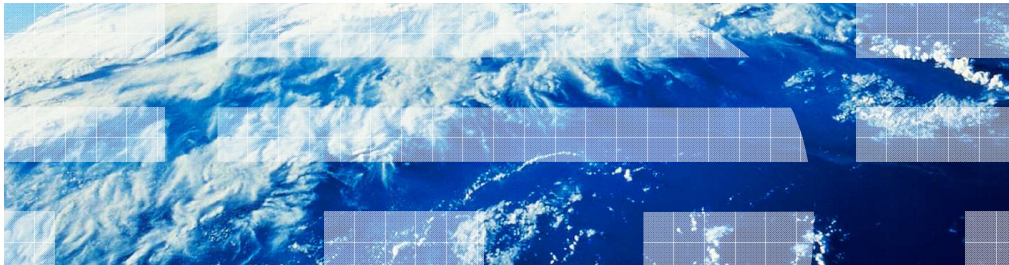


WebSphere Message Broker Version 7

Resource statistics



This session will describe the new statistics features that have been introduced in WebSphere® Message Broker version 7. These statistics enable you to understand what resources are being used within the broker runtime components, and describe how the Message Broker Explorer and other tools can be used to show this information.

Resource statistics overview

- Resource statistics allow you to monitor resource usage for an execution group
 - JVM memory
 - Socket utilization
- You can enable / disable the collection of resource statistics using
 - Message Broker Explorer
 - Broker commands `mqsischangeresourcestats` and `mqsireportresourcestats`
- You can view the resulting statistics using
 - Message Broker Explorer
 - Any tool that can subscribe to the topic that the statistics are published on
- These statistics are referred to as 'resource statistics' in the Information Center and in the rest of this presentation

Message Broker version 7 has introduced a new set of statistics which enables to you better understand the internal behavior of the broker runtime. JVM memory and socket usage for the broker's execution groups are monitored for usage and important aspects of this usage are available as part of this collected information.

To use this feature, you must activate the collection of the resource statistics. This can be done either through the Message Broker Explorer, or through the command interface which has introduced two new commands for this purpose. The broker runtime collects the requested statistics, and publishes then over an interval, using a standard publish / subscribe method.

The collected statistics can be viewed through the Message Broker Explorer. Alternatively, since the statistics are provided using publish / subscribe, any application can subscribe to the specified queues and present the information in any way.

The Information Center refers to this facility as "Resource Statistics".

Enabling resource statistics in Message Broker Explorer

The screenshot shows the IBM WebSphere MQ Explorer interface. The left pane displays a tree view of the broker's components, with the 'default' execution group selected. The right pane shows the 'Execution Group default' properties. A warning message is displayed at the top of the right pane, indicating a performance impact. The 'Statistics' menu is open, showing options to start or stop resource statistics. A table below the menu shows the status of various flows.

Name	Type	Run State	Deployment Time	Modification Time	B...
ST_SOAFADDR_CLIE...	Message Flow	Running	27-Nov-2009 13:08:31...	24-Nov-2009 17:07:44...	C...
ST_SOAFADDR_CLIE...	Message Flow	Running	27-Nov-2009 13:08:31...	24-Nov-2009 17:07:44...	C...
ST_SOAFADDR_SER...	Message Flow	Running	27-Nov-2009 13:09:05...	25-Nov-2009 13:59:20...	C...
ST_SOAFADDR_WO...	xsd zip		27-Nov-2009 13:09:05...	25-Nov-2009 13:59:22...	C...

Warning text shown when statistics are enabled

Start / stop statistics collection using pop-up menu

3 Resource statistics © 2010 IBM Corporation

This slide shows a screen capture of the Message Broker Explorer. With the default execution group highlighted in the broker's navigator, this has opened the pop-up menu. You can select "statistics", then "Start resource statistics". This will tell the broker to start the collection of JMS resourced statistics on the specified execution group. When you do this, a message will be shown indicating that the system will incur a performance overhead.

When statistics collection has been started, the broker will collect and publish the statistics approximately every 20 seconds. The Message Broker Explorer will subscribe to the published data, and will present this information on the statistics views.

Viewing resource statistics in Message Broker Explorer

The screenshot shows the IBM WebSphere MQ Explorer interface. The 'Show View' menu is open, highlighting 'Resource Statistics'. The 'Resource Statistics Graph' window displays a bar chart with the following data series:

Resource	Summary (MB)	Heap Memory (MB)	Non-Heap Memory (MB)
CommittedMemoryInMB	~60	~28	~32
CumulativeGCTimeInSeconds	0	6	0
CumulativeNumberOfGCCollections	0	0	0
InitialMemoryInMB	~60	~28	~32
MaxMemoryInMB	~280	~280	~280
UsedMemoryInMB	~60	~28	~32

The 'Resources Statistics' table at the bottom provides a tabular view of the same data:

name	CommittedMemoryInMB	CumulativeGCTimeInSeconds	CumulativeNumberOfGCCollections	InitialM
summary	60	0	6	32
Heap Memory	32	0	0	32
Non-Heap Mem...	28	0	0	0
Garbage Collect...	0	6	0	0

The published statistics can be viewed by opening a new view in the Message Broker Explorer. This view is the Resource Statistics view, and is opened by selecting “Window”, then “Show View”, then “Resource Statistics”. This will open both a graphical view and a tabular view of the published statistics. You can select either the “JVM” tab or the “Sockets” tab, which will then show the appropriate data.

Filtering resource statistics in Message Broker Explorer

The screenshot displays the IBM WebSphere MQ Explorer interface. The main window shows a 'Resource Statistics Graph' with a bar chart for 'Heap Memory' and a 'Table Metrics' dialog box. A callout points to the 'Filter graph data' button in the top right. Another callout points to the 'Color coded key for selected data' legend. A third callout points to the 'Filter Table' option in the context menu of the table below. A fourth callout points to the 'Table Metrics' dialog box. The table below the graph shows resource statistics for 'JVM Sockets'.

name	CommittedMemoryInMB	CumulativeGCTimeInSeconds	CumulativeNumberOfGCCollections	InitialMemoryInMB	Max...
summary	50	0	6	32	-1
Non-Heap Mem...	28			0	-1
Heap Memory	32			32	256
Garbage Collect...					

The results shown in Message Broker Explorer can be filter to just show the required data. To do this for the graphical view, use the Filter button, as shown on the top right of this screen capture. You can then select or deselect the specific data that you need.

You can perform the same action for the tabular data. In this case, right-click in the area of the table, and make the required selections in the table metrics window.

Enabling resource statistics using the command line

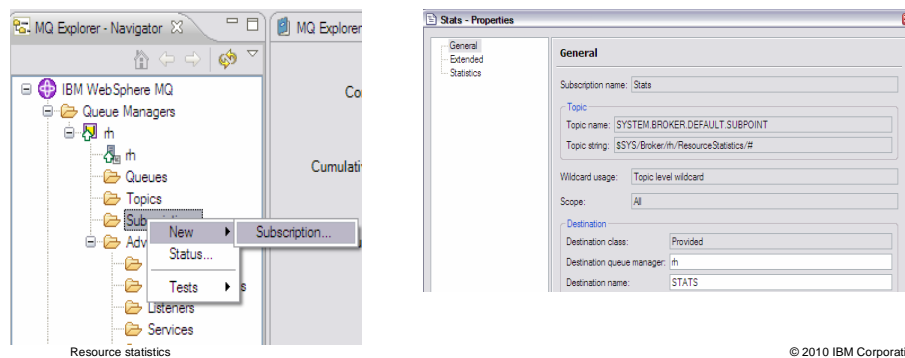
- Enable resource statistics
 - `mqsichangeresourcestats brokerName -c active -e executionGroup`
- Disable resource statistics
 - `mqsichangeresourcestats brokerName -c inactive`
- Display current status
 - `mqsireportresourcestats brokerName -e executionGroup`

```
BIP8940I: Statistics settings for resource type JVM in execution group default - On?:  
inactive, measurements: [CommittedMemoryInMB, ...  
BIP8940I: Statistics settings for resource type Sockets in execution group default -  
On?: inactive, measurements: [AverageBytesReceivedPerMessage, ...
```
- Each command applies to all execution groups if none is specified
- See Information Center for command syntax, or type the command with no parameters

This slide shows how you can enable or disable resource statistics using the command interface. You can also display the current status of the resource statistics. If no execution group is specified, then each command will apply to all execution groups in the specified broker. For this reason, this command should be used carefully, since the activation of resource statistics for an entire broker can result in a performance overhead.

Subscribing to receive statistics

- Resource statistics are published to a well-defined topic
- An application can subscribe to a topic string to view the required statistics
 - `$/SYS/Broker/MB7BROKER/ResourceStatistics/default/#` for statistics related to execution group 'default' on broker 'MB7BROKER'
 - `$/SYS/Broker/+/ResourceStatistics/#` for all execution groups on all brokers
- One option is to set up a subscription to a queue in MQ Explorer



7

© 2010 IBM Corporation

The Message Broker Explorer subscribes to well-defined topic in the broker, on which the statistics data is published. Any other application can also subscribe to this queue. You can specify a range of topic strings to subscribe to, depending on whether you need information for all brokers, all execution groups, or a subset of these. You can manually create the required subscription using the MQ Explorer to specify this. Do this by selecting “Subscriptions”, right-clicking, and selecting new, subscription.

If you want to test this subscription, you can select topics, and select “Test Subscription” from the pop-up menu. In the subscription window, specify the topic string that you want to subscribe to, and click “Subscribe”. Approximately every 20 seconds, the broker will publish the current resource statistics, and they will be shown in the Test Subscription window.

Example publication

```
<ResourceType name="Sockets">
  <resourceIdentifier name="summary" TotalMessages="5"
    TotalSocketsOpened="2" AverageSocketsOpenedPerMinute="6"
    TotalBytesSent="24484" AverageBytesSentPerSecond="1223"/>
  <resourceIdentifier name="localhost.7080" TotalMessages="3"
    TotalSocketsOpened="1" AverageSocketsOpenedPerMinute="3"
    TotalBytesSent="12891" AverageBytesSentPerSecond="644"/>
  ...
</ResourceType>
```

This is an example of the type of data published for a specific resource. This example shows sockets statistics, and shows a subset of the data that is published.

Each resource manager publishes a 'summary' record which collates the information for the whole execution group

Some resource managers also publish finer grained information for individual resources that they manage. For example, in the sockets statistics, information is published about sockets that are managed for each endpoint that the broker is sending messages to. This information is then combined into the summary record.

Sockets statistics data - 1

- **TotalMessages**
 - The number of messages sent out over keepalive sockets by SOAP, SCA and HTTP nodes during the last 20 second reporting period
- **AverageMessagesPerMinute**
 - The average number of messages sent per minute
- **TotalSocketsOpened**
 - The number of sockets opened for outgoing requests by SOAP, SCA and HTTP nodes during the last 20 second reporting period.
- **AverageSocketsOpenedPerMinute**
 - The average number of sockets opened per minute
- **TotalBytesSent/TotalBytesReceived**
 - The number of bytes sent and received over keepalive sockets during the last 20 second reporting period
- **AverageBytesSentPerSecond/AverageBytesReceivedPerSecond**
 - The average number of bytes sent and received per second
- **AverageBytesSentPerMessage/AverageBytesReceivedPerMessage**
 - The average number of bytes sent and received per message

This slide shows the information that is collected and published for the number of socket requests within an execution group. Data is published every 20 seconds, and several values are calculated over a period of time. For example, the MB Explorer calculates the number of messages sent per minute.

The HTTP Request nodes do not use “keepalive” sockets, unless the property “Enable HTTP keep-alive” is set. The published statistics assume that if a socket has been requested, then a message will be sent on this socket. The published data uses the term “message”, rather than “socket request”.

A reply to an asynchronous request is counted as a separate outbound message. However, this will only appear in the published statistics for the execution group which hosts the reply node.

For the “TotalSocketsOpened” field, this represents the number of sockets opened for outgoing requests by SOAP, SCA and HTTP nodes during the last 20 second reporting period. This is not the total number of sockets open. Sockets opened earlier will still be open, whilst others will be closed if they are not in use.

The creation of sockets is an expensive operation, and the number of sockets available is a finite resource. If the workload is continuous and smooth then you should expect to see **TotalSocketsOpened** reflect an initial period of activity, but then fall as sockets start to be reused. If **TotalSocketsOpened** is fairly consistent over time then this might just indicate that the workload is not continuous. There is normally some socket creation going on, because sockets are closed after a period of inactivity or when they have been used many times. If **TotalSocketsOpened** increases over time then this can indicate an increasing workload, but more likely that sockets are not being reused. If you are using the HTTP Request node, check that keepalive is enabled. Check that the endpoint being invoked is configured to use keepAlives.

Sockets statistics data - 2

- The report also includes the number of messages sent and received during the last 20 second reporting period in each of the ranges listed here
- Example
 - Two request messages sent, each 1234 bytes
 - Response 1: 50000 bytes
 - Report will show:
 - SentMessageSize_1KB-10KB=2
 - ReceivedMessageSize_10KB-100KB=1
 - ReceivedMessageSize_100KB-1MB=1

Field name for sent messages	Field name for received messages	Message size range in bytes
SentMessageSize_0-1KB	ReceivedMessageSize_0-1KB	0 to 999
SentMessageSize_1KB-10KB	ReceivedMessageSize_1KB-10KB	1000 to 9999
SentMessageSize_10KB-100KB	ReceivedMessageSize_10KB-100KB	10000 to 99999
SentMessageSize_100KB-1MB	ReceivedMessageSize_100KB-1MB	100000 to 999999
SentMessageSize_1MB-10MB	ReceivedMessageSize_1MB-10MB	1000000 to 9999999
SentMessageSize_Over10MB	ReceivedMessageSize_Over10MB	10000000 or greater

The sockets statistics also provides information on the number and size of messages sent. It also provides a breakdown of sent and received messages. Messages are broken down into size ranges, as shown on this slide.

The **AverageBytesSentPerSecond** and **AverageBytesReceivedPerSecond** give you an average of the data being sent or received. Note that this measurement will be skewed if you have infrequent messages that are significantly larger or smaller than the norm. In this cases the metrics **SentMessageSize*** and **ReceivedMessageSize*** give a better profile of message sizes flowing from and to each endpoint.

JVM statistics data – memory

A JVM has two memory areas, Heap and Non Heap, these properties are reported for each:

- **InitialMemoryInMB**
 - The initial amount of memory (in megabytes) that the Java™ virtual machine requests from the operating system for memory management during the start process
 - Its value may be undefined which is indicated by a value of -1
- **UsedMemoryInMB**
 - The amount of memory currently used (in megabytes)
- **CommittedMemoryInMB**
 - The amount of memory (in megabytes) that is guaranteed to be available for use by the Java virtual machine
- **MaxMemoryInMB**
 - The maximum amount of memory (in megabytes) that can be used for memory management
 - Its value may be undefined which is indicated by a value of -1
- The value -1 is returned for measurements that are undefined or have not been set

It can be useful to know how much memory is being used by the JVM of an execution group. Operating system tools can give you the total memory used by the “data-flow-engine” process, but do not show you how that memory is divided between Java and non-Java processing in the execution group.

The JVM memory published data has several fields which break down this usage.

CommittedMemoryInMB tells you how much memory is currently allocated to the JVM and **MaxMemoryInMB** tells you how big this can grow.

JVM statistics data – garbage collection

A JVM can have one or more garbage collectors to report on. For example one collector might do minor collections which are frequent but take little time and another collector might do major collections which are not as frequent but take considerably more time. The following properties will be reported for each

- resourceIdentifier name
 - The name of the garbage collector that is being reported on
- CumulativeNumberOfGCCollections
 - The total number of garbage collections that have occurred for this instance of the JVM. Its value can be undefined which is indicated by a value of -1
- CumulativeGCTimeInSeconds
 - The accumulated garbage collection elapsed time in seconds for this instance of the JVM.
 - Its value can be undefined which is indicated by a value of -1

The JVM periodically performs a garbage collection. If

CumulativeNumberOfGCCollections is increasing frequently, then the garbage collection might be excessive. A certain level of garbage collection is expected and required, but excessive garbage collection can impact performance. If

CumulativeGCTimeInSeconds is increasing at more than 2 seconds per 20 second interval, then you should consider increasing the “JVM Max Heap size” for your execution group. Also, inspect any Java Plug-in nodes or Java compute nodes in your message flows to ensure that you are not creating and deleting many objects that can be reused.

If the **CumulativeGCTimeInSeconds** is increasing by more than 2 seconds per interval, then consider to increasing the maximum heap size” to reduce this. If your **UsedMemoryInMB** is never close to your **InitialMemoryInMB** then this indicates that you are pre-allocating more memory for the heap than necessary, and you could reduce the JVM minimum heap size value for you execution group to a value closer to the **UsedMemoryInMB** value. Changes to these values this should be done gradually to find the optimum settings.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_Administration_Statistics.ppt

This module is also available in PDF format at: ..\\WMB7_Administration_Statistics.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere 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 other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.