangeBroker 'BrokerName' 'Parms'
```

The *BrokerName* value is the name of the targeted broker. The *Parms* value provides the rest of the necessary parameters that are required or optional for the **mqsi changebroker** (**changebroker** on z/OS systems) command as documented by IBM. Tivoli Enterprise Portal does not parse or verify the parameters.

You can refer to WebSphere Message Broker documentation for the command details.

WMB Change Flow Stats/QI Change Flow Stats command

Use this command to activate or deactivate Message Flow Accounting and Statistics from the Tivoli Enterprise Portal.

Identify the broker by specifying its name in the top field of the Edit Argument window. If the broker name contains spaces, the entire string must be enclosed in two sets of quotation marks (for example, ""aaaaaa bbbb cc""). Specify the new trace parameters in the lower field of the Edit Argument window. The argument must use the same format as used by **mqsi changeflowstats** command except that you need to use two sets of quotation marks when one set of quotation marks is used in the **mqsi changeflowstats** command. For change trace commands targeted to the brokers that are running on z/OS systems, follow the brokers documentation precisely for the **changeflowstats** command, except that you need to use two sets of quotation marks instead of one set of quotation marks. Remember that on z/OS systems any lowercase label names must be enclosed in quotation marks, and commas must be used to separate parameters.

The syntax of the command that is sent to the agent is:

```
QI:mqsiChangeFlowStats 'BrokerName' 'Parms'
```

The *BrokerName* value specifies the name of the targeted broker. The *Parms* value specifies the rest of the parameters for the **mqsichangeflowstats** (**changeflowstats** on z/OS systems) command as documented by IBM. Tivoli Enterprise Portal does not parse or verify the parameters. If you want to view data in the accounting workspaces, you must specify `xml` as the output (`-o xml`).

You can refer to WebSphere Message Broker documentation for the command details.

WMB Change Properties/QI Change Properties command

Use this command to enable Publish-Subscribe Statistics, Multicast Summary Statistics, Multicast Group Statistics, and Multicast Topic Statistics workspaces. This command is available only for the broker subnode affinity.

The command has two parameters. The *BrokerName* value specifies the name of the broker that is being targeted. The *Parms* value specifies the rest of the parameters for the **mqsiChangeProperties** command. This command is not available on z/OS systems.

The syntax of the command that is sent to the agent is:

```
QI:mqsiChangeProperties 'BrokerName' 'Parms'
```

You can refer to WebSphere Message Broker documentation for the command details.

WMB Start Message Flow(s)/QI Start Message Flow(s) command

This command starts one or all of the message flows that are associated with an execution group. Enter the name of the execution group to which the message flow belongs in the top field of the Edit Argument window. Specify the message flow that you want to start in the lower field of the Edit Argument window. If you do not specify a value in the lower field, the command starts the specified execution group and all its associated message flows. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is as follows:

- WMB Start Message Flow(s)
QI:mflowStart 'ExecutionGroup' 'Application' 'Library' 'MessageFlow'
- QI Start Message Flow(s)
QI:mflowStart 'ExecutionGroup' 'MessageFlow'

WMB Stop Message Flow(s)/QI Stop Message Flow(s) command

This command stops one or all of the message flows that are associated with an execution group. Enter the name of execution group to which the message flow belongs in the top field of the Edit Argument window. Specify the message flow that you want to stop in the lower field of the Edit Argument window. If you do not specify a value in the lower field, the command stops all the message flows that are associated with the execution group. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is as follows:

- WMB Stop Message Flow(s)
QI:mflowStop 'ExecutionGroup' 'Application' 'Library' 'MessageFlow'

- QI Stop Message Flow(s)
`QI:mflowStop 'ExecutionGroup' 'MessageFlow'`

QI Delete Retained Publication command

This command is applicable only to the brokers earlier than version 7.0. This command deletes the retained publication for a specified topic and subscription point. Enter the topic of the publication that you want to delete in the top field of the Edit Argument window. Enter the subscription point of the publication that you want to delete in the lower field of the Edit Argument window. If you do not specify a subscription point, the publication for the default subscription point is deleted. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:retainedPubDelete 'Topic' 'SubscriptionPoint'
```

WMB Create User Statistics/QI Create User Statistics command

Use this command to create user statistics for the Tivoli Enterprise Portal logon ID that issues the command.

The logon ID that is used to create the user statistics must also be used to view them. In the Edit Argument window, specify the number of minutes that the user statistics created by this command are retained by the agent without being sampled or queried for display. If you do not specify a value, the data is stored until the **WMB Create User Statistics/QI Destroy User Statistics** command is issued. The target of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:createUserStatistics 'MaxInactivityTime'
```

WMB Sample User Statistics/QI Sample User Statistics command

This command causes the agent to take a sample of CandleMonitor node statistics for the Tivoli Enterprise Portal logon ID that issues the command. A **WMB Create User Statistics/QI Create User Statistics** command must have been issued for the logon ID before for this command to be valid. You can use this command to set an interval during which statistics are gathered and sent to the statistics workspaces. Issue this command for the first time when you want the interval to begin; issue the command for a second time when you want the interval to end. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:sampleUserStatistics
```

WMB Destroy User Statistics/QI Destroy User Statistics command

This command cleans up all user statistics that is associated with the Tivoli Enterprise Portal logon ID that issues the command and ensures that the agent does not store user statistics any more. The storage space that is used by the statistics must be released when it is no longer required. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:destroyUserStatistics
```

WMB Reset Statistics/QI Reset Statistics command

Use this command to reset all the statistics that are collected by the CandleMonitor node to zeros. This is useful when you are using summary statistics to perform

controlled timings. It resets all statistics for all nodes in the broker. It also updates the shared memory area that the CandleMonitor node uses to communicate the data to the agent, so that all uses of these statistics (regular or user) are affected. Without this command, the statistics are reset automatically when the broker is deployed or when overflow is detected. You cannot reset statistics for only part of a broker because inconsistent data might be generated in Broker Statistics and historical data reports.

Issue this command during a period of broker inactivity; otherwise, unpredictable results might be produced, and it might seem that some statistics have not been reset because of current activity. The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:resetStatistics
```

WMB Refresh Broker Data/QI Refresh Broker Data command

Use this command to make the agent collect all broker configuration data from the broker. This data includes information displayed in workspaces with the suffix information, such as Message Flow Information, and the topology workspaces, such as Execution Group Topology.

In most situations it is not necessary to use this command to refresh the information displayed in information and topology workspaces. This information is collected automatically in the following situations:

- The data has never been collected before. This typically occurs the first time the agent is run.
- The persistent file that contains the collected data is missing or corrupted when the agent starts up.
- A broker event indicates that a deploy action has occurred. In this case only data related to the deploy operation is collected.

However, if the agent fails to receive broker events or the reply queue of the agent is deleted while the agent is stopped, the agent cannot receive broker events that cause it to collect data about newly deployed objects. In this case you must run this take action command manually.

The destination of this command must be a broker-managed system.

The syntax of the command that is sent to the agent is:

```
QI:refreshBrokerData
```

Command restrictions on z/OS systems

- The Take Action commands for the WebSphere Message Broker Monitoring agent require the broker name to be the same as the procedure name of the task that is started or stopped.
- If the syntax of the Take Action command contains the **Parms** parameter, and the parameter value contains one or more lower case letters, enclose the value in quotation marks.
- The WebSphere Message Broker Monitoring agent cannot start or stop a queue manager on z/OS systems.
- If you want to issue a command that is not supported by the agent, you can issue it as a system command (a z/OS systems console command) instead. Use the exact syntax of the command as documented by IBM.

- The exact command that is issued by the agent varies depending on the host operating system on which the agent runs (especially on z/OS systems). Also, although the commands can be issued by the agent on all supported operating systems, some differences exist in the supported actions (especially on z/OS systems). Differences in actions, if they exist, are noted for each command in the following sections.

Workspaces

The WebSphere Message Broker Monitoring agent provides two different sets of workspaces for the brokers of version 7.0 or later and brokers that are earlier than version 7.0. In the Navigator physical view, those workspaces are grouped into two different folders according to the broker version. The workspaces for brokers of version 7.0 or later are listed in the **WebSphere Message Broker** folder. And the workspaces for brokers that are earlier than version 7.0 are listed in the **QI Broker** folder.

To access the broker-level workspaces, expand the **WebSphere Message Broker** or the **QI Broker** folder, and the monitored broker managed systems are listed by the broker names. When only one broker is monitored, the broker managed system folder is merged with the **WebSphere Message Broker** or the **QI Broker** folder depending on the broker version.

The general introduction of each workspace included in the WebSphere Message Broker Monitoring agent is described in this section.

For WebSphere Message Broker V7.0 or later

If the broker to be monitored is WebSphere Message Broker V7.0 or later, the monitoring agent can provide the predefined workspaces that are listed in this section.

Archive Accounting Message Flow Statistics workspace

This workspace provides statistics pertaining to entire message flows, for the current archive collection interval. In this top-level view, there is one row per monitored message flow, which contains data for the message flow that was collected during the current collection interval. The interval at which data is collected is configured at the broker. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Message Flow Statistics attributes” on page 137.

Default views

This workspace contains the following default views:

- **Archive Accounting Message Flow Statistics:** Presents a table view that contains accounting statistics arranged by message flow.
- **Message Flow CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time, in microseconds, that was spent processing messages during the current archive interval.

- **Message Flow Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time, in microseconds, that was spent processing messages during the current archive interval.
- **Message Flow Start Rate:** Presents the total input message rate (flow instance start count per second).
- **Message Flow Throughput:** Presents the total input messages (flow instance start count), total committed messages (flow complete count) and total number of backouts performed (flow exception count).

Predefined links

From Archive Accounting Message Flow Statistics workspace, you can link to the following workspaces by right-clicking a row in the **Archive Accounting Message Flow Statistics** table:

- Archive Accounting Node Statistics
- Archive Accounting Terminal Statistics
- Archive Accounting Thread Statistics
- Recent Archive Accounting Message Flow Statistics (Record Type: Archive, Query Type: Recent)
- Historical Archive Accounting Message Flow Statistics (Record Type: Archive, with history timespan specified)

Archive Accounting Node Statistics workspace

The Archive Accounting Node Statistics workspace contains statistical information about the nodes used by message flows during the current archive interval. There is one row per node in a message flow. Typically, you link to this workspace for information about a specific message flow, so all nodes displayed pertain to that message flow. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Node Statistics attributes” on page 140.

Default views

This workspace contains the following default views:

- **Archive Accounting Node Statistics:** Presents a table that contain archive statistics for nodes used by message flows.
- **Node CPU Time:** Presents a bar chart that contains, the minimum, maximum and average CPU time that spent processing an input message
- **Node Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that the node spent processing an input message.
- **Node Throughput:** Presents the node invocation count.
- **Node Throughput Rate:** Presents the node invocation count per second.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Archive Accounting Node Statistics** table view:

- Archive Accounting Terminal Statistics
- Recent Archive Accounting Node Statistics (Record Type: Archive, Query Type: Recent)
- Historical Archive Accounting Node Statistics (Record Type: Archive, with history timespan specified)

The number of rows of data that are displayed in the Recent Archive Accounting Node Statistics workspace depends on the value of the **defaultRetainRecentArchiveSamples** (or **retainRecentArchiveSamples**) parameter, which determines the minimum number of recent archive records to retain. However, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the table view of Archive Accounting Message Flow Statistics workspace, right-click a row and click **Workspace > Archive Accounting Node Statistics**.
- In the Navigator physical view, right-click the **Accounting Message Flow Statistics** node and click **Archive Accounting Node Statistics**.

Archive Accounting Terminal Statistics workspace

The Archive Accounting Terminal Statistics workspace contains statistical information about terminals that are on nodes used by message flows during the current archive interval. There is one row per terminal for each node in a message flow. Typically, you link to this workspace for a specific message flow and node; all terminals for which data is displayed belong to that message flow and node. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Terminal Statistics attributes” on page 141.

Default views

This workspace contains the following default views:

- **Archive Accounting Terminal Statistics:** Presents a table that contains archive statistics for terminals that are on nodes used by message flows, for the current collection interval.
- **Terminal Invocation Rate:** Presents a bar chart that contains the rate at which terminals were invoked during the current interval.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Archive Accounting Terminal Statistics** table view:

- Recent Archive Accounting Terminal Statistics (Record Type: Archive, Query Type: Recent)
- Historical Archive Accounting Terminal Statistics (Record Type: Archive, with history timespan specified)

The number of rows of data that are displayed in the Recent Archive Accounting Terminal Statistics workspace depends on the value of the **defaultRetainRecentArchiveSamples** (or **retainRecentArchiveSamples**) parameter, which indicates the minimum number of recent archive records to retain. However, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the table view of the Archive Accounting Node Statistics workspace, right-click a row and click **Workspace > Archive Accounting Terminal Statistics**.
- In the Navigator physical view, right-click the **Accounting Message Flow Statistics** node and click **Archive Accounting Terminal Statistics**.

Archive Accounting Thread Statistics workspace

This workspace contains archive statistics that pertain to the threads that are used by message flows. There is one row of data per thread for each message flow for the current archive interval. Typically, you link to this workspace for a specific message flow, so all threads that are displayed relate to that message flow. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Thread Statistics attributes” on page 143.

Default views

This workspace contains the following default views:

- **Archive Accounting Thread Statistics:** Presents a table that contains archive statistics for threads used by message flows.
- **Thread Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that was spent processing input messages during the current interval.
- **Thread CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time that was spent processing input messages during the current interval.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Archive Accounting Thread Statistics** table:

- Recent Archive Accounting Thread Statistics (Record Type: Archive, Query Type: Recent)
- Historical Archive Accounting Thread Statistics (Record Type: Archive, with history timespan specified)

The number of rows of data displayed in the Recent Archive Accounting Thread Statistics workspace depends on the value of the

defaultRetainRecentArchiveSamples (or **retainRecentArchiveSamples**) parameter, which indicates the minimum number of recent archive records. However, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the table view of the Archive Accounting Message Flow Statistics workspace, right-click a row and click **Workspace > Archive Accounting Thread Statistics**.
- In the Navigator physical view, right-click the **Accounting Message Flow Statistics** node and click **Archive Accounting Thread Statistics**.

Broker Status workspace

The Broker Status workspace contains information about a broker-managed system, such as the status of the broker, its process ID, information about the operating system where the broker runs, its job on z/OS systems, its associated queue manager. It also contains a graphical view of your broker environment, including execution groups and message flows.

For related attribute groups, see “Broker Status attributes” on page 145, “Execution Group Status attributes” on page 151, and “Message Flow Status attributes” on page 154.

Default views

This workspace contains the following default views:

- **Broker Status:** Presents a table that contains broker status and definition information.
- **Execution Group Status:** Presents a table that contains execution group status information.
- **Message Flow Status:** Presents a table that contains message flow status information.
- **Broker Topology:** Contains a graphical view of your broker environment, including applications, libraries, execution groups, and messages flows.

Predefined links

You can choose to link to the following topology views or workspaces by right-clicking a row in a table view in this workspace:

- From the Broker Status table:
 - Historical Broker Status workspace
- From the Execution Group Status table:
 - Execution Group topology in the Broker Status workspace
 - Execution Group topology in the Historical Broker Status workspace
- From the Message Flow Status table:
 - Message Flow topology in the Broker Status workspace
 - Message Flow topology in the Historical Broker Status workspace
 - Message Flow Node Topology workspace

You can also link to the Message Flow Node Topology workspace by right-clicking a message flow node in the topology view.

Remember: To get data for tables in the Historical Broker Status workspace, you must enable historical data collection for the Broker Status, Execution Group Status, and Message Flow Status attribute groups. Otherwise, the table might be empty in the historical workspace.

Broker Status Events workspace

The Broker Status Events workspace lists the events that are generated on a particular broker managed system as they occur. The number of events that is displayed depends on the number that are retained by the agent; that value is set in the agent parameter file.

The agent receives broker events data with a subscription to broker event publications. If a particular event is not published by a broker, it is not available for display in this workspace.

Tip: The Broker Status attribute in the Broker Status attribute group is the best way to detect whether the broker is started or stopped, because the Broker Event indicating that a broker has stopped might not always occur.

For related attribute group, see “Broker Status Events attributes” on page 147.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised within the whole system.
- **Broker Status Events:** Presents a table that contains the broker events that were retained most recently.

Components workspace

The Components workspace contains a list of IBM broker product components and their state at the agent level.

Remember: On distributed systems, all of the IBM broker product components are listed; on z/OS systems, only the brokers that are specified in the parameter file using the MonitorBroker tag are listed in this workspace.

For related attribute group, see “Components attributes” on page 149.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents, as a table, all situations that were raised in the entire system.
- **Components:** Presents, as a table, all installed broker components on distributed systems, and all monitored broker components on z/OS the system.

Predefined links

By default, you can link from this Components table view to the Components History workspace (Components short-term history collection is inactive by default.)

In addition to the previous workspace, the following workspaces are available in specific circumstances:

- If you have the WebSphere MQ Monitoring agent installed on the same system you can also link to Queue Manager Status workspace of the WebSphere MQ Monitoring agent using the queue manager attribute. For more information about the Queue Manager Status workspace, see the online help for the WebSphere MQ Monitoring agent.
- If you have the OS agent installed on the same system you can also link to System Overview workspace of OS agent using the system name given in the workspaces provided by the WebSphere Message Broker Monitoring agent.

File Resource Statistics workspace

This workspace contains the statistical information for the local file system of any file actions done by any type of file node.

For related attribute group, see “File Resource Statistics attributes” on page 152.

Default views

This workspace contains the following default views:

- **File Resource Statistics:** Presents a table that contains the statistical information about file resources. There is one record for each file resource.
- **Files Read and Created:** Presents a bar chart that displays how many files are read and how many files are created within an execution group.
- **Bytes Read and Written:** Presents a bar chart that displays how many bytes are read and how many bytes are written within an execution group.

Predefined links

You can link to the following workspaces by right-clicking a row in the **File Resource Statistics** table:

- Historical File Resource Statistics

Accessing this workspace

You can link to this workspace by right-clicking the **Resource Statistics** node in the Navigator physical view and then clicking **File Resource Statistics**.

JDBC Connection Pools Resource Statistics workspace

This workspace contains statistical information for each JDBC Provider configurable service.

For related attribute group, see “JDBC Connection Pools Resource Statistics attributes” on page 153.

Default views

This workspace contains the following default views:

- **JDBC Connection Pools Resource Statistics:** Presents a table that contains statistical information for each JDBC Provider configurable service.
- **Total and Delayed Requests:** Presents a bar chart that displays how many requests have been received by the connection pool and how many requests have been delayed within each execution group during the accounting period.
- **Total and Timed Out Requests:** Presents a bar chart that displays how many requests have been received by the connection pool and how many requests cannot be satisfied in 15 seconds within each execution group.

Predefined links

You can link to the following workspaces by right-clicking a row in the **JDBC Connection Pools Resource Statistics** table:

- Historical JDBC Connection Pools Resource Statistics

Accessing this workspace

You can link to this workspace by right-clicking the **Resource Statistics** node in the Navigator physical view and then clicking **JDBC Connection Pools Resource Statistics**.

JVM Resource Statistics workspace

Each execution group starts its own Java virtual machine (JVM). This workspace contains the statistical information about the JVM resources used by execution groups. Statistics are collected for the following JVM resources:

- Heap memory
- Non-heap memory
- Garbage collection

For related attribute group, see “JVM Resource Statistics attributes” on page 154.

Default views

This workspace contains the following default views:

- **JVM Resource Statistics**: Presents a table that contains the statistical information about JVM resources. There is one record for each resource.
- **JVM Performance Summary**: Presents a bar chart that displays how much memory is allocated to the JVM resource and how much memory is in use by the JVM resource.
- **Garbage Collection**: Presents a bar chart that displays how many accumulated garbage collections have occurred within each execution group.

Predefined links

You can link to the following workspaces by right-clicking a row in the **JVM Resource Statistics** table:

- Historical JVM Resource Statistics

Accessing this workspace

You can link to this workspace by right-clicking the **Resource Statistics** node in the Navigator physical view and then clicking **JVM Resource Statistics**.

Message Flow Node Topology workspace

The Message Flow Node Topology workspace contains a graphical view of a message flow (including its nodes and connection between nodes) and a table view of detailed information about the message processing nodes.

For related attribute group, see “Message Processing Nodes attributes” on page 158.

Default views

This workspace contains the following default views:

- **Message Flow Node Topology:** Contains a graphical view of the message flow, including its nodes and connections between nodes.
- **Message Processing Nodes:** Contains detailed information about the given message flow processing node.

Predefined links

By default, you can link from processing nodes in either the Message Processing Nodes table view or the Message Flow Topology view to the Processing Node Attributes workspace to view information about the selected processing node.

Accessing this workspace

You can access this workspace in one of the following ways:

- In the Message Flow Status workspace, right-click a row in the table view and click **Workspace > Message Flow Node Topology**.
- In the Broker Status workspace, right-click a message flow icon in the Broker Topology view and click **Link To > Message Flow Node Topology**.

Monitor Node Base Statistics workspace

The Monitor Node Base Statistics workspace contains message flow statistics for a particular broker-managed system, organized by the CandleMonitor nodes that collect the statistics. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input, output, or other.

For related attribute group, see “Monitor Node Base Statistics attributes” on page 161.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains the message flow rate, organized by message processing node.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times, organized by message processing node.
- **Monitor Node Base Statistics:** Presents a table that contains statistical data, organized by CandleMonitor node.

Predefined links

When historical data collection is enabled for the Monitor Node Base Statistics attribute group, you can link to the Historical Monitor Node Base Statistics workspace by right-clicking a row in the table.

Accessing this workspace

Use one of the following methods to access this workspace:

- In the Navigator physical view, right-click **Monitor Node Broker Statistics** and click **Monitor Node Base Statistics**.

- In the **Monitor Node Message Flow Statistics** workspace, right-click a row in the table and click **Monitor Node Base Statistics**.

Monitor Node Broker Statistics workspace

The Monitor Node Broker Statistics workspace provides high-level message flow statistics that are summarized at the broker level. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input or output.

For related attribute group, see “Monitor Node Broker Statistics attributes” on page 164.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains the message input rate and message output rate for the broker.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times for the broker.
- **Monitor Node Broker Statistics:** Presents a table that contains statistical data summarized for the broker.

Remember: If the broker or the WebSphere Message Broker Monitoring agent stops and restarts during the collection interval, values of the attributes beginning with Current such as Current Interval and Current Msg Input Count, might be wrong in the first interval. After the broker or the monitoring agent starts again, correct statistical data can be displayed from next interval.

Predefined links

By default, you can link to the following workspaces by right-clicking a row in the table view of this workspace:

- Monitor Node Execution Group Statistics
- Historical Monitor Node Broker Statistics

Accessing this workspace

To access this workspace, in the Navigator physical view, click **Monitor Node Broker Statistics**.

Monitor Node Events workspace

The Monitor Node Events workspace lists events that are generated by CandleMonitor nodes on a particular broker-managed system as they occur. The number of events that is displayed depends on the number of events that is retained by the agent; the value is set in the agent parameter file.

For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to other and message text specified in the **eventMessage** attribute.

For related attribute group, see “Monitor Node Events attributes” on page 168.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.
- **Monitor Node Events:** Presents a table that contains the monitor node events that were retained most recently.

Predefined links

By default, you can link to the Historical Monitor Node Events workspace by right-clicking a row in the **Monitor Node Events** table.

Accessing this workspace

To link to this workspace, in the Navigator physical view, right-click **Monitor Node Broker Statistics** and click **Monitor Node Events**.

Monitor Node Execution Group Statistics workspace

The Monitor Node Execution Group Statistics workspace contains high-level message flow statistics summarized at the execution group level for a particular broker-managed system. To obtain data for this workspace, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input or output.

For related attribute group, see “Monitor Node Execution Group Statistics attributes” on page 170.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains message input rate and message output rate, organized by execution group.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times, organized by execution group.
- **Monitor Node Execution Group Statistics:** Presents a table that contains state and statistical data, organized by execution group.

Remember: If the broker or the WebSphere Message Broker Monitoring agent stops and restarts during the collection interval, values of the attributes beginning with Current such as Current Interval and Current Msg Input Count, might be wrong in the first interval. After the broker or the monitoring agent starts again, correct statistical data can be displayed from next interval.

Predefined links

By default, you can link to the following workspaces by right-clicking a row in the **Monitor Node Execution Group Statistics** table:

- Monitor Node Message Flow Statistics
- Historical Monitor Node Execution Group Statistics

Accessing this workspace

To link to this workspace, in the Navigator physical view, right-click **Monitor Node Broker Statistics** and click **Monitor Node Execution Group Statistics**.

Monitor Node Message Flow Statistics workspace

The Monitor Node Message Flow Statistics workspace contains high-level message flow statistics summarized at the message flow level for a particular broker-managed system. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input or output.

For related attribute group, see “Monitor Node Message Flow Statistics attributes” on page 174.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains message input and output rate, organized by message flow.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times, organized by message flow.
- **Monitor Node Message Flow Statistics:** Presents a table that contains state and statistical data, organized by message flow.

Remember: If the broker or the WebSphere Message Broker Monitoring agent stops and restarts during the collection interval, values of the attributes beginning with Current such as Current Interval and Current Msg Input Count, might be wrong in the first interval. After the broker or the monitoring agent starts again, correct statistical data can be displayed from next interval.

Predefined links

By default, you can link to the following workspaces by right-clicking a row in the table view of this workspace:

- Monitor Node Sub-Flow Statistics
- Monitor Node Base Statistics
- Historical Monitor Node Message Flow Statistics

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the Navigator physical view, right-click **Monitor Node Broker Statistics** and click **Monitor Node Message Flow Statistics**.
- In the **Monitor Node Execution Group Statistics** workspace, by right-clicking a row in the table view and click **Monitor Node Message Flow Statistics**.

Monitor Node Sub-Flow Statistics workspace

The Monitor Node Sub-Flow Statistics workspace provides message flow statistics summarized at the subflow level.

If you access the Monitor Node Sub-Flow Statistics workspace from the Navigator view, all subflows in the specified broker are displayed organized by execution groups and message flows. If you link to this workspace from the Monitor Node Message Flow Statistics workspace for a given message flow, only the subflows that are associated with that message flow are displayed. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to subFlowOutput or subFlowInput.

For related attribute group, see “Monitor Node Sub-Flow Statistics attributes” on page 178.

Default views

This workspace contains the following three views:

- **Current Message Rates:** Presents a bar chart that contains the message input and output rates, organized by subflow.
- **Current Average Message Time:** Presents a bar chart that contains the average queue and flow times, organized by subflow.
- **Monitor Node Sub-Flow Statistics:** Presents a table that contains state and statistical data, organized by subflow.

Remember: If the broker or the WebSphere Message Broker Monitoring agent stops and restarts during the collection interval, values of the attributes beginning with Current such as Current Interval and Current Msg Input Count, might be wrong in the first interval. After the broker or the monitoring agent starts again, correct statistical data can be displayed from next interval.

Predefined links

When historical data collection is enabled for the Monitor Node Sub-Flow Statistics attribute group, you can link to the Historical Monitor Node Sub-Flow Statistics workspace by right-clicking a row in the table view of this workspace.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the Navigator physical view, right-click **Monitor Node Broker Statistics** and click **Monitor Node Sub-Flow Statistics**.
- In the **Monitor Node Message Flow Statistics** workspace, by right-clicking a row in the table view and click **Monitor Node Sub-Flow Statistics**.

ODBC Resource Statistics workspace

This workspace contains the statistical information that is collected for ODBC DSN resources. Statistics are reported for each ODBC DSN that has been accessed since the execution group starts.

For related attribute group, see “ODBC Resource Statistics attributes” on page 182.

Default views

This workspace contains the following default views:

- **ODBC Resource Statistics:** Presents a table that contains the statistical information about ODBC DSN resources. There is one record for each ODBC DSN resource.
- **ODBC Execute Results:** Presents a bar chart that displays how many statements have been executed and how many statements have failed within in each execution group.

Predefined links

You can link to the following workspaces by right-clicking a row in the **ODBC Resource Statistics** table:

- Historical ODBC Resource Statistics

Accessing this workspace

You can link to this workspace by right-clicking the **Resource Statistics** node in the Navigator physical view and then clicking **ODBC Resource Statistics**.

Parsers Resource Statistics workspace

All message flows in an execution group create parsers to parse and write input and output messages. This workspace contains statistical information about the parser resources within each execution group. The statistics are summarized by parser names used within the execution group.

For related attribute group, see “Parsers Resource Statistics attributes” on page 183.

Default views

This workspace contains the following default views:

- **Parsers Resource Statistics:** Presents a table that contains the statistical information about parser resources. There is one record for each parser resource.
- **Reads and Failed Reads:** Presents a bar chart that displays how many parsers and failed parsers have occurred for each execution group.
- **Writes and Failed Writes:** Presents a bar chart that displays how many writes and failed writes have occurred for each execution group.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Parsers Resource Statistics** table:

- Historical Parsers Resource Statistics

Accessing this workspace

You can link to this workspace by right-clicking the **Resource Statistics** node in the Navigator physical view and then clicking **Parsers Resource Statistics**.

Processing Node Attributes workspace

The Processing Node Attributes workspace lists all attribute values for the selected message processing node (within a message flow).

Message processing node attributes are defined in the broker product Message Brokers Toolkit when you configure a node within a message flow. Each node has its own distinct attributes that you can use to customize node functionality.

This workspace only contains data if node data collection is enabled in the `kqi.xml` configuration file. Note, however, that if message brokers in your environment are configured with a large number of message flows, or message flows that have a lot of message processing node data, enabling this option might drastically degrade system performance.

For more information about enabling node data collection for all message brokers, see “defaultCollectNodeData” on page 24, and for a single message broker, see “collectNodeData” on page 31.

For related attribute group, see “Processing Node Attributes” on page 184.

Default views

This workspace contains the following default views:

- **Message Log:** Presents a table that gives you an overview of changes in the status of your monitored system.
- **Processing Node Attributes:** Presents a table that contains a list of each attribute and its value for the selected processing node.

Accessing this workspace

You link to this workspace from the Message Flow Node Topology workspace in one of the following ways:

- In the Message Flow Topology view, right-click the message flow node and click **Workspace > Processing Node Attributes**.
- In the Message Processing Nodes table view, right-click a row and click **Link To > Processing Node Attributes**.

Product Events workspace

The Product Events workspace contains a list of product events that are generated by a particular agent when a problem occurs that affects the agent ability to collect data from the broker. The number of events displayed depends on the number of events that are retained by the agent, as set by the **retainProductEvents** parameter in the agent configuration file.

For an explanation of the agent-generated product events and how they can be used in problem determination, see *Troubleshooting Guide*.

For related attribute group, see “Product Events attributes” on page 185.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Product Events:** Presents a table that contains the product events that were retained most recently.

Predefined links

By default, you can link from this Product Events table view to the Product Events History. (Product Events short-term history collection is disabled by default.)

Snapshot Accounting Message Flow Statistics workspace

This workspace provides statistics pertaining to entire message flows, for the current snapshot collection interval. In this top-level view, there is one row per monitored message flow, which contains data for the message flow that was collected during the current collection interval. The interval at which data is collected is configured at the broker. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Message Flow Statistics attributes” on page 137.

Default views

This workspace contains the following default views:

- **Message Flow CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time, in microseconds, that was spent processing messages during the current snapshot interval.
- **Message Flow Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time, in microseconds, that was spent processing messages during the current snapshot interval.
- **Message Flow Throughput:** Presents the total input messages (flow instance start count), total committed messages (flow complete count) and total number of backouts performed (flow exception count).
- **Message Flow Start Rate:** Presents the total input message rate (flow instance start count per second).
- **Snapshot Accounting Message Flow Statistics:** Presents a table view that contains accounting statistics arranged by message flow.

Predefined links

From Snapshot Accounting Message Flow Statistics workspace, you can link to the following workspaces by right-clicking a row in the **Snapshot Accounting Message Flow Statistics** table:

- Snapshot Accounting Thread Statistics
- Snapshot Accounting Node Statistics
- Snapshot Accounting Terminal Statistics
- Recent Snapshot Accounting Message Flow Statistics (Record Type: Snapshot, Query Type: Recent)

Snapshot Accounting Node Statistics workspace

The Snapshot Accounting Node Statistics workspace contains statistical information about the nodes used by message flows during the current snapshot interval. There is one row per node in a message flow. Typically, you link to this workspace for information about a specific message flow, so all nodes displayed pertain to that message flow. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Node Statistics attributes” on page 140.

Default views

This workspace contains the following default views:

- **Node CPU Time:** Presents a bar chart that contains, the minimum, maximum and average CPU time that spent processing an input message
- **Node Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that the node spent processing an input message.

- **Node Throughput:** Presents the node invocation count.
- **Node Throughput Rate:** Presents the node invocation count per second.
- **Snapshot Accounting Node Statistics:** Presents a table that contain snapshot statistics for nodes used by message flows.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Snapshot Accounting Node Statistics** table view:

- Snapshot Accounting Terminal Statistics
- Recent Snapshot Accounting Node Statistics (Record Type: Snapshot, Query Type: Recent)

The number of rows of data that are displayed in the Recent Snapshot Accounting Node Statistics workspace depends on the value of the **defaultRetainRecentSnapshotSamples** (or **retainRecentSnapshotSamples**) parameter, which determines the minimum number of recent snapshot records to retain. However, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the table view of Snapshot Accounting Message Flow Statistics workspace, right-click a row and click **Workspace > Snapshot Accounting Node Statistics**.
- In the Navigator physical view, right-click the **Accounting Message Flow Statistics** node and click **Snapshot Accounting Node Statistics**.

Snapshot Accounting Terminal Statistics workspace

The Snapshot Accounting Terminal Statistics workspace contains statistical information about terminals that are on nodes used by message flows during the current snapshot interval. There is one row per terminal for each node in a message flow. Typically, you link to this workspace for a specific message flow and node; all terminals for which data is displayed belong to that message flow and node. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Terminal Statistics attributes” on page 141.

Default views

This workspace contains the following default views:

- **Snapshot Accounting Terminal Statistics:** Presents a table that contains snapshot statistics for terminals that are on nodes used by message flows, for the current collection interval.
- **Terminal Invocation Rate:** Presents a bar chart that contains the rate at which terminals were invoked during the current interval.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Snapshot Accounting Terminal Statistics** table view:

- Recent Snapshot Accounting Terminal Statistics (Record Type: Snapshot, Query Type: Recent)

The number of rows of data that are displayed in the Recent Snapshot Accounting Terminal Statistics workspace depends on the value of the **defaultRetainRecentSnapshotSamples** (or **retainRecentSnapshotSamples**) parameter, which indicates the minimum number of recent snapshot records to retain. However, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the table view of the Snapshot Accounting Node Statistics workspace, right-click a row and click **Workspace > Snapshot Accounting Terminal Statistics**.
- In the Navigator physical view, right-click the **Accounting Message Flow Statistics** node and click **Snapshot Accounting Terminal Statistics**.

Snapshot Accounting Thread Statistics workspace

This workspace contains snapshot statistics that pertain to the threads that are used by message flows. There is one row of data per thread for each message flow for the current snapshot interval. Typically, you link to this workspace for a specific message flow, so all threads that are displayed relate to that message flow. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the WMB Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Accounting Thread Statistics attributes” on page 143.

Default views

This workspace contains the following default views:

- **Snapshot Accounting Thread Statistics:** Presents a table that contains snapshot statistics for threads used by message flows.
- **Thread Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that was spent processing input messages during the current interval.
- **Thread CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time that was spent processing input messages during the current interval.

Predefined links

You can link to the following workspaces by right-clicking a row in the **Snapshot Accounting Thread Statistics** table:

- Recent Snapshot Accounting Thread Statistics (Record Type: Snapshot, Query Type: Recent)

The number of rows of data displayed in the Recent Snapshot Accounting Thread Statistics workspace depends on the value of the **defaultRetainRecentSnapshotSamples** (or **retainRecentSnapshotSamples**) parameter, which indicates the minimum number of recent snapshot records. However, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Accessing this workspace

You can link to this workspace in one of the following ways:

- In the table view of the Snapshot Accounting Message Flow Statistics workspace, right-click a row and click **Workspace > Snapshot Accounting Thread Statistics**.
- In the Navigator physical view, right-click the **Accounting Message Flow Statistics** node and click **Snapshot Accounting Thread Statistics**.

SOAP Input Resource Statistics workspace

The SOAPInput and SOAPReply nodes send and receive SOAP messages. This workspace contains statistical information about SOAP nodes on a per-operation basis. Statistics for the SOAP nodes are collected with both HTTP and JMS transport. You can review the name of the applied policy set if one is defined.

For related attribute group, see “SOAP Input Resource Statistics attributes” on page 191.

Default views

This workspace contains the following default views:

- **SOAP Input Resource Statistics:** Presents a table that contains statistical information about how many inbound messages the SOAPInput node receives, how many replies the SOAPReply node sends, and how many of those calls are successful or result in SOAP Faults.
- **Successful and Faulted Inbound Messages:** Presents a bar chart that displays how many messages have made the flow without faulting and how many messages have faulted before reaching the flow.
- **Successful and Faulted Replies:** Presents a bar chart that displays how many successful replies and SOAP Fault replies have been sent to the client within each execution group.

Predefined links

You can link to the following workspaces by right-clicking a row in the **SOAP Input Resource Statistics** table:

- Historical SOAP Input Resource Statistics

Accessing this workspace

You can link to this workspace by right-clicking the **Resource Statistics** node in the Navigator physical view and then clicking **SOAP Input Resource Statistics**.

For WebSphere Message Broker V6 (all releases)

If the broker to be monitored is WebSphere Message Broker V6 (all releases), the monitoring agent can provide the predefined workspaces that are listed in this section.

ACL Entries workspace

The ACL Entries workspace contains a list of Access Control List entries on a particular managed system, organized by topic. Access Control List entries define access to publish and subscribe topics.

For related attribute group, see “ACL Entries attributes” on page 192.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system, all situations that were raised in the entire system.
- **ACL Entries:** Presents, as a table, all ACL Entries for the broker.

Agent Status Workspace (agent-level summary)

The Agent Status workspace summarizes event and broker product component information from a particular agent managed system at the agent level.

For related attribute groups, see “Components attributes” on page 149 and “Product Events attributes” on page 185.

Default views

This workspace contains the following default views:

- **Product Events:** Presents a table that contains the product events that were most recently retained.
- **Components:** Presents, as a table, all components that have been created on the system that the agent is monitoring.

Accessing this workspace

To access this workspace, in the Navigator physical view click the **QI Agent** folder icon.

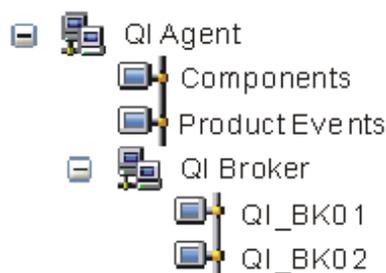


Figure 19. QI Agent in the Navigator physical view

Archive Message Flow Accounting workspace

This workspace provides statistics pertaining to entire message flows, for the current archive collection interval. In this top-level view, there is one row per

monitored message flow, which contains data for the message flow that was collected during the current collection interval. The interval at which data is collected is configured at the broker. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Message Flow Accounting attributes” on page 213.

Default views

This workspace contains the following default views:

- **Message Flow Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time, in microseconds, that was spent processing messages during the current archive interval.
- **Message Flow CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time, in microseconds, that was spent processing messages during the current archive interval.
- **Message Flow Throughput:** Presents the total input messages (flow instance start count), total committed messages (flow complete count) and total number of backouts performed (flow exception count).
- **Message Flow Start Rate:** Presents the total input message rate (flow instance start count per second).
- **Archive Message Flow Accounting:** Presents a table view that contains accounting statistics arranged by message flow.

Predefined links

This workspace contains links to the Archive Thread Accounting and Archive Node Accounting workspaces. This workspace also contains links to Recent Archive Message Flow Accounting and Historical Archive Message Flow Accounting workspaces for the selected message flow. The number of rows of data that are displayed in the Recent Archive Message Flow Accounting workspace depends on the value assigned to the **defaultRetainRecentArchiveSamples** (or **retainRecentArchiveSamples**) parameter, which specifies the minimum number of recent archive records to retain; however, the number of records that are retained might be greater than specified by these parameters, depending on the data retention requirements of historical data collection.

Archive Node Accounting workspace

This workspace contains statistical information about nodes used by message flows during the current archive interval. There is one row per node in a message flow. Typically, you link to this workspace for information about a specific message flow, so all nodes displayed pertain to that message flow. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Node Accounting attributes” on page 236.

Default views

This workspace contains the following default views:

- **Archive Node Accounting:** Presents a table that contain archive statistics for nodes used by message flows.
- **Node Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that the node spent processing an input message.
- **Node Throughput:** Presents the node invocation count.
- **Node Throughput Rate:** Presents the node invocation count per second.
- **Node CPU Time:** Presents a bar chart that contains, the minimum, maximum and average CPU time that spent processing an input message

Predefined links

This workspace contains a link to the Archive Terminal Accounting workspace. This workspace also contains links to Recent Archive Node Accounting and Historical Archive Node Accounting workspaces for the selected message flow. The number of rows of data that are displayed in the Recent Archive Node Accounting workspace depends on the value assigned to the **defaultRetainRecentArchiveSamples** (or **retainRecentArchiveSamples**) parameter, which determines the minimum number of recent archive records to retain; however, the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Archive Terminal Accounting workspace

This workspace contains statistical information about terminals that are on nodes used by message flows during the current archive interval. There is one row per terminal for each node in a message flow. Typically, you link to this workspace for a specific message flow and node; all terminals for which data is displayed belong to that message flow and node. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Terminal Accounting attributes” on page 254.

Default views

This workspace contains the following default views:

- **Archive Terminal Accounting:** Presents a table that contains archive statistics for terminals that are on nodes used by message flows, for the current collection interval.
- **Terminal Invocation Rate:** Presents a bar chart that contains the rate at which terminals were invoked during the current interval.

Predefined links

This workspace contains links to the Recent Archive Terminal Accounting and Historical Archive Terminal Accounting workspaces for the message flow and node that are displayed when you select a row in this workspace. The number of rows of data that are displayed in the Recent Archive Terminal Accounting workspace depends on the value assigned to the **defaultRetainRecentArchiveSamples** (or

retainRecentArchiveSamples) parameter, which indicates the minimum number of recent archive records to retain, although the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Archive Thread Accounting workspace

This workspace contains archive statistics that pertain to threads that are used by message flows. There is one row of data per thread for each message flow for the current archive interval. Typically, you link to this workspace for a specific message flow, so all threads that are displayed relate to that message flow. For archive accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect archive accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Thread Accounting attributes” on page 255.

Default views

This workspace contains the following default views:

- **Archive Thread Accounting:** Presents a table that contains archive statistics for threads used by message flows.
- **Thread Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that was spent processing input messages during the current interval.
- **Thread CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time that was spent processing input messages during the current interval.

Predefined links

This workspace contains links to Recent Archive Thread Accounting and Historical Archive Thread Accounting workspaces for the message flow that is displayed when you select a row in this workspace. The number of rows of data displayed in the Recent Archive Thread Accounting workspace depends on the value assigned to the **defaultRetainRecentArchiveSamples** (or **retainRecentArchiveSamples**) parameter, which indicates the minimum number of recent archive records, although the number of records that is retained and the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Broker Events workspace

The Broker Events workspace lists events that are generated on a particular broker managed system as they occur. The number of events that is displayed depends on the number that are retained by the agent; that value is set in the agent parameter file.

The agent receives broker events data with a subscription to broker event publications. If a particular event is not published by an IBM broker product, it is not available for display in this workspace.

Tip: The Broker Status attribute in the Broker Information attribute group is the best way to detect whether the broker is started or stopped, because the Broker Event indicating that a broker has stopped might not always occur.

For related attribute group, see “Broker Events attributes” on page 193.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised within the whole system.
- **Broker Events:** Presents a table that contains the broker events that were retained most recently.

Predefined links

By default, you can link from the Broker Events table view to the Broker Events History. (Broker Events short-term history collection is inactive by default.)

Broker Information workspace

The Broker Information workspace contains information about a broker-managed system, including the status of the broker, its process ID, information about the operating system on which it runs, its job on z/OS systems, the queue manager associated with its configuration manager, its associated queue manager, and whether or not the agent can connect to the queue manager of the broker. It also contains a graphical view of your broker environment, including execution groups and messages flows.

For related attribute groups, see “Broker Information attributes” on page 194 and “Broker Topology attributes” on page 148.

Default views

This workspace contains the following default views:

- **Broker Information:** Presents a table that contains broker status and definition data.
- **Broker Topology:** Contains a graphical view of your broker environment, including execution groups and messages flows.

Predefined links

By default, you can link from the Broker Information table view to Broker Statistics and Broker Information History (Broker Information short-term history collection is inactive by default.) The broker topology view also includes the following links, available by right-clicking the execution group and message flow icons:

- Message flow:
 - Message Flow Topology workspace
 - Recent Archive Message Flow Accounting
 - Recent Snapshot Message Flow Accounting
 - Message Flow Statistics
 - Message Flow Information
- Execution group:
 - Execution Group Topology

- Execution Group Statistics

In addition to the previous workspaces, the following workspaces are available in specific circumstance:

- If you have the WebSphere MQ Monitoring agent installed on the same system you can also link to Queue Manager Status workspace of the WebSphere MQ Monitoring agent. For more information about the Queue Manager Status workspace see the online help for WebSphere MQ Monitoring agent.
- If you have OS agent installed on the same system you can also link to System Overview workspace of OS agent using the system name given in the workspaces provided by the WebSphere Message Broker Monitoring agent.

Broker Statistics workspace

The Broker Statistics workspace provides high-level message flow statistics that are summarized at the broker level. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input or output.

For related attribute group, see “Broker Statistics attributes” on page 197.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains the message input rate and message output rate for the broker.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times for the broker.
- **Broker Statistics:** Presents a table that contains statistical data summarized for the broker.

Tip: Move the cursor over different areas of a chart to display hover help details about the items in the chart.

Predefined links

By default, you can link from the Broker Statistics table view to the Broker Information and Broker Statistics History workspaces (Broker Statistics short-term history collection is inactive by default.) You can also link to the User Broker Statistics and User Execution Group Statistics workspaces.

In addition to the previous workspaces, if you have an OS agent installed on the same system, you can also link to the System Overview workspace of OS agent from the Broker Statistics table view.

Broker Status workspace (broker-level summary)

The Broker Status workspace summarizes event, status, and definition information for a particular broker.

For related attribute groups, see “Broker Events attributes” on page 193, “Broker Information attributes” on page 194, and “Message Flow Events attributes” on page 218.

Default views

This workspace contains the following default views:

- **Broker Events:** Presents a table that contains the broker events that were retained most recently.
- **Message Flow Events:** Presents a table that contains the message flow events that were retained most recently.
- **Broker Information:** Presents a table that contains broker status and definition data.

Accessing this workspace

To access this workspace, in the Navigator physical view click one of the broker managed system folder icons.

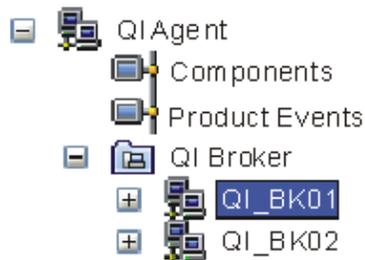


Figure 20. Broker managed system

If only one broker is monitored, the broker managed system folder icon is merged with the **QI Broker** folder icon.



Figure 21. Broker managed system merged with the QI Broker

Broker Summary workspace (multiple monitored brokers)

The Broker Summary workspace contains information about a QI Broker folder node when multiple brokers are monitored by the same agent. It lists information about each monitored broker.

For related attribute group, see “Broker Information attributes” on page 194.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.
- **Broker Information :** Presents a table that contains the broker status and definition data for each broker monitored by the agent.

Accessing this workspace

To access this workspace, in the Navigator physical view click the **QI Broker** folder icon.

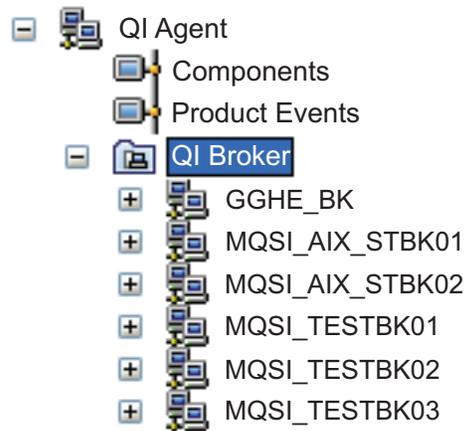


Figure 22. QI broker in the Navigator physical view

CandleMonitor Node Statistics workspace

The CandleMonitor Node Statistics workspace contains message flow statistics for a particular broker-managed system, organized by the CandleMonitor nodes that collect the statistics. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input, output, or other.

For related attribute group, see “CandleMonitor Node Statistics attributes” on page 203.

Default views

This workspace contains the following default views:

- **Current Message Rate:** Presents a bar chart that contains the message flow rate, organized by message processing node.
- **Current Average Message Time :** Presents a bar chart that contains average queue and flow times, organized by message processing node.
- **CandleMonitor Node Statistics:** Presents a table that contains statistical data, organized by CandleMonitor node.

Tip: Move the cursor over different areas of a chart to display hover help details about the items in the chart.

Predefined links

By default, you can link from the CandleMonitor Node Statistics table view to Message Processing Node Information (filtered by execution group and Message Flow), and CandleMonitor Node Statistics History (CandleMonitor Node Statistics short-term history collection is inactive by default.) You can also link to the User CandleMonitor Node Statistics.

Components workspace

The Components workspace contains a list of IBM broker product components and their state at the agent level.

Remember: On distributed systems, all of the IBM broker product components are listed; on z/OS systems, only the brokers that are specified in the parameter file using the MonitorBroker tag are listed in this workspace.

For related attribute group, see “Components attributes” on page 149.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents, as a table, all situations that were raised in the entire system.
- **Components:** Presents, as a table, all installed broker components on distributed systems, and all monitored broker components on z/OS the system.

Predefined links

By default, you can link from this Components table view to the Components History workspace (Components short-term history collection is inactive by default.)

In addition to the previous workspace, the following workspaces are available in specific circumstances:

- If you have the WebSphere MQ Monitoring agent installed on the same system you can also link to Queue Manager Status workspace of the WebSphere MQ Monitoring agent using the queue manager attribute. For more information about the Queue Manager Status workspace, see the online help for the WebSphere MQ Monitoring agent.
- If you have the OS agent installed on the same system you can also link to System Overview workspace of OS agent using the system name given in the workspaces provided by the WebSphere Message Broker Monitoring agent.

Execution Group Information workspace

The Execution Group Information workspace contains details about execution group definitions on a particular broker.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.
- **Execution Group Information:** Presents a table that contains execution group information.

Predefined links

By default, you can link from the Execution Group Information table view to Execution Group Statistics (filtered by execution group), Message Flow Information (filtered by execution group workspaces), and Execution Group Topology workspaces (filtered by execution group).

Execution Group Statistics workspace

The Execution Group Statistics workspace contains high-level message flow statistics summarized at the execution group level for a particular broker-managed system. To obtain data for this workspace, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input or output.

For related attribute group, see “Execution Group Statistics attributes” on page 209.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains message input rate and message output rate, organized by execution group.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times, organized by execution group.
- **Execution Group Statistics:** Presents a table that contains state and statistical data, organized by execution group.

Tip: Move the cursor over different areas of a chart to display hover help information about the items in the chart.

Predefined links

By default, you can link from the Execution Group Statistics table view to Message Flow Statistics (filtered by execution group), Execution Group Information (filtered by execution group), and Execution Group Statistics History (Execution Group Statistics short-term history collection is inactive by default.) You can also link to User Execution Group Statistics and User Message Flow Statistics.

In addition to the previous workspaces, if you have an OS agent installed on the same system you can also link to System Overview workspace of OS Agent from the Execution Group Statistics table view.

Execution Group Topology workspace

The Execution Group Topology workspace contains a graphical view of an execution group, including its message flows and status information such as whether each message flow is started or stopped.

Default views

This workspace contains the following default views:

- **Execution Group Topology:** Contains a graphical view of the execution group, including its message flows and status information.
- **Message Flow Information:** Contains information about the message flows in the execution group.

Predefined links

By default, you can link from processing nodes in either the Execution Group Topology table view or the Execution Group Topology view to the following workspaces to view information about the selected message flow.

- Message Flow Topology
- Recent Archive Message Flow Accounting
- Recent Snapshot Message Flow Accounting

- Message Flow Statistics

You can also link from the execution group node in the Execution Group Topology view to the Execution Group Statistics and Execution Group Information workspaces.

Accessing this workspace

You can link to the Execution Group Topology workspace as follows:

- In the Broker Topology view of the Broker Information workspace, right-click the execution group that you are interested in, and then click **Link To > Execution Group Topology**.
- In the Execution Group Information workspace, right-click a row in the table view, and then click **Link To > Execution Group Topology**.

Message Flow Events workspace

The Message Flow Events workspace lists events that are generated by CandleMonitor nodes on a particular broker-managed system as they occur. The number of events that is displayed depends on the number of events that is retained by the agent; the value is set in the agent parameter file.

For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to other and message text specified in the **eventMessage** attribute.

For related attribute group, see “Message Flow Events attributes” on page 218.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.
- **Message Flow Events:** Presents a table that contains the Message Flow Events that were retained most recently.

Predefined links

By default, you can link from this Message Flow Events table view to the Message Flow Events History workspace. (Message Flow Events short-term history collection is inactive by default.)

Message Flow Information workspace

The Message Flow Information workspace contains information about the message flows that are defined on a particular broker-managed system, organized by execution group. This workspace also contains a range of settings that you can use to modify the workspace contents.

For related attribute group, see “Message Flow Information attributes” on page 220.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.

- **Message Flow Information:** Presents a table that contains message flow information.

Predefined links

By default, you can link from the Message Flow Information table view to the Message Flow Statistics (filtered by execution group and Message Flow), Message Processing Node Information (filtered by execution group and Message Flow), and Message Flow Topology workspace (filtered by message flow) workspaces.

Message Flow Statistics workspace

The Message Flow Statistics workspace contains high-level message flow statistics summarized at the message flow level for a particular broker-managed system. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to input or output.

For related attribute group, see “Message Flow Statistics attributes” on page 223.

Default views

This workspace contains the following default views:

- **Current Message Rates:** Presents a bar chart that contains message input and output rate, organized by message flow.
- **Current Average Message Time:** Presents a bar chart that contains average queue and flow times, organized by message flow.
- **Message Flow Statistics:** Presents a bar chart that contains state and statistical data, organized by message flow.

Tip: Move the cursor over different areas of a chart to display hover help information about the items in the chart.

Predefined links

By default, you can link from the Message Flow Statistics table view to the CandleMonitor Node Statistics (filtered by execution group and Message Flow), Message Flow Information (filtered by execution group and Message Flow), and Message Flow Statistics History views (Message Flow Statistics short-term history collection is inactive by default.) You can also link to sub-flow Statistics, User Message Flow Statistics, User sub-flow Statistics, and User CandleMonitor Node Statistics.

Message Flow Topology workspace

The Message Flow Topology workspace contains a graphical view of a message flow, including its nodes and connection between nodes.

For related attribute group, see “Message Flow Node Topology attributes” on page 157.

Default views

This workspace contains the following default views:

- **Message Flow Topology:** Contains a graphical view of the message flow, including its nodes and connections between nodes.
- **Message Processing Node Information:** Contains information about each node in the message flow.

Predefined links

By default, you can link from processing nodes in either the Message Flow Topology table view or the Message Flow Topology view to the following workspaces to view information about the selected processing node.

- Message Processing Node Attributes
- Recent Archive Node Accounting
- Recent Snapshot Node Accounting

If you have the WebSphere MQ Monitoring agent installed on the same system, you can also link to Queue Status workspace of WebSphere MQ Monitoring agent from the MQ nodes in Message Flow Topology workspace. For more information about the Queue Status workspace see the online help for WebSphere MQ Monitoring agent.

Accessing this workspace

You can link to the Message Flow Topology workspace as follows:

- In the Broker Topology view of the Broker Information workspace, right-click the message flow that you are interested in, and then click **Link To > Message Flow Topology**.
- In the Message Flow Information workspace, right-click a row in the table view, and then click **Link To > Message Flow Topology**.

Message Processing Node Attributes workspace

The Message Processing Node Attributes workspace lists all attribute values for the selected message processing node (within a message flow).

Message processing node attributes are defined in the broker product Message Brokers Toolkit when you configure a node within a message flow. Each node has its own distinct attributes that you can use to customize node functionality.

This workspace only contains data if node data collection is enabled in the `kqi.xml` configuration file. Note, however, that if message brokers in your environment are configured with a large number of message flows, or message flows that have a lot of message processing node data, enabling this option might drastically degrade system performance. For more information about enabling node data collection for all message brokers, see “defaultCollectNodeData” on page 24, and for a single message broker, see “collectNodeData” on page 31.

For related attribute group, see “Message Processing Node attributes” on page 228.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.
- **Message Processing Node Attributes:** Presents a table that contains a list of each attribute and its value for the selected processing node.

Accessing this workspace

To access this workspace, in the Message Processing Node Information table view, right-click a row, and then click **Message Processing Node Attributes**.

Message Processing Node Information workspace

The Message Processing Node Information workspace contains information about nodes that process information, organized by execution group and message flow, for a particular broker managed system.

This workspace only contains data if node data collection is enabled in the `kqi.xml` configuration file. Note, however, that if message brokers in your environment are configured with a large number of message flows, or message flows that have a lot of message processing node data, enabling this option might drastically degrade system performance.

For more information about enabling node data collection for all message brokers, see “`defaultCollectNodeData`” on page 24, and for a single message broker, see “`collectNodeData`” on page 31.

For related attribute group, see “Message Processing Node Information attributes” on page 229.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Message Processing Node Information:** Presents a table that contains Message Processing Node Information.

Predefined links

By default, you can link from the Message Processing Node Information table view to the CandleMonitor Node Statistics (filtered by Execution Group and Message Flow), Message Processing Node Attributes (filtered by Execution Group, Message Flow, and Processing Node Label), and Message Processing Nodes with Queue (filtered by Attribute 1 when it is a queue name).

If you have the WebSphere MQ Monitoring agent installed on the same system you can also link to Queue Status workspace of WebSphere MQ Monitoring agent using available queue and queue manager names. For more information about the Queue Status workspace see the online help for WebSphere MQ Monitoring agent.

Message Processing Nodes with Queue workspace

The Message Processing Nodes with Queue workspace lists all message processing nodes in the enterprise that have the same queue name.

You can also use Message Processing Nodes with Queue workspace as the target workspace of a link from a workspace that is associated with the MQ monitoring agent (if you are using the Tivoli OMEGAMON DE feature package for Tivoli Enterprise Portal). To define such a link, study the example link provided in the Message Processing Node Information table view to this workspace, and see the Tivoli Enterprise Portal online help for information about customizing a workspace link.

For related attribute group, see “Message Processing Node Information attributes” on page 229.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Message Processing Nodes with <queue name>:** Presents a table that contains message processing node information for nodes throughout your enterprise for which the queue name attribute matches the queue name used in the link.

Accessing this workspace

To access this workspace, in the Message Processing Node Information table view, right-click a row that has a queue name attribute in the Attribute 1 column, and then click **Message Processing Nodes with Queue**.

Information presented in the Message Processing Nodes with <queue name> table view

The Message Processing Nodes with <queue name> table view presents data that is provided by the Message Processing Node Information attributes described in “Message Processing Node Information workspace” on page 303.

Multicast Group Statistics workspace

The Multicast Group Statistics workspace contains current multicast group statistics for a particular broker-managed system. For publish-subscribe statistics to be available in this workspace, you need to issue the QI Change Properties command.

For related attribute group, see “Multicast Group Statistics attributes” on page 233.

Default views

This workspace contains the following default views:

- **Multicast Group Statistics:** Presents multicast statistics at the group level in the form of a table.
- **Messages Sent Rates:** Presents a bar chart that contains the rate at which multicast messages were sent by the broker during the latest interval for the selected multicast group.
- **Bytes Sent Rates:** Presents a bar chart that contains number of bytes in multicast messages that have been sent and resent by the broker during the latest interval for the selected multicast group.

Predefined links

By default, you can link from the Multicast Group Statistics table view to Recent Multicast Group Statistics and Historical Multicast Group Statistics workspaces. You can also link to the Multicast Topic Statistics workspace. The Multicast Topic Statistics workspace uses the Multicast Topic Statistics attributes described in “Multicast Topic Statistics workspace” on page 305.

Multicast Summary Statistics workspace

The Multicast Summary Group Statistics workspace contains a summary of current multicast group statistics for a particular broker-managed system. For publish-subscribe statistics to be available in this workspace, you need to issue the QI Change Properties command.

For related attribute group, see “Multicast Summary Statistics attributes” on page 234.

Default views

This workspace contains the following default views:

- **Multicast Summary Statistics:** Presents a summary of multicast statistics in the form of a table.
- **Messages Sent Rate:** Presents a bar chart that contains rate at which multicast messages were sent by the broker over the latest interval.
- **Bytes Sent Rate:** Presents a bar chart that contains number of bytes in multicast messages that have been sent and resent by the broker over the latest interval.

Predefined links

By default, you can link from the Multicast Summary Statistics table view to Recent Multicast Summary Statistics and Historical Multicast Summary Statistics workspaces. You can also link to the Multicast Group Statistics and Multicast Topic Statistics workspaces. The Multicast Group Statistics workspace uses the Multicast Group Statistics attributes described in “Multicast Group Statistics workspace” on page 304, and the Multicast Topic Statistics workspace uses the Multicast Topic Statistics attributes described in “Multicast Topic Statistics workspace.”

Multicast Topic Statistics workspace

The Multicast Topic Statistics workspace contains current multicast topic statistics for a particular broker-managed system. For publish-subscribe statistics to be available in this workspace, you need to issue the QI Change Properties command.

For related attribute group, see “Multicast Topic Statistics attributes” on page 235.

Default views

This workspace contains the following default views:

- **Multicast Topic Statistics:** Presents multicast statistics at the topic level in the form of a table.
- **Messages Sent Rates:** Presents a bar chart that contains rate at which multicast messages were sent by the broker during the interval for the selected multicast topic.
- **Bytes Sent Rates:** Presents a bar chart that contains number of bytes in multicast messages that have been sent and resent by the broker during the interval for the selected multicast topic.

Predefined links

By default, you can link from the Multicast Topic Statistics table view to the Recent Multicast Topic Statistics and the Historical Multicast Topic Statistics workspaces.

Neighbors workspace

The Neighbors workspace contains a list of the neighbors of a particular broker in the publish subscribe topology, and also their connection attributes.

For related attribute group, see “Neighbors attributes” on page 239.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Neighbors:** Presents, as a table, neighbor information for all of the broker's neighbors.

Predefined links

By default, you can link from the Neighbors table view to the Neighbor Broker Information (which provides details about the neighbor broker, such as the neighbor broker's name). The Neighbor Broker Information workspace uses the Broker Information attributes described in “Broker Information workspace” on page 294.

In addition to the previous workspace, if you have the WebSphere MQ Monitoring agent installed on the same system, you can link to Queue Manager Status workspace of the WebSphere MQ Monitoring agent using available connection queue manager name. For more information about these workspaces, see the online help for WebSphere MQ Monitoring agent.

Neighbor Broker Information workspace

The Neighbor Broker Information workspace provides details about the neighbor broker, such as the neighbor broker's name for the selected row in the Neighbors workspace.

For related attribute group, see “Broker Information attributes” on page 194.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Neighbor Broker Information:** Presents a table that contains broker status and definition data for selected neighbor broker.

Accessing this workspace

To access this workspace, in the Neighbors workspace, right-click the table view, and then click **Link To > Neighbor Broker Information**.

Product Events workspace

The Product Events workspace contains a list of product events that are generated by a particular agent when a problem occurs that affects the agent ability to collect data from the broker. The number of events displayed depends on the number of events that are retained by the agent, as set by the **retainProductEvents** parameter in the agent configuration file.

For an explanation of the agent-generated product events and how they can be used in problem determination, see *Troubleshooting Guide*.

For related attribute group, see “Product Events attributes” on page 185.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Product Events:** Presents a table that contains the product events that were retained most recently.

Predefined links

By default, you can link from this Product Events table view to the Product Events History. (Product Events short-term history collection is disabled by default.)

Publish-Subscribe Statistics workspace

The Publish-Subscribe Statistics workspace contains information about a broker-managed system. It presents the current data from KQITPSST. For publish-subscribe statistics to be available in this workspace, you need to issue the QI Change Properties command.

For related attribute group, see “Publish-Subscribe Statistics attributes” on page 245.

Default views

This workspace contains the following default views:

- **Publish-Subscribe Statistics:** Presents all publish-subscribe statistics, excluding multicast data, in the form of a table.
- **Client Message Rates:** Presents a bar chart that contains the rate at which messages were sent and received by the broker.
- **Client Byte Rates:** Presents a bar chart that contains the rate at which messages are sent and received by the broker.
- **Neighbor Message Rates:** Presents a bar chart that contains all rates of messages of the broker and its neighboring brokers.
- **Neighbor Byte Rates:** Presents a bar chart that contains the rate at which bytes are sent and received between the selected broker and other neighboring brokers.

Predefined links

By default, you can link from the Publish-Subscribe Statistics table view to the Recent Publish-Subscribe Statistics and Historical Publish-Subscribe Statistics workspaces.

Retained Publications workspace

The Retained Publications workspace contains a list of topic and subscription points that have a publication that is retained, the publication of which is retained within a particular broker-managed system.

Remember: The data in this workspace is updated no more frequently than the length of the cycle indicated by the `defaultStatisticInterval` (or, if specified for a specific broker, `statisticInterval`). The default for this interval is 1 minute. If the parameter setting has not been modified, it can take up to 1 minute for a change to retained publication data to be displayed in the workspace.

For related attribute group, see “Retained Publications attributes” on page 248.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in the entire system.
- **Retained Publications:** Presents a table bar chart that contains all retained publications for the broker.

Snapshot Message Flow Accounting workspace

This workspace contains rows of snapshot statistics that pertain to entire message flows, for the current snapshot collection interval. In this top-level view, there is one row per monitored message flow, which contains the data collected for the message flow during the current collection interval. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the QI Change Flow Stats command.

If the broker has not been set to collect the required data, no data is displayed in this workspace. The data displayed in this view is available to situations by default.

For related attribute group, see “Message Flow Accounting attributes” on page 213.

Default views

This workspace contains the following default views:

- **Message Flow Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time, in microseconds, that was spent processing messages during the current snapshot interval.
- **Message Flow CPU Time:** Presents a bar chart that contains the minimum, maximum, and average CPU time, in microseconds, that was spent processing messages during the current snapshot interval.
- **Message Flow Throughput:** Presents the total input messages (flow instance start count), total committed messages (flow complete count) and total number of backouts performed (flow exception count).
- **Message Flow Start Rate:** Presents the total input message rate (flow instance start count per second).
- **Snapshot Message Flow Accounting:** Presents a table that contains accounting statistics, organized by message flow, for the current snapshot interval.

Predefined links

This workspace contains links to the Snapshot Thread Accounting and Snapshot Node Accounting workspaces. This workspace also contains links to the Recent Snapshot Message Flow Accounting and Historical Snapshot Message Flow Accounting workspaces for the selected message flow. The number of rows of data displayed in the Recent Snapshot Message Flow Accounting workspace depends on the value assigned to the **defaultRetainRecentSnapshotSamples** (or **retainRecentSnapshotSamples**) parameter, which indicates the minimum number of recent snapshot records that are retained; however, the number of records retained

might be greater, depending on requirements for retaining data for historical data collection.

Snapshot Node Accounting workspace

This workspace presents rows of statistics for nodes used by message flows for the current snapshot interval. There is one row per node for each message flow. Typically, you link to this workspace for a specific message flow; all nodes that are displayed belong to that message flow. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Node Accounting attributes” on page 236.

Default views

This workspace contains the following default views:

- **Snapshot Node Accounting:** Presents a table that contains snapshot statistics for nodes that are used by message flows.
- **Node Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average time that nodes spend processing input messages.
- **Node Throughput:** Presents the node invocation count.
- **Node Throughput Rate:** Presents the node invocation count per second.
- **Node CPU Time:** Presents a bar chart that contains the minimum, maximum, and average amount of CPU time that nodes used to process input messages.

Predefined links

This workspace contains a link to the Snapshot Terminal Accounting workspace. This workspace also contains links to the Recent Snapshot Node Accounting and Historical Snapshot Node Accounting workspaces for the selected message flow. The number of rows of data displayed in the Recent Snapshot Node Accounting workspace depends on the value assigned to the **defaultRetainRecentSnapshotSamples** (or **retainRecentSnapshotSamples**) parameter, which indicates minimum number of recent archive records that must be retained; however, the number of records that are retained might be greater than specified by these parameters, depending on the data retention requirements of historical data collection.

Snapshot Terminal Accounting workspace

This workspace contains statistics that related to terminals that are on nodes used by message flows during the current snapshot interval. For each message flow, there is one row per terminal on each node. Typically, you link to this workspace for a specific message flow and node; all terminals for which data is displayed belong to that message flow and node. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Terminal Accounting attributes” on page 254.

Default views

This workspace contains the following default views:

- **Snapshot Terminal Accounting:** Presents a table that contains snapshot statistics for terminals that are on nodes that are used by message flows.
- **Terminal Invocation Rate:** Presents the chart that contains the rate at which terminals are invoked.

Predefined links

This workspace contains links to the Recent Snapshot Terminal Accounting and Historical Snapshot Terminal Accounting workspaces for the message flow and node selected in this workspace. The number of rows of data displayed in the Recent Snapshot Terminal Accounting workspace depends on the value assigned to the **defaultRetainRecentArchiveSamples** (or **retainRecentArchiveSamples**) parameter, which indicates the minimum number of recent archive records that must be retained; the number of records that are retained and available for viewing might be greater than the number specified by these variables.

Snapshot Thread Accounting workspace

This workspace presents snapshot statistics that pertain to threads that are used by message flows. There is one row of data per thread for each message flow for the current snapshot interval. Typically, you link to this workspace for a specific message flow; all threads that displayed relate to a single message flow. For snapshot accounting statistics to be available in this workspace, you need to configure the brokers to collect it and set the destination of the statistics output (-o) to xml. You can configure the broker to collect snapshot accounting statistics by issuing the QI Change Flow Stats command.

The data displayed in this view is available to situations by default.

For related attribute group, see “Thread Accounting attributes” on page 255.

Default views

This workspace contains the following default views:

- **Snapshot Thread Accounting:** Presents a table that contains snapshot statistics for threads that are used by message flows.
- **Thread Elapsed Time:** Presents a bar chart that contains the minimum, maximum, and average amount of time that threads spent processing input messages during the collection interval.
- **Thread CPU Time:** Presents a bar chart that contains the minimum, maximum, and average amount of CPU time that threads used processing input messages during the collection interval.

Predefined links

This workspace contains links to the Recent Snapshot Thread Accounting and Historical Snapshot Thread Accounting workspaces for the message flow selected in this workspace. The number of rows of data displayed in the Recent Snapshot Thread Accounting workspace depends on the value assigned to the **defaultRetainRecentSnapshotSamples** (or **retainRecentSnapshotSamples**) parameter, which indicates the minimum number of recent archive records that

must be retained; however, the number of records that are retained might be greater than specified by these parameters, depending on the data retention requirements of historical data collection.

Sub-Flow Statistics workspace

The Sub-Flow Statistics workspace provides message flow statistics summarized at the subflow level.

If you access the Sub-Flow Statistics workspace from the Navigator view, all subflows in the specified broker are displayed organized by execution groups and message flows. If you link to this workspace from the Message Flow Statistics workspace for a given message flow, only the subflows that are associated with that message flow are displayed. For this workspace to contain data, the broker must have at least one deployed CandleMonitor node with the **type** attribute set to `subFlowOutput` or `subFlowInput`.

For related attribute group, see “Sub-Flow Statistics attributes” on page 249.

Default views

This workspace contains the following three views:

- **Current Message Rates:** Presents a bar chart that contains the message input and output rates, organized by subflow.
- **Current Average Message Time:** Presents a bar chart that contains the average queue and flow times, organized by subflow.
- **Sub-Flow Statistics:** Presents a table that contains state and statistical data, organized by subflow.

Predefined links

By default, you can link from the Sub-Flow Statistics table view to the Sub-Flow Statistics History workspace. Links are also available to the CandleMonitor Node Statistics workspace, so that the associated nodes and base statistical data can be viewed for each node that is part of the subflow, and to User Sub-Flow Statistics and User CandleMonitor Node Statistics workspaces.

Subscriptions workspace

The Subscriptions workspace contains a list of subscriptions to topics on a particular broker-managed system.

For related attribute group, see “Subscriptions attributes” on page 253.

Default views

This workspace contains the following default views:

- **Enterprise Event Log:** Presents a table that contains all situations that were raised in an entire system.
- **Subscriptions:** Presents a table that contains all subscription information for the broker.

Appendix A. Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. With the major accessibility features in this product, users can do the following things:

- Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on the screen.

In addition, the product documentation was modified to include the following features to aid accessibility:

- All documentation is available in both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software.
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

Magnifying what is displayed on the screen

You can enlarge information on the product windows using facilities provided by the operating systems on which the product is run. For example, in a Microsoft Windows system environment, you can lower the resolution of the screen to enlarge the font sizes of the text on the screen. Refer to the documentation provided by your operating system for more information.

Navigating the interface using the keyboard

Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

Appendix B. Disk space requirements for historical data tables

This appendix provides information about disk space requirements for historical data tables.

Historical data tables

The amount of default space required for a 24-hour period on a monitored system varies greatly depending on customer configuration. The following estimates are taken from an example Windows system with the following WebSphere Message Broker components installed: configuration manager, user name server, Message Brokers Toolkit, and a single message broker. The broker has two monitored execution groups, four message flows being monitored configured with two subflows, 12 CandleMonitor nodes, a total of 50 nodes with 150 terminals, and two threads per message flow. Accounting statistics has been turned on to collect all possible archive data for the four message flows with the default interval in place at 60 minutes, and the default agent parameter setting is in place to only collect archive data for history, not snapshot data.

Important: The historical collection interval must be set to the same value for each of the five statistics tables (broker statistics, execution group statistics, message flow statistics, subflow statistics, and CandleMonitor node statistics). Also, the historical collection interval must be set to the same value for each of the four accounting tables (message flow accounting, thread accounting, node accounting, and terminal accounting). The historical collection interval can be set to be a different value for the two groups of tables (statistics and accounting). The default for all collection is 15 minutes. The WebSphere Message Broker Monitoring agent does not recommend multiple collection intervals for either the statistics group of tables or the accounting group of tables, because different settings of historical collection interval lengths can cause the data recorded historically to be inaccurate for these attribute groups.

Table 19. Historical data tables of the WebSphere Messaging Broker Monitoring agent (WebSphere Message Broker V7.0 or later)

Attribute history table	File name for historical data	Estimated space required per managed system per 24-hour period
Accounting Message Flow Statistics	kqitacmf	168 kilobytes
Accounting Node Statistics	kqitacnd	2371 kilobytes
Accounting Terminal Statistics	kqitactr	7340 kilobytes
Accounting Thread Statistics	kqitacth	310 kilobytes
Broker Status	kqitbrks	111 kilobytes
Broker Status Events	kqitbsev	28 kilobytes
Components	kqitcomp	330 kilobytes
Execution Group Status	kqitegrs	144 kilobytes
File Resource Statistics	kqitrsfl	147 kilobytes

Table 19. Historical data tables of the WebSphere Messaging Broker Monitoring agent (WebSphere Message Broker V7.0 or later) (continued)

Attribute history table	File name for historical data	Estimated space required per managed system per 24-hour period
JDBC Connection Pools Resource Statistics	kqitrsjd	10968 kilobytes
JVM Resource Statistics	kqitrsjv	34956 kilobytes
Message Flow Status	kqitmfls	1935 kilobytes
Message Processing Nodes	kqitmpns	2857 kilobytes
Monitor Node Base Statistics	kqitmfn	1791 kilobytes
Monitor Node Broker Statistics	kqitmnr	66 kilobytes
Monitor Node Events	kqitmnev	23 kilobytes
Monitor Node Execution Group Statistics	kqitmneg	179 kilobytes
Monitor Node Message Flow Statistics	kqitmnmf	834 kilobytes
Monitor Node Sub-Flow Statistics	kqitmnsf	255 kilobytes
ODBC Resource Statistics	kqitrsod	8775 kilobytes
Parsers Resource Statistics	kqitrsps	26730 kilobytes
Product Events	kqitprev	29 kilobytes
SOAPInput Resource Statistics	kqitrssp	9956 kilobytes
Total Default Space		107446 kilobytes

Table 20. Historical data tables of the WebSphere Messaging Broker Monitoring agent (WebSphere Message Broker V6 all releases)

Attribute history table	File name for historical data	Estimated space required per managed system per 24-hour period
ACL Entries	kqitdsea	1383 kilobytes
Broker Events	kqitbrev	51 kilobytes
Broker Information	kqitbrkr	155 kilobytes
Broker Statistics	kqitstbr	90 kilobytes
CandleMonitor Node Statistics	kqitstfn	3231 kilobytes
Components	kqitcomp	330 kilobytes
Execution Group Information	kqitdfeg	244 kilobytes
Execution Group Statistics	kqitsteg	275 kilobytes
Message Flow Accounting	kqitasmf	154 kilobytes
Message Flow Events	kqitflev	42 kilobytes
Message Flow Information	kqitdfmf	2218 kilobytes
Message Flow Statistics	kqitstmf	1410 kilobytes

Table 20. Historical data tables of the WebSphere Messaging Broker Monitoring agent (WebSphere Message Broker V6 all releases) (continued)

Attribute history table	File name for historical data	Estimated space required per managed system per 24-hour period
Message Processing Node Information	kqitdffn	4261 kilobytes
Multicast Group Statistics	kqitpsms	68 kilobytes
Multicast Summary Statistics	kqitpsmg	98 kilobytes
Multicast Topic Statistics	kqitpsmt	140 kilobytes
Neighbors	kqitdsen	157 kilobytes
Node Accounting	kqitasnd	2054 kilobytes
Product Events	kqitprev	29 kilobytes
Publish-subscribe statistics	kqitpsst	80 kilobytes
Retained Publications	kqitdser	620 kilobytes
Sub-Flow Statistics	kqitstsf	447 kilobytes
Subscriptions	kqitdses	6345 kilobytes
Terminal Accounting	kqitastr	6047 kilobytes
Thread Accounting	kqitasth	260 kilobytes
Total Default Space		30189 kilobytes

Historical table record sizes

The following tables present the record size and frequency for each historical data table.

Table 21. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V7.0 or later)

History table	Record size	Frequency
Accounting Message Flow Statistics	1796 bytes	1 row per message flow with accounting feature turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data
Accounting Node Statistics	2024 bytes	1 row per node per message flow with the accounting feature being turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data
Accounting Terminal Statistics	2088 bytes	1 row per terminal per node per message flow with the accounting feature being turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data

Table 21. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V7.0 or later) (continued)

History table	Record size	Frequency
Accounting Thread Statistics	1656 bytes	1 row per thread per message flow with the accounting feature being turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data
Broker Status	1184 bytes	1 row per interval
Broker Status Events	972 bytes	1 row per broker event publication (pure event table, so not affected by interval)
Components	880 bytes	1 row per WebSphere broker component installed on system monitored by agent per interval
Execution Group Status	772 bytes	1 row per execution group per interval
File Resource Statistics	1052 bytes	1 row per execution group per interval (pure event table, so not affected by interval)
JDBC Connection Pools Resource Statistics	1300 bytes	1 row per execution group every 20 seconds (pure event table, so not affected by interval)
JVM Resource Statistics	1036 bytes	4 rows per execution group every 20 seconds (pure event table, so not affected by interval)
Message Flow Status	1720 bytes	1 row per message flow per interval
Message Processing Nodes	2540 bytes	1 row per message processing node per interval
Monitor Node Base Statistics	1592 bytes	1 row per CandleMonitor node per interval
Monitor Node Broker Statistics	704 bytes	1 row per interval
Monitor Node Events	1604 bytes	1 row per message flow event detected (pure event table, so not affected by interval)
Monitor Node Execution Group Statistics	956 bytes	1 row per monitored execution group per interval
Monitor Node Message Flow Statistics	1112 bytes	1 row per monitored message flow per interval
Monitor Node Sub-Flow Statistics	1360 bytes	1 row per monitored subflow per interval

Table 21. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V7.0 or later) (continued)

History table	Record size	Frequency
ODBC Resource Statistics	1040 bytes	1 row per execution group every 20 seconds (pure event table, so not affected by interval)
Parsers Resource Statistics	1056 bytes	3 rows or more per execution group every 20 seconds (pure event table, so not affected by interval)
Product Events	1204 bytes	1 row per product monitoring event noted by agent (pure event table, so not affected by interval)
SOAPInput Resource Statistics	1180 bytes	1 row per execution group every 20 seconds (pure event table, so not affected by interval)

Table 22. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V6 all releases)

History table	Record size	Frequency
ACL Entries	1844 bytes	1 row per ACL entry per interval
Broker Events	1740 bytes	1 row per broker event publication (pure event table, so not affected by interval)
Broker Information	1648 bytes	1 row per broker per interval
Broker Statistics	960 bytes	1 row per broker per interval
CandleMonitor Node Statistics	2872 bytes	1 row per CandleMonitor node per interval
Components	880 bytes	1 row per WebSphere broker component installed on system monitored by agent per interval
Execution Group Information	1300 bytes	1 row per execution group per interval
Execution Group Statistics	1468 bytes	1 row per monitored execution group per interval
Message Flow Accounting	1644 bytes	1 row per message flow with accounting feature turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data
Message Flow Events	2884 bytes	1 row per message flow event detected (pure event table, so not affected by interval)
Message Flow Information	1972 bytes	1 row per message flow per interval

Table 22. Historical table record sizes of the WebSphere Messaging Broker Monitoring agent (WebSphere Messaging Broker V6 all releases) (continued)

History table	Record size	Frequency
Message Flow Statistics	1880 bytes	1 row per monitored message flow per interval
Message Processing Node Information	3788 bytes	1 row per message processing node per interval
Multicast Group Statistics	720 bytes	1 row per interval
Multicast Summary Statistics	1040 bytes	1 row per interval
Multicast Topic Statistics	1488 bytes	1 row per interval
Neighbors	836 bytes	1 row per neighbor to the broker per interval
Node Accounting	1752 bytes	1 row per node per message flow with the accounting feature being turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data
Product Events	1204 bytes	1 row per product monitoring event noted by agent (pure event table, so not affected by interval)
Publish-Subscribe Statistics	856 bytes	1 row per interval
Retained Publications	2204 bytes	1 row per retained publication per interval
Sub-Flow Statistics	2384 bytes	1 row per monitored subflow per interval
Subscriptions	3384 bytes	1 row per subscription per interval
Terminal Accounting	1720 bytes	1 row per terminal per node per message flow with the accounting feature being turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data
Thread Accounting	1384 bytes	1 row per thread per message flow with the accounting feature being turned on per accounting interval for archive data and, if selected for history, per 20 seconds for snapshot data

Historical space requirement worksheets

Use the following worksheets to estimate expected file sizes and additional disk space requirements for your site. A sample calculation is provided for each historical data collection table.

Table 23. Components (kqitcomp) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	880 bytes	$(60/15 \times 24 \times 880 \times 4) / 1024$ for 4 installed components	330 kilobytes

Table 24. Product Events (kqitprev) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
N/A	1204 bytes	$(1204 \times 25) / 1024$ for 25 product monitoring events occurring	29 kilobytes

Table 25. Accounting Message Flow Statistics (kqitacmf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	1796 bytes	$(60/60 \times 24 \times 1796 \times 4) / 1024$ for 4 monitored message flows	168 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 26. Accounting Node Statistics (kqitacnd) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	2024 bytes	$(60/60 \times 24 \times 2024 \times 50) / 1024$ for a total of 50 nodes in monitored message flows	2371 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 27. Accounting Terminal Statistics (kqitactr) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	2088 bytes	$(60/60 \times 24 \times 2088 \times 150)/1024$ for a total of 150 terminals in monitored message flows	7340 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 28. Accounting Thread Statistics (kqitacth) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	1656 bytes	$(60/60 \times 24 \times 1656 \times 8)/1024$ for 4 monitored message flows with 2 threads each	310 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 29. Broker Status (kqitbrks) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
5 min.	1184 bytes	$(60/15 \times 24 \times 1184 \times 1)/1024$ for 1 broker	111 kilobytes

Table 30. Broker Status Events (kqitbsev) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
N/A	972 bytes	$(972 \times 30)/1024$ for 30 broker events occurring	28 kilobytes

Table 31. Execution Group Status (kqitegrs) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	772 bytes	$(60/15 \times 24 \times 772 \times 2)/1024$ for 2 execution groups	144 kilobytes

Table 32. File Resource Statistics (kqitrsfl) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
20 sec.	1052 bytes	$(60/20 \times 60 \times 24 \times 1052 \times 1 \times 2)/1024$ for 2 execution groups	147 kilobytes

Table 33. JDBC Connection Pools Resource Statistics (kqitrsjd) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
20 sec.	1300 bytes	$(60/20 \times 60 \times 24 \times 1300 \times 1 \times 2)/1024$ for 2 execution groups	10968 kilobytes

Table 34. JVM Resource Statistics (kqitrsjv) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
20 sec.	1036 bytes	$(60/20 \times 60 \times 24 \times 1036 \times 4 \times 2)/1024$ for 2 execution groups	34956 kilobytes

Table 35. Message Flow Status (kqitmfls) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1720 bytes	$(60/15 \times 24 \times 1720 \times 12)/1024$ for 12 message flows	1935 kilobytes

Table 36. Message Processing Nodes (kqitmpns) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	2540 bytes	$(60/15 \times 24 \times 2540 \times 12)/1024$ for 12 message flows	2857 kilobytes

Table 37. Monitor Node Base Statistics (kqitmfn) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1592 bytes	$(60/15 \times 24 \times 1592 \times 12)/1024$ for 12 CandleMonitor nodes in flows	1791 kilobytes

Table 38. Monitor Node Broker Statistics (kqitmnr) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	704 bytes	$(60/15 \times 24 \times 704 \times 1)/1024$ for 1 broker	66 kilobytes

Table 39. Monitor Node Events (kqitmnev) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
N/A	1604 bytes	$(1604 \times 15)/1024$ for 15 message flow events occurring	23 kilobytes

Table 40. Monitor Node Execution Group Statistics (kqitmneg) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	956 bytes	$(60/15 \times 24 \times 956 \times 2)/1024$ for 2 monitored execution groups	179 kilobytes

Table 41. Monitor Node Message Flow Statistics (kqitmnmf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1112 bytes	$(60/15 \times 24 \times 1112 \times 8)/1024$ for 4 monitored message flows	834 kilobytes

Table 42. Monitor Node Sub-Flow Statistics (kqitmnsf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1360 bytes	$(60/15 \times 24 \times 1360 \times 2)/1024$ for 2 monitored sub-flows	255 kilobytes

Table 43. ODBC Resource Statistics (kqitrsod) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
20 sec.	1040 bytes	$(60/20 \times 60 \times 24 \times 1040 \times 1 \times 2)/1024$ for 2 execution groups	8775 kilobytes

Table 44. Parsers Resource Statistics (kqitrsps) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
20 sec.	1056 bytes	$(60/20 \times 60 \times 24 \times 1056 \times 3 \times 2)/1024$ for 2 execution groups	26730 kilobytes

Table 45. SOAPInput Resource Statistics (kqitrssp) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
20 sec.	1180 bytes	$(60/20 \times 60 \times 24 \times 1180 \times 1 \times 2)/1024$ for 2 execution groups	9956 kilobytes

Table 46. ACL Entries (kqitdsea) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1844 bytes	$(60/15 \times 24 \times 1844 \times 8) / 1024$ for 8 ACL entries	1383 kilobytes

Table 47. Broker Events (kqitbrev) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
N/A	1740 bytes	$(1740 \times 30) / 1024$ for 30 broker events occurring	51 kilobytes

Table 48. Broker Information (kqitbrkr) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1648 bytes	$(60/15 \times 24 \times 1648 \times 1) / 1024$ for 1 broker	155 kilobytes

Table 49. Broker Statistics (kqitstbr) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	960 bytes	$(60/15 \times 24 \times 960 \times 1) / 1024$ for 1 broker	90 kilobytes

Table 50. CandleMonitor Node Statistics (kqitstfn) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	2872 bytes	$(60/15 \times 24 \times 2872 \times 12) / 1024$ for 12 CandleMonitor nodes in flows	3231 kilobytes

Table 51. Execution Group Information (kqitdfeg) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1300 bytes	$(60/15 \times 24 \times 1300 \times 2) / 1024$ for 2 execution groups	244 kilobytes

Table 52. Execution Group Statistics (kqitsteg) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1468 bytes	$(60/15 \times 24 \times 1468 \times 2) / 1024$ for 2 monitored execution groups	275 kilobytes

Table 53. Message Flow Accounting (kqitasmf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	1644 bytes	$(60/60 \times 24 \times 1644 \times 4) / 1024$ for 4 monitored message flows	154 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 54. Message Flow Events (kqitflev) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
N/A	2884 bytes	$(2884 \times 15) / 1024$ for 15 message flow events occurring	42 kilobytes

Table 55. Message Flow Information (kqitdfmf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1972 bytes	$(60/15 \times 24 \times 1972 \times 12) / 1024$ for 12 message flows	2218 kilobytes

Table 55. Message Flow Information (kqitdfmf) worksheet (continued)

Interval	Record size	Formula	Expected file size per 24-hour period

Table 56. Message Flow Statistics (kqitstmf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1880 bytes	$(60/15 \times 24 \times 1880 \times 8) / 1024$ for 4 monitored message flows	1410 kilobytes

Table 57. Message Processing Node Information (kqitdfn) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	3788 bytes	$(60/15 \times 24 \times 3788 \times 12) / 1024$ for 12 message flows	4261 kilobytes

Table 58. Multicast Group Statistics (kqitpsms) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	720 bytes	$(60/15 \times 24 \times 720) / 1024$	68 kilobytes

Table 59. Multicast Summary Statistics (kqitpsmg) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1040 bytes	$(60/15 \times 24 \times 1040) / 1024$	98 kilobytes

Table 60. Multicast Topic Statistics (kqitpsmt) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	1488 bytes	$(60/15 \times 24 \times 1488) / 1024$	140 kilobytes

Table 61. Neighbors (kqitdsen) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	836 bytes	$(60/15 \times 24 \times 836 \times 2) / 1024$ for 2 neighbors to the broker	157 kilobytes

Table 62. Node Accounting (kqitasnd) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	1752 bytes	$(60/60 \times 24 \times 1752 \times 50) / 1024$ for a total of 50 nodes in monitored message flows	2054 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 63. Publish-subscribe statistics (kqitpsst) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.856	856 bytes	$(60/15 \times 24 \times 856) / 1024$	80 kilobytes

Table 64. Retained Publications (kqitdser) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	2204 bytes	$(60/15 \times 24 \times 2204 \times 3) / 1024$ for 3 retained publications	620 kilobytes

Table 65. Sub-Flow Statistics (kqitstsf) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	2384 bytes	$(60/15 \times 24 \times 2384 \times 2) / 1024$ for 2 monitored sub-flows	447 kilobytes

Table 65. Sub-Flow Statistics (kqitstsf) worksheet (continued)

Interval	Record size	Formula	Expected file size per 24-hour period

Table 66. Subscriptions (kqitdses) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
15 min.	3384 bytes	$(60/15 \times 24 \times 3384 \times 20) / 1024$ for 20 subscriptions	6345 kilobytes

Table 67. Terminal Accounting (kqitastr) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	1720 bytes	$(60/60 \times 24 \times 1720 \times 150) / 1024$ for a total of 150 terminals in monitored message flows	6047 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

Table 68. Thread Accounting (kqitasth) worksheet

Interval	Record size	Formula	Expected file size per 24-hour period
1 hour *	1384 bytes	$(60/60 \times 24 \times 1384 \times 8) / 1024$ for 4 monitored message flows with 2 threads each	260 kilobytes

Remember: * This is the default interval; even if you set the history interval to less, the data can only be produced as often as the default interval occurs.

In the worksheet examples, the minimum collection interval unit of 15 minutes is used. You can create a summary table that provides a representative disk storage space figure for all of the history files and archived files for a one-week time period, if all collection is done at the remote agent managed system. To do so, multiply the expected file size per 24 hours total times seven. Note that historical collection cannot be turned on for those tables not collected by default. If historical data is desired for those tables, a much longer collection interval than the default

15 minutes is required because the data is not expected to change often. You must spread the disk space requirements among the systems where data collection is performed.

Historical disk space summary worksheet

The following tables are disk space summary worksheets for the WebSphere Message Broker Monitoring agent.

Table 69. Disk space summary worksheet for historical tables (WebSphere Message Broker V7.0 or later)

History table	Historical data table size (kilobytes) (24 hours)	Number of archives	Subtotal space required (kilobytes)
Accounting Message Flow Statistics			
Accounting Node Statistics			
Accounting Terminal Statistics			
Accounting Thread Statistics			
Broker Status			
Broker Status Events			
Components			
Execution Group Status			
File Resource Statistics			
JDBC Connection Pools Resource Statistics			
JVM Resource Statistics			
Message Flow Status			
Message Processing Nodes			
Monitor Node Base Statistics			
Monitor Node Broker Statistics			
Monitor Node Events			
Monitor Node Execution Group Statistics			
Monitor Node Message Flow Statistics			
Monitor Node Sub-Flow Statistics			

Table 69. Disk space summary worksheet for historical tables (WebSphere Message Broker V7.0 or later) (continued)

History table	Historical data table size (kilobytes) (24 hours)	Number of archives	Subtotal space required (kilobytes)
ODBC Resource Statistics			
Parsers Resource Statistics			
Product Events			
SOAPInput Resource Statistics			
Total disk space required			

Table 70. Disk space summary worksheet for historical tables (WebSphere Message Broker V6 all releases)

History table	Historical data table size (kilobytes) (24 hours)	Number of archives	Subtotal space required (kilobytes)
ACL Entries			
Broker Events			
Broker Information			
Broker Statistics			
CandleMonitor Node Statistics			
Components			
Execution Group Information			
Execution Group Statistics			
Message Flow Accounting			
Message Flow Information			
Message Flow Events			
Message Flow Statistics			
Message Processing Node Information			
Multicast Group Statistics			
Multicast Summary Statistics			
Multicast Topic Statistics			
Neighbors			
Node Accounting			
Product Events			

Table 70. Disk space summary worksheet for historical tables (WebSphere Message Broker V6 all releases) (continued)

History table	Historical data table size (kilobytes) (24 hours)	Number of archives	Subtotal space required (kilobytes)
Publish-Subscribe Statistics			
Retained Publications			
Subscriptions			
Sub-Flow Statistics			
Terminal Accounting			
Thread Accounting			
Total disk space required			

Appendix C. Language codes

Table 71 lists the languages supported by the WebSphere Message Broker Monitoring agent, and their corresponding language codes.

Table 71. Language codes for the supported languages

Language	code
English	en_US
German	de_DE
Spanish	es_ES
French	fr_FR
Italian	it_IT
Japanese	ja_JP
Korean	ko_KR
Portuguese (Brazilian)	pt_BR
Simplified Chinese	zh_CN
Traditional Chinese	zh_TW

Appendix D. Architecture codes

Abbreviations are used in IBM Tivoli software to represent the various operating system architectures. The following table shows the most current listing of these abbreviations.

This information can also be found in the following file on UNIX systems:
install_dir/registry/archdsc.tbl.

Table 72. Operating system architecture abbreviations

Abbreviation	Operating system architecture
aix513	AIX v5.1 (32 bit)
aix516	AIX v5.1 (64 bit)
aix523	AIX v5.2 (32 bit)
aix526	AIX v5.2 (64 bit)
aix533	AIX v5.3 (32 bit)
aix536	AIX v5.3 (64 bit)
citrix	Citrix Metaframe
hp10	HP-UX v10.01/10.10
hp102	HP-UX v10.20
hp11	HP-UX v11
hp116	HP-UX v11 (64 bit)
li622	Linux Intel v2.2
li6223	Linux Intel v2.2 (32 bit)
li624	Linux Intel v2.4
li6242	Linux Intel v2.4 GCC 2.9.5 (32 bit)
li6243	Linux Intel v2.4 (32 bit)
li6245	Linux Intel v2.4 GCC 2.9.5 (64 bit)
li6246	Linux Intel v2.4 (64 bit)
li6262	Linux Intel v2.6 GCC 2.9.5 (32 bit)
li6263	Linux Intel v2.6 (32 bit)
li6265	Linux Intel v2.6 GCC 2.9.5 (64 bit)
li6266	Linux Intel v2.6 (64 bit)
ls322	Linux zSeries, 2.2 kernel
ls3223	Linux zSeries, v2.2 (32 bit)
ls3226	Linux zSeries, v2.2 (64 bit)
ls324	Linux zSeries, v2.4
ls3243	Linux zSeries, v2.4 (32 bit)
ls3246	Linux zSeries, v2.4 (64 bit)
ls3262	Linux S390 v2.6 GCC 2.9.5 (32 bit)
ls3263	Linux S390 v2.6 (32 bit)
ls3265	Linux S390 v2.6 GCC 2.9.5 (64 bit)

Table 72. Operating system architecture abbreviations (continued)

Abbreviation	Operating system architecture
ls3266	Linux S390 v2.6 (64 bit)
osf1	Digital UNIX (before V5.0)
os390	OS/390 [®] or z/OS systems
os400	OS/400 [®]
sol24	Solaris v2.4
sol25	Solaris v2.5
sol26	Solaris v2.6
sol273	Solaris v7 (32 bit)
sol276	Solaris v7 (64 bit)
sol283	Solaris v8 (32 bit)
sol286	Solaris v8 (64 bit)
sol293	Solaris v9 (32 bit)
sol296	Solaris v9 (64 bit)
sol503	Solaris v10 (32 bit)
sol506	Solaris v10 (64 bit)
sol603	Solaris v10 Opteron (32 bit)
sol606	Solaris v10 Opteron (64 bit)
tsf50	Tru64 v5.0
unix	UNIX system
winnt	Windows 2000 and Windows 2003 Server

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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 OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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 and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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 www.ibm.com/legal/copytrade.shtml.

Intel and Itanium 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, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

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 Sun Microsystems, Inc. in the United States, other countries, or both.

Glossary

This glossary includes terms and definitions for ITCAM Agents for WebSphere Messaging.

The following cross-references are used in this glossary:

- See refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- See also refers you to a related or contrasting term.

To view glossaries for other IBM products, go to www.ibm.com/software/globalization/terminology (opens in new window).

A

access The ability to read, update, or otherwise use a resource. Access to protected resources is usually controlled by system software.

access management

The process of controlling access to IT services, data, or other assets.

address space

The range of addresses available to a computer program or process. Address space can refer to physical storage, virtual storage, or both. See also buffer pool.

agent Software that is installed to monitor systems. An agent collects data about an operating system, a subsystem, or an application.

aggregation

The process of collecting, interpreting, and sorting data from various locations into a single file.

alert A message or other indication that signals an event or an impending event. See also event.

attribute

1. The application properties that are measured and reported on, such as the amount of memory that is used or a message ID. See also attribute group.
2. Data that is associated with a component. For example, a host name,

IP address, or the number of hard drives can be attributes associated with a server component.

attribute group

A set of related attributes that can be combined in a view or a situation. See also attribute, situation, view.

audit A process that logs modifications to the database and plan.

B

batch

1. Pertaining to a group of jobs to be run on a computer sequentially with the same program with little or no operator action.
2. A group of records or data processing jobs brought together for processing or transmission.

batch job

A predefined group of processing actions submitted to the system to be performed with little or no interaction between the user and the system.

batch mode

The condition established so that batch processing can be performed.

BPM See business performance management.

broker

A set of execution processes that host one or more message flows. See also execution group, message flow.

buffer pool

An area of memory into which data pages are read and in which they are modified and held during processing. See also address space.

bundle

A packaged collection of software products that is purchased as one item and that has its own product identifier (PID).

business performance management (BPM)

The monitoring, management, and tuning

of business performance in real time through the analysis of business relevant information.

C

channel

A WebSphere MQ object that defines a communication link between two queue managers (message channel) or between a client and a queue manager (MQI channel). See also queue manager.

client A software program or computer that requests services from a server. See also host, server.

cluster

1. In WebSphere MQ, a group of two or more queue managers on one or more computers, providing automatic interconnection, and allowing queues to be advertised among them for load balancing and redundancy.
2. In Microsoft Cluster Server, a group of computers, connected together and configured in such a way that, if one fails, MSCS performs a failover, transferring the state data of applications from the failing computer to another computer in the cluster and reinitiating their operation there.

cluster queue manager

A queue manager that is a member of a cluster. A queue manager can be a member of more than one cluster.

component

A software item that is part of a software product, and might be separately identified, but is not individually licensed.

condition

1. An expression that consists of an agent attribute, an operator such as greater than or equal to, and a value. It can be read as, "If - system condition - compared to - value - is true. See also situation.
2. A test of a situation or state that must be in place for a specific action to occur.

configuration

The manner in which the hardware and

software of a system, subsystem, or network are organized and interconnected.

D

data set

The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

dead-letter queue (DLQ)

A queue to which a queue manager or application sends messages that cannot be delivered to their correct destination.

deployment

The process of installing and configuring a software application and all its components.

DLQ See dead-letter queue.

dynamic queue

A local queue created when a program opens a model queue object.

E

enterprise

The composite of all operational entities, functions, and resources that form the total business concern and that require an information system.

event An occurrence of significance to a task or system. Events can include completion or failure of an operation, a user action, or the change in state of a process. See also alert, situation.

execution group

A named process or set of processes within a broker in which message flows are executed. The broker is guaranteed to enforce some degree of isolation between message flows in distinct execution groups by ensuring that they execute in separate address spaces, or as unique processes. See also broker, message flow.

F

full repository

A complete set of information about every queue manager in a cluster. This set of information is called the repository or sometimes the full repository and is usually held by two of the queue managers in the cluster. See also partial repository.

function

Any instruction or set of related instructions that performs a specific operation.

H

host A computer that is connected to a network and that provides an access point to that network. The host can be a client, a server, or both a client and server simultaneously. See also client, server.

hot standby

A redundant server that, if the primary server or hub server fails, assumes the responsibilities of the failed server.

I

integration

The software development activity in which separate software components are combined into an executable whole.

L

launch-in-context

An operation in which a user starts a secondary application from a primary application to perform a specific task. Using the parameters, navigation instructions, and user credentials that are supplied by the primary application, the secondary application opens to the specific place in which to complete the task.

M

managed object

A resource that is subject to management as viewed from a systems management perspective. Examples of such resources are a connection, a scalable system, or a line.

managed system

A system that is being controlled by a given system management application.

manager

An entity that monitors or controls one or more managed objects by (a) receiving notifications regarding the objects and (b) requesting management operations to modify or query the objects.

message flow

A sequence of processing steps that execute in the broker when an input message is received. Message flows are defined in the workbench by including a number of message flow nodes, each of which represents a set of actions that define a processing step. The connections in the flow determine which processing steps are carried out, in which order, and under which conditions. See also broker, execution group, subflow.

middleware

Software that acts as an intermediate layer between applications or between client and server. It is used most often to support complex, distributed applications in heterogeneous environments.

module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

monitoring agent

See agent.

multi-instance queue manager

A queue manager that is configured to share the use of queue manager data with other queue manager instances. One instance of a running multi-instance queue manager is active, other instances are on standby ready to take over from the active instance. See also queue manager.

O

offering

1. A logical unit of software packaging and sharing that has a managed development and maintenance life cycle and customer visible attributes (offering features, product IDs, licenses, maintenance contracts, and so forth). An offering is a serviceable software asset that is orderable by an IBM customer. It can be a collection of common components, assemblies, and other offerings.
2. The element or integrated set of elements (hardware, software, services) designed to satisfy the wants and needs of current and/or prospective customers. A solution is the application of the offering in a specific customer environment. See also solution.

P

partial repository

A partial set of information about queue managers in a cluster. A partial repository is maintained by all cluster queue managers that do not host a full repository. See also full repository.

performance management

1. The discipline that encompasses capacity planning, collecting performance data, and tuning resources.
2. The management processes and systems needed to effectively deliver business services.

PID See product identifier.

platform

The combination of an operating system and hardware that makes up the operating environment in which a program runs.

policy A set of considerations that influence the behavior of a managed resource or a user.

product ID

See product identifier.

product identifier (PID, product ID)

A unique value that identifies an IBM

software product. Every mainframe and distributed IBM software product has a PID.

Q

query In a Tivoli environment, a combination of statements that are used to search the configuration repository for systems that meet certain criteria. The query object is created within a query library.

queue An object that holds messages for message-queueing applications. A queue is owned and maintained by a queue manager.

queue manager

A component of a message queuing system that provides queuing services to applications. See also channel, multi-instance queue manager.

queue-sharing group

In WebSphere MQ for z/OS, a group of queue managers in the same sysplex that can access a single set of object definitions stored in the shared repository, and a single set of shared queues stored in the coupling facility.

R

registry

A repository that contains access and configuration information for users, systems, and software.

S

sampled event

An event that happens when a situation becomes true. Situations sample data at regular intervals. When the situation is true, it opens an event, which is closed automatically when the situation returns to false.

segment

A set of customers/buyers within a market who have common wants, needs, characteristics and buying behavior. These wants and needs are sufficiently homogeneous that a consistent set of strategies, marketing campaigns and sales tactics can be directed toward them.

server A software program or a computer that

provides services to other software programs or other computers. See also client, host.

service request

A request from a user for help, information, advice, or access to an IT service.

severity level

A classification for an event that indicates its degree of severity. The predefined severity levels, in order of descending severity, are: fatal, critical, warning, minor, harmless, and unknown.

situation

A set of conditions that, when met, creates an event. See also attribute group, condition, event.

snapshot

A capture of data at a point time for performance analysis.

solution

A combination of products that addresses a particular customer problem or project.

started task

In MVS, a process that begins at system start and runs unattended. Started tasks are generally used for critical applications. The UNIX equivalent of a started task is a daemon.

state

An indication associated with an icon, color, and severity level assigned to a situation at a point in time. A situation can reflect one of the following states: critical, warning, or informational.

status The true or false condition of a situation.

subflow

A sequence of processing steps, implemented using message flow nodes, that is designed to be embedded in a message flow or in another subflow. A subflow must include at least one Input or Output node. A subflow can be executed by a broker only as part of the message flow in which it is embedded, and therefore it cannot be deployed. See also message flow.

subnet

See subnetwork.

subnetwork (subnet)

A network that is divided into smaller independent subgroups, which still are interconnected.

subscription

In a Tivoli environment, the process of identifying the subscribers that the profiles are distributed to.

summarization

The process of aggregating events and then submitting the set of events with a much smaller number of summary events.

system

A computer and its associated devices and programs.

T

TCP/IP

See Transmission Control Protocol/Internet Protocol.

threshold

A customizable value for defining the acceptable tolerance limits (maximum, minimum, or reference limit) for an application resource or system resource. When the measured value of the resource is greater than the maximum value, less than the minimum value, or equal to the reference value, an exception or event is raised.

transaction

A unit of processing consisting of one or more application programs, affecting one or more objects, that is initiated by a single request.

Transmission Control Protocol/Internet Protocol (TCP/IP)

An industry-standard, nonproprietary set of communication protocols that provides reliable end-to-end connections between applications over interconnected networks of different types.

transmission queue

A local queue on which prepared messages destined for a remote queue manager are temporarily stored.

U

upgrade

To install a new version or release of a product to replace an earlier version or release of the same product.

user profile

A description of a user that includes such information as user ID, user name, password, access authority, and other attributes that are obtained when the user logs on.

V

view A window pane, or frame, in a workspace. It may contain data from an agent in a chart or table, or it may contain a terminal session or notepad, for example. A view can be split into two separate, autonomous views. See also attribute group.

W

workspace

1. A window comprised of one or more views.
2. In Tivoli management applications, the working area of the user interface, excluding the Navigator pane, that displays one or more views pertaining to a particular activity. Predefined workspaces are provided with each Tivoli application, and systems administrators can create customized workspaces.

Index

A

- accessibility xiii
 - keyboard 313
 - overview 313
 - screen 313
- Accounting Message Flow Statistics attributes 137
- Accounting Node Statistics attributes 140
- Accounting Terminal Statistics 141
- Accounting Thread Statistics attributes 143
- ACL Entries attributes 192
- ACL Entries workspace 290
- activateNode attribute 260
- active traces
 - detecting 62
- active/active clustering
 - AIX, prerequisites 127
 - Windows 117
 - Windows, prerequisites 115
- active/passive clustering
 - AIX, prerequisites 129
 - Windows 122
 - Windows, prerequisites 120
- ad hoc queries 102
- agent configuration
 - disabling data collection 37
 - disabling persistent data collection 37
 - dividing broker monitoring
 - Linux 35
 - UNIX 35
 - Windows 35
 - enabling persistent data collection 37
 - monitoring brokers 34
- agent instances
 - creating, for HACMP clustering 131
 - creating, for MSCS clustering 117, 122
 - creating, on Linux 18
 - creating, on UNIX 18
 - creating, on Windows 17
- agent parameter files
 - overview 9
 - statement syntax 11
- agent parameters
 - adding 12
 - descriptions 18
 - modifying 12
- Agent Status workspace 290
- agentId attribute of KqiAgent 19
- agents
 - creating shutdown files for 134
 - creating startup files for 133
- alias attribute of MonitorBroker 28
- application message flow performance
 - monitoring 82
- architecture codes 337
- Archive Accounting Message Flow Statistics workspace 271
- Archive Accounting Node Statistics workspace 272
- Archive Accounting Terminal Statistics workspace 273
- Archive Accounting Thread Statistics workspace 274
- Archive Message Flow Accounting workspace 290
- Archive Node Accounting workspace 291
- Archive Terminal Accounting workspace 292
- Archive Thread Accounting workspace 293
- attributes
 - Accounting Message Flow Statistics 137
 - Accounting Node Statistics 140
 - Accounting Terminal Statistics 141
 - Accounting Thread Statistics 143
 - ACL Entries 192
 - Broker Events 193
 - Broker Information 194
 - Broker Statistics 197
 - Broker Status 145
 - Broker Status Events 147
 - Broker Topology 148, 201
 - CandleMonitor node 258
 - CandleMonitor Node Statistics 203
 - CandleMonitor nodes
 - activateNode 260
 - collectQueueTime 259
 - eventMessage 259
 - subFlowName 259
 - type 258
 - Components 149, 206
 - Execution Group Information 207
 - Execution Group Statistics 209
 - Execution Group Status 151
 - File Resource Statistics 152
 - JDBC Connection Pools Resource Statistics 153
 - JVM Resource Statistics 154
 - Message Flow Accounting 213
 - Message Flow Events 218
 - Message Flow Information 220
 - Message Flow Node Topology 157
 - Message Flow Statistics 223
 - Message Flow Status 154
 - Message Flow Topology 227
 - Message Processing Node 228
 - Message Processing Node Information 229
 - Message Processing Nodes 158
 - Monitor Node Base Statistics 161
 - Monitor Node Broker Statistics 164
 - Monitor Node Events 168
 - Monitor Node Execution Group Statistics 170
 - Monitor Node Message Flow Statistics 174
 - Monitor Node Sub-Flow Statistics 178
 - Multicast Group Statistics 233
 - Multicast Summary Statistics 234
 - Multicast Topic Statistics 235
 - Neighbors 239
 - Node Accounting 236
 - ODBC Resource Statistics 182
 - overview 3
 - Parsers Resource Statistics 183
 - Processing Node Attributes 184
 - Product Events 185, 239
 - Publish-Subscribe Statistics 245
 - Retained Publications 248
 - SOAP Input Resource Statistics 191
 - Sub-Flow Statistics 249
 - Subscriptions 253
 - Terminal Accounting 254
 - Thread Accounting 255

- authorization
 - Take Action command users 61

B

- books
 - See publications xi, xiii
- broker accounting and statistics data
 - comparing with CandleMonitor node statistics data 69
- broker data collection
 - overview 4
- broker environment topology
 - viewing 87
- Broker Events attributes 193
- Broker Events workspace 293
- Broker Information attributes 194
- Broker Information workspace 294
- Broker Statistics attributes 197
- Broker Statistics workspace 295
- Broker Status attributes 145
- Broker Status Events attributes 147
- Broker Status Events workspace 276
- Broker Status workspace 275, 295
- Broker Summary workspace 296
- Broker Topology attributes 148, 201
- brokers
 - capacity planning 85
 - configuration, verifying 84
 - dividing monitoring
 - Linux 35
 - UNIX 35
 - Windows 35
 - monitoring configuration 34
 - starting automatically 64

C

- CandleMonitor node
 - known problems, using with WebSphere Message Broker 8.0 56
- CandleMonitor node configuration variables
 - changing values
 - Linux 54
 - UNIX 54
 - Windows 54
 - KQIActivateNode 262
 - KQIActivateNodeForBROKERNAME 262
 - KQIMemorySize 261
 - KQINodeTrace 262
 - KQITempDirectory 261
- CandleMonitor node files 44
- CandleMonitor Node Statistics attributes 203
- CandleMonitor node statistics data
 - comparing with broker accounting statistics data 69
- CandleMonitor Node Statistics workspace 297
- CandleMonitor nodes
 - attributes 258
 - activateNode 260
 - collectQueueTime 259
 - eventMessage 259
 - subFlowName 259
 - type 258
 - best practice 53
 - configuration variables 258
 - configuring
 - Linux 54

- CandleMonitor nodes (*continued*)
 - configuring (*continued*)
 - UNIX 54
 - Windows 54
 - customizing 53
 - deleting
 - Message Brokers Toolkit V6.0 55
 - Message Brokers Toolkit V6.1 55
 - Message Brokers Toolkit V7.0 or later 56
 - inserting in message flows 48
 - installing
 - Linux 42
 - overview 40
 - UNIX 42
 - Windows 41
 - making available
 - Message Brokers Toolkit V6.0 44
 - Message Brokers Toolkit V6.1 46
 - Message Brokers Toolkit V7.0 or later 48
 - monitoring message flows 49, 52
 - monitoring subflows 50
 - monitoring Type I subflows 50
 - monitoring Type II subflows 51
 - overview 39
 - placing in message flows 48
 - prerequisites 40
 - producing event messages 52
- checking response time
 - message flows 85
- cluster environment configuration
 - AIX 127
 - Windows 113
- cluster groups
 - displaying running agents in Tivoli Enterprise Portal, HACMP 133
 - displaying running agents in Tivoli Enterprise Portal, MSCS 119, 124
- clustering configuration
 - active/active
 - AIX, prerequisites 127
 - Windows 117
 - Windows, prerequisites 115
 - active/passive
 - AIX, prerequisites 129
 - Windows 122
 - Windows, prerequisites 120
 - HACMP 130
 - MSCS
 - active/active 117
 - active/active, prerequisites 115
 - active/passive 122
 - active/passive, prerequisites 120
- codes
 - architecture 337
 - language 335
- Cognos reports
 - Broker Daily Availability report 104
 - Broker Elapsed Microseconds report 104
 - Broker Execution Group Daily Availability report 104
 - Broker Execution Group Weekly Availability report 104
 - Broker Weekly Availability report 104
 - installation 101
 - Message Flow Daily Availability report 104
 - Message Flow Detail report 104
 - Message Flow Weekly Availability report 104
 - troubleshooting 110

- collecting trace data
 - on remote systems 63
- collectNodeData attribute of MonitorBroker 31
- collectQueueTime attribute 259
- commands
 - agent managed systems 264
 - broker-managed systems 265
 - restrictions on z/OS 270
 - See Take Actions command 60
 - tacmd configureSystem 16
 - Take Action 4
- commandTimeoutInterval attribute of KqiAgent 26
- Components attributes 149, 206
- Components workspace 276, 298
- configuration
 - agent, remote
 - prerequisites 14
 - through the command line 16
 - through Tivoli Enterprise Portal 15
 - HACMP clustering 130
 - MSCS clustering
 - active/active 117
 - active/passive 122
 - starting historical data collection 94
 - stopping historical data collection 95
- configuration files
 - See parameter files 9
- configuration variables 261
 - CandleMonitor node 258
- ConnectQueueManager tag 33
- ConnectQueueManager tag attributes
 - name 33
 - replyQueueModel 34
 - replyQueueName 33
- conventions
 - typeface xiv
- creating directories, for historical and situation data files 132
- creating multiple agent instances
 - Linux 18
 - UNIX 18
 - Windows 17
- creating shutdown files for the agent 134
- creating startup files for the agent 133
- creating user statistics workspaces 81
- creating workspaces 80
- customizing workspaces 67

D

- data collection
 - broker 4
 - disabling 37
 - historical 6, 91
 - offline 97
- data model
 - Cognos reports 103
- debugging message flows 84
- default parameter files
 - Linux 10
 - UNIX 10
 - Windows 10
 - z/OS 10
- defaultCollectNodeData attribute of KqiAgent 24
- defaultFlowEventInterval attribute of KqiAgent 21
- defaultHistoricalAccountingType attribute of KqiAgent 21
- defaultPersistentBrokerData attribute of KqiAgent 25
- defaultRefreshInterval attribute of KqiAgent 26

- defaultReplyQueueModel attribute of KqiAgent 24
- defaultReplyQueueName attribute of KqiAgent 23
- defaultRetainBrokerEvents attribute of KqiAgent 19
- defaultRetainFlowEvents attribute of KqiAgent 19
- defaultRetainMonitoringEvents attribute of KqiAgent 23
- defaultRetainRecentArchiveSamples attribute of KqiAgent 22
- defaultRetainRecentPubSubSamples attribute of KqiAgent 22
- defaultRetainRecentResourceSamples attribute of KqiAgent 22
- defaultRetainRecentSnapshotSamples attribute of KqiAgent 21
- defaultStatisticInterval attribute of KqiAgent 20
- defaultTakeActionAuthUsers attribute of KqiAgent 24
- defaultWMBInstallDirectory attribute of KqiAgent 27
- defaultWMQInstallDirectory attribute of KqiAgent 28
- deleting CandleMonitor nodes
 - Message Brokers Toolkit V6.0 55
 - Message Brokers Toolkit V6.1 55
 - Message Brokers Toolkit V7.0 or later 56
- determining message delivery failures 83
- directories
 - creating, for storing historical and situation data files 132
 - disabling persistent data collection 37
 - discoveryInterval attribute of KqiAgent 20
 - displaying historical data for a selected time frame 95

E

- education
 - See Tivoli technical training xiii
- enabling persistent data collection 37
- enabling shared memory on AIX 40
- envfileDirectory attribute of MonitorBroker 29
- eventMessage attribute 259
- EXE files
 - kqicm600.exe 44
 - kqicm700.exe 44
- Execution Group Information attributes 207
- Execution Group Information workspace 298
- Execution Group Statistics attributes 209
- Execution Group Statistics workspace 299
- Execution Group Status attributes 151
- execution group topology
 - viewing 89
- Execution Group Topology workspace 299

F

- File Resource Statistics attributes 152
- File Resource Statistics workspace 277
- flowEventInterval attribute of MonitorBroker 29

G

- glossary 343

H

- HACMP clustering configuration
 - creating agent instances 131
 - setting shutdown files for the agent 134
 - setting startup files for the agent 134
- historical data
 - collection 91
 - disk space summary worksheets 331

- historical data (*continued*)
 - offline collection 97
 - space requirement worksheet 321
 - table record size 317
 - tables 315
 - viewing for a selected time frame 95
- historical data collection
 - initial settings 91
 - overview 6
 - starting 94
 - stopping 95
- historical data files
 - creating the storage directories 132
- historicalAccountingType attribute of MonitorBroker 30
- holdTimeForQuery attribute of KqiAgent 23

I

- IBM Tivoli Monitoring
 - OMEGAMON DE feature package 8
 - overview 6
 - Tivoli Enterprise Monitoring agents 7
 - Tivoli Enterprise Monitoring Server 7
 - Tivoli Enterprise Portal 7
- installing CandleMonitor nodes
 - Linux 42
 - overview 40
 - UNIX 42
 - Windows 41
- issuing Take Action commands 61

J

- JDBC Connection Pools Resource Statistics attributes 153
- JDBC Connection Pools Resource Statistics workspace 277
- JVM Resource Statistics attributes 154
- JVM Resource Statistics workspace 278

K

- kqi.xml file
 - adding parameters 12
 - default values 10
 - Linux 9
 - modifying parameters 12
 - parameter format 11
 - UNIX 9
 - Windows 9
- KQIActivateNode parameter 262
- KQIActivateNodeForBROKERNAME parameter 262
- KqiAgent tag 19
- KqiAgent tag attributes
 - agentId 19
 - commandTimeoutInterval 26
 - defaultCollectNodeData 24
 - defaultFlowEventInterval 21
 - defaultHistoricalAccountingType 21
 - defaultPersistentBrokerData 25
 - defaultRefreshInterval 26
 - defaultReplyQueueModel 24
 - defaultReplyQueueName 23
 - defaultRetainBrokerEvents 19
 - defaultRetainFlowEvents 19
 - defaultRetainMonitoringEvents 23
 - defaultRetainRecentArchiveSamples 22
 - defaultRetainRecentPubSubSamples 22

- KqiAgent tag attributes (*continued*)
 - defaultRetainRecentResourceSamples 22
 - defaultRetainRecentSnapshotSamples 21
 - defaultStatisticInterval 20
 - defaultTakeActionAuthUsers 24
 - defaultWMBInstallDirectory 27
 - defaultWMQInstallDirectory 28
 - discoveryInterval 20
 - holdTimeForQuery 23
 - maximumAgentCollectionThreads 27
 - maximumCommandRetryCount 26
 - maximumMessageLength 25
 - persistentDataPath 27
 - refreshInterval 32
 - retainProductEvents 20
 - version 19
 - WMBInstallDirectory 33
 - WMQInstallDirectory 33
- kqicm600.exe file 44
- kqicm600.tar file 44
- kqicm700.exe file 44
- kqicm700.tar file 44
- KQIMemorySize parameter 261
- kqinode.cfg file 53
- kqinode.lil file 39
- kqinode64.lil file 39
- KQINodeTrace parameter 262
- KQITempDirectory parameter 261
- KQIXML file
 - default values 10
 - overview 9
 - parameter format 11

L

- language codes 335
- LIL file
 - kqinode.lil 39
 - kqinode64.lil file 39
- local variables
 - setting, for HACMP clustering 132
 - setting, for MSCS clustering 118, 123

M

- manuals
 - See publications xi, xiii
- maximumAgentCollectionThreads attribute of KqiAgent 27
- maximumCommandRetryCount attribute of KqiAgent 26
- maximumMessageLength attribute of KqiAgent 25
- message delivery failures
 - determining 83
- Message Flow Accounting attributes 213
- Message Flow Events attributes 218
- Message Flow Events workspace 300
- Message Flow Information attributes 220
- Message Flow Information workspace 300
- Message Flow Node Topology attributes 157
- Message Flow Node Topology workspace 278
- Message Flow Statistics attributes 223
- Message Flow Statistics workspace 301
- Message Flow Status attributes 154
- message flow topology
 - viewing 88
- Message Flow Topology attributes 227
- Message Flow Topology workspace 301

- message flows
 - debugging 84
 - determining when failed 63
 - ensuring adequate response time 85
 - events 52
 - monitoring 52
 - starting and stopping at periodic intervals 64, 65
 - stopping when output queue is full 64
- Message Processing Node attributes 228
- Message Processing Node Attributes workspace 302
- Message Processing Node Information attributes 229
- Message Processing Node Information workspace 303
- Message Processing Nodes attributes 158
- Message Processing Nodes with Queue workspace 303
- Monitor Node Base Statistics attributes 161
- Monitor Node Base Statistics workspace 279
- Monitor Node Broker Statistics attributes 164
- Monitor Node Broker Statistics workspace 280
- Monitor Node Events attributes 168
- Monitor Node Events workspace 280
- Monitor Node Execution Group Statistics attributes 170
- Monitor Node Execution Group Statistics workspace 281
- Monitor Node Message Flow Statistics attributes 174
- Monitor Node Message Flow Statistics workspace 282
- Monitor Node Sub-Flow Statistics attributes 178
- Monitor Node Sub-Flow Statistics workspace 282
- MonitorBroker tag 28
- MonitorBroker tag attributes
 - alias 28
 - collectNodeData 31
 - envfileDirectory 29
 - flowEventInterval 29
 - historicalAccountingType 30
 - name 28
 - persistentBrokerData 32
 - retainBrokerEvents 29
 - retainFlowEvents 30
 - retainMonitoringEvents 31
 - retainRecentArchiveSamples 30
 - retainRecentPubSubSamples 30
 - retainRecentResourceSamples 31
 - retainRecentSnapshotSamples 30
 - statisticInterval 29
 - takeActionAuthUsers 30
- monitoring application message flow performance 82
- monitoring message flow input or output 49
- monitoring situations 3
- monitoring subflows 50
- MSCS clustering configuration
 - creating agent instances, active/active 117
 - creating agent instances, active/passive 122
- MSCS clusters
 - description 113
 - limitations 125
 - overview 113
- Multicast Group Statistics attributes 233
- Multicast Group Statistics workspacebao 304
- Multicast Summary Statistics attributes 234
- Multicast Summary Statistics workspace 304
- Multicast Topic Statistics attributes 235
- Multicast Topic Statistics workspace 305

N

- name attribute of ConnectQueueManager 33
- name attribute of MonitorBroker 28
- Neighbor Broker Information workspace 306

- Neighbors attributes 239
- Neighbors workspace 305
- Node Accounting attributes 236

O

- ODBC Resource Statistics attributes 182
- ODBC Resource Statistics workspace 283
- OMEGAMON DE feature package 8
- online publications
 - accessing xiii
- ordering publications xiii

P

- parameter descriptions 18
- parameter files
 - adding 12
 - default values
 - Linux 10
 - UNIX 10
 - Windows 10
 - z/OS 10
 - Linux 9
 - modifying 12
 - statement syntax 11
 - UNIX 9
 - Windows 9
 - z/OS 9
- parameters
 - ConnectQueueManager 33
 - KqiAgent 19
 - MonitorBroker 28
- Parsers Resource Statistics attributes 183
- Parsers Resource Statistics workspace 284
- performance
 - broker monitoring 34
 - data collection 4
- persistent data collection
 - disabling 37
 - enabling 37
- persistentBrokerData attribute of MonitorBroker 32
- persistentDataPath attribute of KqiAgent 27
- policy management 8
- predefined workspaces
 - provided information 2
- Processing Node Attributes 184
- Processing Node Attributes workspace 284
- Product Events attributes 185, 239
- Product Events workspace 285, 306
- publications xi
 - accessing online xiii
 - ordering xiii
- Publish-Subscribe Statistics attributes 245
- Publish-Subscribe Statistics workspace 307

Q

- QI Change Broker command 267
- QI Change Flow Stats command 267
- QI Change Properties command 268
- QI Change Trace Broker command 266
- QI Change Trace Component command 265
- QI Create User Statistics command 81, 269
- QI Delete Retained Publication command 269
- QI Destroy User Statistics command 269

- QI Refresh Broker Data command 270
- QI Reset Statistics command 269
- QI Sample User Statistics command 269
- QI Start Broker command 265
- QI Start Component command 264
- QI Start Message Flow(s) command 268
- QI Stop Broker and its Queue Manager command 266
- QI Stop Broker command 266
- QI Stop Component and its Queue Manager command 264
- QI Stop Component command 264
- QI Stop Message Flow(s) command 268
- QI_Automation_Start_Component situation 59
- QI_Average_Flow_Time_High situation 59
- QI_Broker_Not_Started situation 59
- QI_Broker_QMgr_Not_Connected situation 59
- QI_Exception_Terminal_Invoked situation 59
- QI_Message_Flow_Events situation 59
- QI_MsgFlow_Elapsed_Time_High situation 59
- QI_Product_Events situation 59
- QI_Publication_Expiry_Event situation 59
- QI_Status_Stop_Event situation 59
- QI_Subscription_Expiry_Event situation 59

R

- reflex automation 61
- refreshInterval attribute of KqiAgent 32
- remote configuration
 - through the command line 16
 - through Tivoli Enterprise Portal 15
- remote configuration prerequisites 14
- replyQueueModel attribute of ConnectQueueManager 34
- replyQueueName attribute of ConnectQueueManager 33
- report package 99
- reports
 - creating, Web-based 102
- requirements
 - data availability 67
- retainBrokerEvents attribute of MonitorBroker 29
- Retained Publications attributes 248
- Retained Publications workspace 307
- retainFlowEvents attribute of MonitorBroker 30
- retainMonitoringEvents attribute of MonitorBroker 31
- retainProductEvents attribute of KqiAgent 20
- retainRecentArchiveSamples attribute of MonitorBroker 30
- retainRecentPubSubSamples attribute of MonitorBroker 30
- retainRecentResourceSamples attribute of MonitorBroker 31
- retainRecentSnapshotSamples attribute of MonitorBroker 30

S

- security failures on a subscription, handling 86
- sending Take Action commands 61
- setting variables
 - for HACMP clustering, local 132
 - for MSCS clustering, local 118, 123
- shared memory
 - enabling on AIX 40
- shutdown scripts
 - setting, in HACMP 134
 - writing 134
- situation data files
 - creating the storage directories 132
- situations 59
 - overview 3

- situations (*continued*)
 - QI_Automation_Start_Component 59
 - QI_Average_Flow_Time_High 59
 - QI_Broker_Not_Started 59
 - QI_Broker_QMgr_Not_Connected 59
 - QI_Exception_Terminal_Invoked 59
 - QI_Message_Flow_Events 59
 - QI_MsgFlow_Elapsed_Time_High 59
 - QI_Product_Events 59
 - QI_Publication_Expiry_Event 59
 - QI_Status_Stop_Event 59
 - QI_Subscription_Expiry_Event 59
 - using scenarios
 - collecting trace data on remote systems 63
 - determining when a message flow has failed 63
 - preventing inadvertent use of trace active 62
 - starting and stopping message flows at periodic intervals 64, 65
 - starting stopped brokers automatically 64
 - stopping a message flow with a full output queue 64
 - WMB_Automation_Start_Component 59
 - WMB_Average_Flow_Time_High 59
 - WMB_Broker_Not_Started 59
 - WMB_Broker_QMgr_Not_Connected 59
 - WMB_Exception_Terminal_Invoked 59
 - WMB_Message_Flow_Events 59
 - WMB_MsgFlow_Elapsed_Time_High 59
 - WMB_Product_Events 59
- Snapshot Accounting Message Flow Statistics workspace 285
- Snapshot Accounting Node Statistics workspace 286
- Snapshot Accounting Terminal Statistics workspace 287
- Snapshot Accounting Thread Statistics workspace 288
- Snapshot Message Flow Accounting workspace 308
- Snapshot Node Accounting workspace 309
- Snapshot Terminal Accounting workspace 309
- Snapshot Thread Accounting workspace 310
- SOAP Input Resource Statistics attributes 191
- SOAP Input Resource Statistics workspace 289
- starting message flows at periodic intervals
 - individual message flows 64
 - within an execution group 65
- starting stopped brokers automatically 64
- startup scripts
 - setting, in HACMP 134
 - writing 133
- statisticInterval attribute of MonitorBroker 29
- stopping message flows at periodic intervals
 - individual message flows 64
 - within an execution group 65
- stopping message flows with a full output queue 64
- storage
 - historical data 91
- Sub-Flow Statistics attributes 249
- Sub-Flow Statistics Workspace 311
- subFlowName attribute 259
- subflows
 - monitoring 50, 51
 - monitoring Type I subflows 50
 - monitoring Type II subflows 51
 - overview 50
- subscriptions
 - verifying 86
- Subscriptions attributes 253
- Subscriptions workspace 311

T

- tacmd configureSystem command 16
- tags
 - ConnectQueueManager 33
 - KqiAgent 19
 - MonitorBroker 28
- Take Action commands 59
 - agent managed systems 264
 - authorizing users 61
 - broker-managed systems 265
 - overview 4, 60
 - QI Change Broker 267
 - QI Change Flow Stats 267
 - QI Change Properties 268
 - QI Change Trace Broker 266
 - QI Change Trace Component 265
 - QI Create User Statistics 81, 269
 - QI Delete Retained Publication 269
 - QI Destroy User Statistics 269
 - QI Refresh Broker Data 270
 - QI Reset Statistics 269
 - QI Sample User Statistics 269
 - QI Start Broker 265
 - QI Start Component 264
 - QI Start Message Flow(s) 268
 - QI Stop Broker 266
 - QI Stop Broker and its Queue Manager 266
 - QI Stop Component 264
 - QI Stop Component and its Queue Manager 264
 - QI Stop Message Flow(s) 268
 - restrictions on z/OS 270
 - sending 61
 - using in situations 61
 - using scenarios
 - collecting trace data on remote systems 63
 - determining when a message flow has failed 63
 - preventing inadvertent use of trace active 62
 - starting and stopping message flows at periodic intervals 64, 65
 - starting stopped brokers automatically 64
 - stopping a message flow with a full output queue 64
- WMB Change Broker 267
- WMB Change Flow Stats 267
- WMB Change Properties 268
- WMB Change Trace Broker 266
- WMB Create User Statistics 269
- WMB Destroy User Statistics 269
- WMB Refresh Broker Data 270
- WMB Reset Statistics 269
- WMB Sample User Statistics 269
- WMB Start Broker 265
- WMB Start Message Flow(s) 268
- WMB Stop Broker 266
- WMB Stop Broker and its Queue Manager 266
- WMB Stop Message Flow(s) 268
- takeActionAuthUsers attribute of MonitorBroker 30
- TAR files
 - kqicm600.tar 44
 - kqicm700.tar 44
- Terminal Accounting attributes 254
- Thread Accounting attributes 255
- Tivoli Common Reporting
 - creating ad-hoc reports 102
 - creating Web-based reports 102
 - data model 103
 - installing Cognos reports 101
 - overview 99

- Tivoli Common Reporting (*continued*)
 - prerequisites 99
 - running Cognos reports 102
 - sample reports 104
 - supported attribute groups 99
 - troubleshooting 110
- Tivoli Enterprise Monitoring agents 7
- Tivoli Enterprise Monitoring Server 7
- Tivoli Enterprise Portal
 - configuring, to list agents in the cluster groups, HACMP 133
 - configuring, to list agents in the cluster groups, MSCS 119, 124
 - historical data collection 6
 - overview 7
 - Policy management 8
 - Take Action commands 4
- Tivoli Enterprise Portal Server 7
- Tivoli Information Center xiii
- Tivoli technical training xiii
- Tivoli user groups xiv
- trace data
 - collecting on remote systems 63
- training, Tivoli technical xiii
- type attribute 258
- Type I subflows
 - description 50
 - monitoring 50
- Type II subflows
 - description 50
 - monitoring 51
- typeface conventions xiv

U

- user groups, Tivoli xiv
- user statistics workspaces
 - creating 81

V

- variables, configuration 261
- verifying broker configuration 84
- verifying subscriptions, without Message Brokers Toolkit 86
- version attribute of KqiAgent 19
- viewing broker environment topology 87
- viewing execution group topology 89
- viewing message flow topology 88

W

- WebSphere Message Broker Monitoring agent
 - clustering configuration
 - AIX 130
 - Windows 117, 122
 - new in this release 1
 - overview 1, 2
 - what's new 1
- WMB Change Broker command 267
- WMB Change Flow Stats command 267
- WMB Change Properties command 268
- WMB Change Trace Broker command 266
- WMB Create User Statistics command 269
- WMB Destroy User Statistics command 269
- WMB Refresh Broker Data command 270
- WMB Reset Statistics command 269

- WMB Sample User Statistics command 269
- WMB Start Broker command 265
- WMB Start Message Flow(s) command 268
- WMB Stop Broker and its Queue Manager command 266
- WMB Stop Broker command 266
- WMB Stop Message Flow(s) command 268
- WMB_Automation_Start_Component situation 59
- WMB_Average_Flow_Time_High situation 59
- WMB_Broker_Not_Started situation 59
- WMB_Broker_QMgr_Not_Connected situation 59
- WMB_Exception_Terminal_Invoked situation 59
- WMB_Message_Flow_Events situation 59
- WMB_MsgFlow_Elapsed_Time_High situation 59
- WMB_Product_Events situation 59
- WMB_Status_Stop_Event situation 59
- WMBInstallDirectory attribute of KqiAgent 33
- WMQInstallDirectory attribute of KqiAgent 33
- workflow editor 8
- worksheets
 - historical disk space summary 331
 - historical space requirement 321
- workspaces
 - ACL Entries 290
 - Agent Status 290
 - Archive Accounting Message Flow Statistics 271
 - Archive Accounting Node Statistics 272
 - Archive Accounting Terminal Statistics 273
 - Archive Accounting Thread Statistics 274
 - Archive Message Flow Accounting 290
 - Archive Node Accounting 291
 - Archive Terminal Accounting 292
 - Archive Thread Accounting 293
 - Broker Events 293
 - Broker Information 294
 - Broker Statistics 295
 - Broker Status 295
 - Broker Status Events 276
 - Broker Summary 296
 - CandleMonitor Node Statistics 297
 - Components 276, 298
 - creating 80
 - customizing 67
 - data availability conditions 67
 - Execution Group Information 298
 - Execution Group Statistics 299
 - Execution Group Topology 299
 - File Resource Statistics 277
 - Historical Publish-Subscribe Statistics 307
 - JDBC Connection Pools Resource Statistics 277
 - JVM Resource Statistics 278
 - Message Flow Events 300
 - Message Flow Information 300
 - Message Flow Node Topology 278
 - Message Flow Statistics 301
 - Message Flow Topology 301
 - Message Processing Node Attributes 302
 - Message Processing Node Information 303
 - Message Processing Nodes with Queue 303
 - Monitor Node Base Statistics 279
 - Monitor Node Broker Statistics 280
 - Monitor Node Events 280
 - Monitor Node Execution Group Statistics 281
 - Monitor Node Message Flow Statistics 282
 - Monitor Node Sub-Flow Statistics workspace 282
 - Multicast Group Statistics 304
 - Multicast Summary Statistics 304
 - Multicast Topic Statistics 305

- workspaces (*continued*)
 - Neighbor Broker Information 306
 - Neighbors 305
 - ODBC Resource Statistics 283
 - Parsers Resource Statistics 284
 - Processing Node Attributes 284
 - Product Events 285, 306
 - Publish-Subscribe Statistics 307
 - Retained Publications 307
 - Snapshot Accounting Message Flow Statistics 285
 - Snapshot Accounting Node Statistics 286
 - Snapshot Accounting Terminal Statistics 287
 - Snapshot Accounting Thread Statistics 288
 - Snapshot Message Flow Accounting 308
 - Snapshot Node Accounting 309
 - Snapshot Terminal Accounting 309
 - Snapshot Thread Accounting 310
 - SOAP Input Resource Statistics 289
 - Sub-Flow Statistics 311
 - Subscriptions 311
 - summary
 - accounting 78
 - agent and application status 75
 - broker and message flow information 76
 - event 76
 - publish and subscribe 79
 - resource statistics 80
 - statistics 77
 - user statistics 81
 - using scenarios
 - debugging a message flow 84
 - determining message delivery failures 83
 - ensuring reasonable message flow response times 85
 - handling security failures on a subscriptions 86
 - monitoring application message flow performance 82
 - planning broker capacity 85
 - verifying broker configuration 84
 - verifying subscriptions without accessing the Message Brokers Toolkit 86
 - viewing the broker environment topology 87
 - viewing the execution group topology 89
 - viewing the message flow topology 88
 - views 2

X

- XML files
 - kqi.xml
 - default values 10
 - Linux 9
 - parameter format 11
 - UNIX 9
 - Windows 9



Printed in USA

SC14-7524-00

