

WMB/IIB Must gather: problems with an Execution Group on UNIX or Linux

Problem(Abstract)

You are experiencing an Execution Group problem with WebSphere Message Broker (WMB) or IBM Integration Bus (IIB) on UNIXes or Linux.

This mustgather document explains how to troubleshoot this problem by yourself, how to generate diagnostic data when the problem is happening, how to collect the available data from your system, and how to share that data with IBM support.

Environment

These instructions are for WebSphere Message Broker or IBM Integration Bus on AIX, HP-UX, Linux, and Solaris systems.

Return to the [WebSphere Message Broker MustGather index](#) for other systems.

Resolving the problem

Step 1: Self-guided Troubleshooting

1. Consider [Using the Test Client in trace and debug mode](#).
2. A [user level trace](#) is intended to assist users with understanding how their ESQL is processed. It will also show a high level view of the message flow behavior.
3. Check for known issues:
[Troubleshooting IIB/WMB](#)
[Try a dynamic guide to help with troubleshooting Message Broker](#)
[Recovering after the broker fails in IIB/WMB](#)
[IBM and WMB Frequently Asked Questions](#)
[Recovering after an execution group fails in IIB/WMB](#)
[WebSphere Message Broker support site](#)

Step 2: Collect Data

1. **mqsidc**
See: [mqsidc usage instructions](#)

Collecting data in console mode with IBM Support Assistant Data Collector

You can use the IBM® Support Assistant Data Collector in console mode to collect diagnostic documents for submission to IBM.

Before you start:

Before contacting IBM Software Support, ensure that your company has an active IBM software subscription and support contract, and that you are authorized to submit problems to IBM. See [Contacting IBM Software Support](#) for more details.

To run the IBM Support Assistant Data Collector and collect diagnostic documents, complete the following steps:

1. Ensure that the WebSphere® Message Broker run time variables are set correctly for your environment by using the **mqsiprofile** command. See [Environment variables after installation](#) for more information.

2. At a command prompt, enter the **mqsidc** command.

Note: You can ensure that the script is executable by entering the following command to change the file permissions: **chmod 755 mqsidc**

The IBM Support Assistant Data Collector starts in console mode.

3. Start the data collection. Available options are presented as numbered lists.
 - a. Type a file name for saving the collected data, or press Enter to generate a unique file name.
 - b. At the input field prompts, type the number of the required option and press Enter.
4. Choose a data transfer method.
 - a. Type the number of the required option for transferring the diagnostic documents to IBM and press Enter. The available options are:
 - i. Send the documents to IBM Software Support using secure transfer (HTTPS). You require a problem management record (PMR) number obtained through IBM Software Support.
 - ii. FTP the documents to IBM Software Support (unencrypted). You require a PMR number obtained through IBM Software Support. This option is less secure than option i.
 - iii. FTP the documents to another location (unencrypted). You are required to provide a target FTP location and appropriate authentication to access the documents.
 - iv. End the collection without sending.

If you choose any of the first three options above, you are required to enter additional information to complete the upload.

- b. Type the number of the required option to confirm the data collection.

When the IBM Support Assistant Data Collector completes, a .zip file is created at the location specified during the collection. You can extract the compressed files and examine the collected data by using a suitable tool.

- Always gather broker-specific data when possible using the **Broker Collector**
-
2. **General Broker information**
- Record the WebSphere Message Broker version, release, and maintenance level. This can be captured using the command 'mqsiservice -v'..
 - Record the operating system version, release, and maintenance level.
 - Record the version, release, and maintenance level of any related products and components for the problematic application.
 - Collect the local error log. On UNIX and Linux systems, the local error log is the syslog. The location of your syslog is configured in the syslog daemon.
See: [Configuring the syslog daemon](#)

Linux and UNIX systems: Configuring the syslog daemon

On Linux and UNIX systems, all WebSphere® Message Broker messages (other than messages that are generated by the command-line utilities) are sent to the syslog subsystem.

UNIX You must configure this subsystem so that all diagnostic messages that enable you to monitor the performance and behavior of your broker environment are displayed.

The configuration steps you make to ensure that all relevant messages are displayed depend on the version of Linux and UNIX that you are using. Refer to your operating system documentation relating to syslog (or syslog-ng for some versions of Linux) for information about how to configure the syslog subsystem.

WebSphere Message Broker processes call the syslog commands on the operating system but only those messages that correspond to the filter defined for the output destination are displayed. WebSphere Message Broker messages have:

- A facility of `user`.
- A level of `err`, `warn`, or `info`, depending on the severity of the situation causing the message to be issued.

To record all WebSphere Message Broker messages, create a filter on the `user` facility for messages of level `info` or greater. It is good practice to write these messages to a separate file; there might be a high number of them and they are more likely to be of interest to broker administrators rather than to system administrators.

The following line in a `syslog.conf` file causes all WebSphere Message Broker events to be written to a file `/var/log/user.log`

```
user.info /var/log/user.log
```

Many UNIX systems provide a command-line utility, known as logger, to help you test and refine your configuration of the syslog subsystem.

On UNIX, syslog entries are restricted in length and messages that are sent to the syslog are truncated by the new line character. To record a large amount of data in a log on UNIX, set the *Destination* property on theTrace node to File or User Trace instead of Local Error Log.

- Collect the Standard Output/Error logs.
WMB writes information to both STDOUT and STDERR. These files are located under the Message Broker workpath.
See: [Standard System Logs](#)

3. Select the point of failure:

- ↕ [Failures](#)
- ↕ [Abends](#)
- ↕ [Startup](#)
- ↕ [Creation](#)

Failures

- Execution Group [service trace](#)

Abends

- Abend file
Abend files are recorded in the syslog, however if there is more data to be written than is appropriate for the log, a new file is created, and the log entry tells you the file name. The new file is called: *filename.abend* and is written to the `/var/mqsi/common/errors` directory.
- Execution Group [service trace](#)

Startup

- [Startup trace](#)

New Trace Procedure for WMB V7.0.0.3 and WMB V8.0.0.0 and later

1. Stop the broker.

mqsisstop <BrokerName>

Where <BrokerName> is the name of your broker.

2. Enable broker agent trace.

mqsischangetrace <BrokerName> -t -b -l debug -c <file size in KB>

3. Enable the Execution Group startup trace.

mqsischangetrace <BrokerName> -t -e <EGName> -l debug -c <file size in KB>

Where <BrokerName> is the name of your broker.

This command can be run even when broker / EG are in stopped state.

Note: If you are tracing an Execution Group startup abend problem, then use the following commands:

mqsischangetrace <BrokerName> -t -e <EGName> -l debug -c <file size in KB> -m temp

4. Start the broker to recreate the problem.

5. Turn off trace

```
mqsisichangetrace <BrokerName> -t -b -l none  
mqsisichangetrace <BrokerName> -t -e <EGName> -l none
```

6. Retrieve the trace log for specified component.

```
mqsireadlog <BrokerName> -t -b agent -f -o agent.xml  
mqsireadlog <BrokerName> -t -e <EGName> -f -o <EGName>.xml
```

7. Format the XML log file

```
mqsiformatlog -i agent.xml -o agent.txt  
mqsiformatlog -i <EGName>.xml -o <EGName>.txt
```

8. After a PMR is open, submit the files agent.txt and <EGName>.txt to IBM Support.

Creation

- Exact command and error message
- [Utility trace](#)

WMB Broker creation trace

Technote (troubleshooting)

Problem(Abstract)

Use this procedure in WebSphere Message Broker (WMB) if you are having problems creating your broker using the `mqsicreatebroker` command.

Gathering this information before calling IBM Support will help familiarize you with the troubleshooting process and save you time.

Resolving the problem

To capture a broker creation trace, perform the following steps:

1. Set the following environment variable. Ensure you reset the environment variable when the command you are tracing has completed.

On Windows:

```
set MQSI_UTILITY_TRACE=debug
```

// Depending on whether a service trace or user trace is required

On UNIX platforms:

```
export MQSI_UTILITY_TRACE=debug
```

Optionally, you can set the `MQSI_UTILITY_TRACESIZE` environment variable to increase the trace file size up to 2GB.

2. Run the `mqsicreatebroker` command until you get the failure.

3. Retrieve the trace log for the specified component.

```
mqsireadlog <brokername> -t -b mqsicreatebroker -f -o broker.xml
```

where **<brokername>** is the name of broker

4. Format the XML trace file.

```
mqsiformatlog -i broker.xml -o broker.log
```

5. Submit the file **broker.log** to IBM Support for problem determination.

6. **Unset** the environment variable - **MQSI_UTILITY_TRACE=debug**

ATTENTION: A good problem description is one of the most important tools IBM needs to analyze your data!

When sending data to IBM, be sure to update your PMR or send a note with the following information:

- Tell us what errors you saw, where you saw them, and what time they happened
- Let us know if you made any changes to WebSphere Message Broker or the system before the problem
- Share any other observations which you think will help us to better understand the problem

Step 3: Submit Data to IBM

- Use [IBM Service Request](#) to open or view a problem record with IBM.
- [Send your data to IBM](#) for further analysis.

1. Consider [Using the Test Client in trace and debug mode](#).

Using the Test Client in trace and debug mode

You can run the Test Client in trace and debug mode to trace the path of a test message through a message flow.

Before you start:

Before you can test your message flow, you must have configured a broker. The broker must be running. You must also have created an execution group to which to deploy your message flows.

You can use the trace and debug mode to complete the following actions:

- Stop the test message at breakpoints in the message flow by using the flow debugger.

- Trace the message nodes and terminals that the test message passes through.
- See the test message change as it passes through the message flow.
- View a message node, where an exception occurs, and the associated exception message and trace details.

Note: If the message flow does not contain an MQOutput, HTTPReply, JMSOutput, or SOAPReply node, the Test Client stops the test after the test message is sent to the broker. Therefore, you might not see trace events when the flow is run.

All output from the trace and debug mode is written to the "Message flow test events" section on the **Message flow Test Events** tab.

To use the Test Client in trace and debug mode, complete the following steps:

1. Configure the flow debug port for the execution group in the WebSphere® Message Broker Toolkit:
 - a. In the Brokers view, right-click the execution group with which you want to work.
 - b. Click **Launch Debugger**.
 - c. To set the Flow Debug Port, click **Configure**, and enter a number for the port.
 - d. Click **OK**.

The debugger is now enabled on the selected execution group. Click **Terminate Debugger** to stop the debugger.
2. In the Test Client, click **Send Message** to open the Deployment Location wizard.
3. If your broker is not connected, click **Connect**. From the list in the wizard, select the execution group to which you want to deploy your message flow.
4. Click **Trace and debug**.
5. Optional: If you want the test message to stop at a breakpoint after the input node, click **Stop at the beginning of the flow during debug**.
6. Click **Next** and modify the test settings as required.
7. Click **Finish** to save the settings and deploy the message flow.

Next:

You can modify the deployment location settings from run mode to trace and debug mode by using the Deployment Location wizard:

1. Click **Configuration** in the Test Client.
2. Click **Deployment** to display the deployment settings.
3. Click **Change** to open the Deployment Location wizard.

1. A [user level trace](#) is intended to assist users with understanding how their ESQL is processed. It will also show a high level view of the message flow behavior.

Procedure to take a user or service trace of message flow at an execution group level

Problem(Abstract)

Use this procedure to capture a user or service level trace of a WebSphere Message Broker message flow at an execution group level.

Gathering this information before calling IBM Support will help familiarize you with the troubleshooting process and save you time.

Resolving the problem

✚ [Service Level Trace](#) : Service trace provides more detailed information than that provided by the entries that are written to the Syslogs, Event Logs, Trace Node Outputs, or User Trace. Typically, IBM support uses service trace for debugging problems as it can trace internal calls in addition to the brokers, execution groups, and deployed message flows.

✚ [User Level Trace](#) : User trace provides more information than that provided by the entries that are written to the logs. Typically, you use user trace for debugging your applications, as it can trace brokers, execution groups, and deployed message flows.

Service Level Trace

1. Start trace.

```
mqsichangetrace <brokername> -t -e <egroup> -l debug -r -c 100000
```

2. Put a message on the queue to cause the failure to occur.

3. Stop trace.

```
mqsichangetrace <brokername> -t -e <egroup> -l none
```

4. Retrieve the trace log for the specified component.

```
mqsireadlog <brokername> -t -e <egroup> -f -o flowtrace.xml
```

5. Format the XML trace file.

```
mqsiformatlog -i flowtrace.xml -o serviceflowtrace.txt
```

The serviceflowtrace.txt will be in the current working directory.

User Level Trace

1. Start trace .

```
mqsichangetrace <brokername> -u -e <egroup> -f "<Message Flow Name>" -l debug -r -c 50000
```

<brokername> is the name of your broker

<egroup> is the name of your execution group

You can also start tracing, by going to **Operations** tab, and right click **Message Flow**, and click **Start Trace**

2. Put a message on the input node queue to cause the failure to occur.

3. Stop trace.

```
mqsichangetrace <brokername> -u -e <egroup> -f "<Message Flow Name>" -l none
```

You can also stop tracing, by going to **Operations** tab, and right click **Message Flow**, and click **Stop Trace**

4. Retrieve the trace log for the specified component.

```
mqsireadlog <brokername> -u -e <egroup> -f -o flowtrace.xml
```

5. Format XML tracefile.

```
mqsiformatlog -i flowtrace.xml -o userflowtrace.txt
```

The userflowtrace.txt will be in the current working directory.

Logs

If an error is reported by a WebSphere® Message Broker component, start your investigations into its causes by looking at the product and systems logs to which information is written during component operation.

The information that is recorded in a log typically consists of a time stamp, which indicates when the error occurred, and a brief description of the error, which can be expanded to provide more details of the error. In some situations, a general error is written and is followed by a group of more specific errors that provide details about the general error.

When an error occurs, check STDOUT, STDERR, and the local error log first. These logs record information about major activities within the system. All components of WebSphere Message Broker provide diagnostic information whenever error or warning conditions affect broker operation. These conditions include:

- Unsuccessful attempts to write a message to a WebSphere MQ output queue
- Errors interacting with databases
- The inability to parse an input message

Additional logs that are specific to WebSphere Message Broker are written to record runtime errors, internal errors that are produced by the operating system or your code, or errors related to the work that you are doing in a particular perspective, all of which you can view using the WebSphere Message Broker Toolkit.

Data is written to the following logs:

- [Standard system logs](#)
- [Local error logs](#)
- [WebSphere Message Broker logs](#)
- [WebSphere MQ logs](#)
- [Database logs](#)

Trace

If you cannot get enough information about a particular problem from the entries that are available in the various logs, the next troubleshooting method to consider is using trace. Trace provides more details about what is happening while code runs. The information produced from trace is sent to a specified trace record, so that you or IBM® support personnel can analyze it to discover the cause of your problem.

Trace is inactive by default, and must be explicitly activated by a command, or by the IBM Integration Toolkit.

There are two main types of trace available in IBM Integration Bus: user trace and service trace. Typically, you utilize user trace for debugging your applications; you can trace brokers, integration servers, deployed subflows, and deployed message flows. With service trace, you can activate more comprehensive broker tracing, and start tracing for the IBM

Integration Toolkit. You can also trace the execution of all the commands described in [Commands](#).

When you start user tracing, you cause additional processing for every activity in the component that you are tracing. Large quantities of data are generated by the components. Expect performance to be affected while trace is active. You can limit this additional processing by being selective about what you trace, and by restricting the time during which trace is active.

Commands

All WebSphere® Message Broker Toolkit and runtime commands that are provided on distributed systems are listed, grouped by function, with references to command details.

For information about the equivalent commands on z/OS®, see [Summary of commands on Linux, UNIX, Windows, and z/OS systems](#).

WebSphere Message Broker Toolkit commands

Command name	Topic reference
mqsipplybaroverride	mqsipplybaroverride command
mqsicreatebar	mqsicreatebar command
mqsicreatemsgdefs	mqsicreatemsgdefs command
mqsicreatemsgdefsfromwsdl	mqsicreatemsgdefsfromwsdl command
mqsireadbar	mqsireadbar command

WebSphere Message Broker commands

Table 1. Broker commands

Command name	Topic reference
mqsiaaddbrokerinstance	mqsiaaddbrokerinstance command
mqsiaapplybaroverride	mqsiaapplybaroverride command
mqsibackupbroker	mqsibackupbroker command
mqsichangebroker	mqsichangebroker command
mqsicreatebroker	mqsicreatebroker command
mqsicreateexecutiongroup	mqsicreateexecutiongroup command
mqsideletebroker	mqsideletebroker command
mqsideleteexecutiongroup	mqsideleteexecutiongroup command
mqsimode	mqsimode command
mqsireadbar	mqsireadbar command
mqsireload	mqsireload command
mqsiremovebrokerinstance	mqsiremovebrokerinstance command
mqsireportbroker	mqsireportbroker command

Table 1. Broker commands

Command name	Topic reference
mqsirestorebroker	mqsirestorebroker command

Table 2. Database commands

Command name	Topic reference
mqsimanagexalinks	mqsimanagexalinks command
mqsisetdbparms	mqsisetdbparms command

Table 3. Security commands

Command name	Topic reference
mqsireloadsecurity	mqsireloadsecurity command
mqsisetdbparms	mqsisetdbparms command
mqsisetsecurity	mqsisetsecurity command
mqsiwebuseradmin	mqsiwebuseradmin command

Table 4. Start and stop commands

Command name	Topic reference
--------------	-----------------

Table 3. Security commands

Command name	Topic reference
mqsisstart	mqsisstart command
mqsisstartmsgflow	mqsisstartmsgflow command
mqsisstop	mqsisstop command
mqsisstopmsgflow	mqsisstopmsgflow command

Table 5. List and trace commands

Command name	Topic reference
mqsischangetrace	mqsischangetrace command
mqsisformatlog	mqsisformatlog command
mqsilist	mqsilist command
mqsireadlog	mqsireadlog command
mqsireporttrace	mqsireporttrace command

Table 6. Migration commands

Command name	Topic reference
--------------	-----------------

Table 5. List and trace commands

Command name	Topic reference
mqsigratecomponents	mqsigratecomponents command

Table 7. Properties commands

Command name	Topic reference
mqsichangeproperties	mqsichangeproperties command
mqsireportproperties	mqsireportproperties command

Table 8. Monitoring commands

Command name	Topic reference
mqsichangeflowmonitoring	mqsichangeflowmonitoring command
mqsireportflowmonitoring	mqsireportflowmonitoring command

Table 9. Statistics commands

Command name	Topic reference
mqsichangeflowstats	mqsichangeflowstats command
mqsichangeresourcestats	mqsichangeresourcestats command

Table 9. Statistics commands

Command name	Topic reference
mqsiereportflowstats	mqsiereportflowstats command
mqsiereportresourcestats	mqsiereportresourcestats command

Table 10. Miscellaneous commands

Command name	Topic reference
mqsiAssemblyInstall	mqsiAssemblyInstall command
mqsicacheadmin	mqsicacheadmin command
mqsichangeflowuserexits	mqsichangeflowuserexits command
mqsicommandconsole	mqsicommandconsole command
mqsicreateconfigurableservice	mqsicreateconfigurableservice command
mqsicvp	mqsicvp command
mqsideleteconfigurableservice	mqsideleteconfigurableservice command
mqsideploy	mqsideploy command
mqsiexplain	mqsiexplain command

Table 9. Statistics commands

Command name	Topic reference
mqsipackagebar	mqsipackagebar command
mqsireportflowuserexits	mqsireportflowuserexits command

Summary of commands on Linux, UNIX, Windows, and z/OS systems

The following table summarizes the runtime commands that are available on Linux, UNIX, and Windows systems, and provides the z/OS equivalent, where it is available.

Command on Windows platforms, Linux, and UNIX systems	z/OS equivalent: type	z/OS equivalent	z/OS References
mqsipapplybaroverride	Utility JCL	BIPOBAR	Contents of the broker PDSE
mqsibackupbroker	Utility JCL	BIPBUBK	Contents of the broker PDSE
mqsicacheadmin	Utility JCL	BIPCCAD	Contents of the broker PDSE
mqsichangebroker	1. Console command: modify	1. changebroker 2. BIPCHBK	1. mqsichangebroker command 2. Contents of the broker PDSE

Command on Windowsplatforms, Linux, andUNIX systems	z/OSequivalent: type	z/OS equivalent	z/OS References
	2. Utility JCL		
mqsichangeflowmonitoring	1. Console command: modify 2. Utility JCL	1. changeflowmonitoring 2. BIPCHME	1. mqsichangeflowmonitoring command 2. Contents of the broker PDSE
mqsichangeflowstats	1. Console command: modify 2. Utility JCL	1. changeflowstats 2. BIPCHMS	1. mqsichangeflowstats command 2. Contents of the broker PDSE
mqsichangeflowuserexits	1. Console command: modify 2. Utility JCL	1. changeflowuserexits 2. BIPCHUE	1. mqsichangeflowuserexits command 2. Contents of the broker PDSE
mqsichangeproperties	Utility JCL	BIPCHPR	Contents of the broker PDSE
mqsichangeresources	Utility JCL	BIPCHRS	mqsichangeresources command

Command on Windowsplatforms, Linux, andUNIX systems	z/OSequivalent: type	z/OS equivalent	z/OS References
mqsichange trace	Console command: modify	changetrace	mqsichange trace command
mqsi createbroker	Utility JCL	BIPCRBK	Contents of the broker PDSE
mqsi createconfigurable service	Utility JCL	BIPJADPR	Contents of the broker PDSE
mqsi createexecutiongroup	Utility JCL	BIPCREG	Contents of the broker PDSE
mqsi cvp	Not applicable		
mqsi deletebroker	Utility JCL	BIPDLBK	Contents of the broker PDSE
mqsi deleteconfigurable service	Utility JCL	BIPJADPR	Contents of the broker PDSE
mqsi deleteexecutiongroup	Utility JCL	BIPDLEG	Contents of the broker PDSE
mqsi deploy	<ol style="list-style-type: none"> 1. Console command: modify 2. Utility JCL 	<ol style="list-style-type: none"> 1. deploy 2. BIPDPLY 	<ol style="list-style-type: none"> 1. mqsideploy command 2. Contents of the broker PDSE
mqsi explain	Utility JCL	BIPEXPL	Contents of the broker PDSE

Command on Windowsplatforms, Linux, andUNIX systems	z/OSequivalent: type	z/OS equivalent	z/OS References
mqsiformatlog	Utility JCL	BIPFMLG	Contents of the broker PDSE
mqsilist	<ol style="list-style-type: none"> 1. Console command: modify 2. Utility JCL 	<ol style="list-style-type: none"> 1. list 2. BIPLIST 	<ol style="list-style-type: none"> 1. mqsilist command 2. Contents of the broker PDSE
mqsimanagexalinks	Not applicable		
mqsimigratecomponents	Utility JCL	BIPMGCMP	mqsimigratecomponents command
mqsimode	Utility JCL	BIPMODE	Contents of the broker PDSE
mqsipackagebar	Utility JCL	BIPPACK	Contents of the broker PDSE
mqsireadbar	Utility JCL	BIPRBAR	Contents of the broker PDSE
mqsireadlog	Utility JCL	BIPRELG	Contents of the broker PDSE
mqsireload	Console command: modify	reload	mqsireload command
mqsireloadsecurity	<ol style="list-style-type: none"> 1. Console command: 	<ol style="list-style-type: none"> 1. reloadsecurity 2. BIPRLSEC 	<ol style="list-style-type: none"> 1. mqsireloadsecurity command 2. Contents of the broker PDSE

Command on Windowsplatforms, Linux, andUNIX systems	z/OSequivalent: type	z/OS equivalent	z/OS References
	modify 2. Utility JCL		
mqsiereportbroker	Utility JCL	BIPRPBK	Contents of the broker PDSE
mqsiereportflowmonitoring	1. Console command: modify 2. Utility JCL	1. reportflowmonitoring 2. BIPRPME	1. mqsiereportflow monitoring command 2. Contents of the broker PDSE
mqsiereportflowstats	1. Console command: modify 2. Utility JCL	1. reportflowstats 2. BIPRPMS	1. mqsiereportflow stats command 2. Contents of the broker PDSE
mqsiereportflowuserexits	1. Console command: modify 2. Utility JCL	1. reportflowuserexits 2. BIPRPUE	1. mqsiereportflow userexits command 2. Contents of the broker PDSE

Command on Windowsplatforms, Linux, andUNIX systems	z/OSequivalent: type	z/OS equivalent	z/OS References
mqsireportproperties	Utility JCL	BIPRPPR	Contents of the broker PDSE
mqsireportresourcestats	Utility JCL	BIPRPRS	mqsireportresourcestats command
mqsiporttrace	Console command: modify	reporttrace	mqsiporttrace command
mqsiestorebroker	Utility JCL	BIPRSBK	Contents of the broker PDSE
mqsiestdbparms	Utility JCL	BIPSDBP	mqsiestdbparms command
mqsiestsecurity	Not applicable		
mqsi_setupdatabase	Not applicable		
mqsiestart	<ol style="list-style-type: none"> 1. Console command: start 2. Console command: modify 	<ol style="list-style-type: none"> 1. Standard MVS™ start command 2. startcomponent 	<ol style="list-style-type: none"> 1. - 2. mqsiestart command
mqsiestartmsgflow	Utility JCL	BIPSTMF	Contents of the broker PDSE
mqsiestop	<ol style="list-style-type: none"> 1. Console 	<ol style="list-style-type: none"> 1. Standard MVS stop command 	<ol style="list-style-type: none"> 1. -

Command on Windowsplatforms, Linux, andUNIX systems	z/OSequivalent: type	z/OS equivalent	z/OS References
	command: stop 2. Console command: modify	2. 'p' stopcomponent	2. mqsisstop command
mqsisstopmsgflow	Utility JCL	BIPSPMF	Contents of the broker PDSE
mqsiwebuseradmin	1. Console command: modify 2. Utility JCL	1. webuseradmin 2. BIPWUA	1. mqsiwebuseradmin command 2. Contents of the broker PDSE

Trace logging from a user-defined C extension

Message processing nodes and parsers that are written to the C programming language API can write entries to trace.

You can use two types of trace:

- **Service Trace:** entries usually describe what is happening within the code and are only useful to the owner of the code, such as the user-defined extension developer.
- **User Trace:** entries usually describe what is happening at an external level and are useful to the user of the code. Users of the code include message flow designers, and broker domain administrators.

For each trace type, there are three levels:

- None
- Normal
- Debug

For C user-defined extensions, the following utility functions are available for each trace type:

- **cciServiceTrace** and **cciUserTrace**: these functions write an entry to the respective trace type only when trace has been activated, that is, trace is at normal or debug level.
- **cciServiceDebugTrace** and **cciUserDebugTrace**: these functions write an entry to the respective trace type only when trace is active at debug level.

To help avoid making function calls in the case where no trace is written, the **cciIsTraceActive** utility function is provided. **cciIsTraceActive** reports whether trace is active and the level at which trace is active.

The **cci*Trace** functions can be used by a user-defined extension regardless of the trace settings. The functions determine if trace is active and only write entries which are appropriate for the trace settings. When calling the **cci*Trace** functions, some additional processing can be required. The **cciIsTraceActive** function is provided to allow the user-defined extension to query the trace settings and avoid this extra processing when trace is inactive.

In many cases, it is sufficient to treat the value returned from the **cciIsTraceActive** function as a Boolean value. If the returned value is non zero, trace is active at some level and it is appropriate to call any of the **cci*Trace** functions. The returned value can also be inspected closely in the cases when details of the trace settings are required.

Trace settings can be changed at any time so it is advisable to query them regularly. For example, use **cciIsTraceActive** to query the trace settings when an implementation function is entered.

Trace entries can be associated with certain objects, which allows for further refinement of control for writing trace. A trace entry can be associated with a node or parser and trace is written according to the trace setting for that object. The object's trace setting is inherited from the message flow to which the node or parser belongs. If no object is specified, the trace is associated with the integration server.

Dumps

Under exceptional circumstances, Windows MiniDumps, UNIX core dumps, or z/OS® SVC or core dumps might be produced.

For example, if errors occur in the broker executable files, or in the infrastructure main program. The amount and complexity of data

produced in these cases typically requires the assistance of your IBM® Support Center, see [Contacting your IBM Support Center](#).

Dumps on Windows

Windows MiniDumps might be produced by broker processes in extreme cases. Windows MiniDumps are never produced during typical operation.

Windows MiniDumps are accompanied by a BIP2111 error message (a message broker internal error) that contains the path to the MiniDump file in your `workpath/errors` directory. MiniDump files have the extension `.dmp`. You can analyze these MiniDumps by using a suitable debugger; contact IBM for assistance.

Dumps on UNIX systems

UNIX core dumps are produced by broker processes in extreme cases. UNIX core dumps are never produced during typical operation.

A BIP2060 error message (the integration server stopped unexpectedly) might be produced. Look in the directory where the broker was started to find the core dump file. If this directory is not writable by the service ID, the core dumps are produced in the service user ID's home directory.

You can analyze these core dumps by using a suitable debugger; contact IBM for assistance.

Dumps on IBM Integration Bus for z/OS

A broker produces an SVC dump that is written to a data set that is named by using the system defined naming convention.

The format of this data set name can be displayed by using the dump options command `/D dump`.

Typically, the name definition has the following format:

```
DUMP.&SYSNAME..&JOBNAME..D&DATE..T&LTIME..S&SEQ
```

and, for example, this format displays a resolved name of:

```
DUMP.MVS1.MQ83BRK.D080924.T171755.S00005
```

An SVC dump that is caused by z/OS or IBM Integration Bus for z/OS is written to the system's dump directory, and can be formatted with IPCS. The name of the dump is listed on the z/OS syslog.

The following example shows the format of the output that you receive:

```
IEA794I SVC DUMP HAS CAPTURED:
```

```

DUMPID=006 REQUESTED BY JOB (MQ83BRK )
DUMP TITLE=MQ83BRK ,ABN=S0C4-00000004,C=M7500.600.BRKR
IEA611I COMPLETE DUMP ON DUMP.MVS1.MQ83BRK.D080924.T171851.S00006
DUMPID=006 REQUESTED BY JOB (MQ83BRK )
FOR ASID (00B8)
INCIDENT TOKEN: PLEXS      MVS1      09/24/2008 16:19:11
ID = MQ83BRK ,ABN=S0C4-00000004

```

Under some circumstances, SVC dumps are not produced. Typically, dumps are suppressed because of time or space problems, or security violations.

In addition, you can suppress SVC dumps that duplicate previous dumps by using z/OS dump analysis and elimination (DAE), for example, use the command **SET DAE=xx**. (DAE is a z/OS service that enables an installation to suppress SVC dumps and ABEND SYSDUMP dumps that are not required because they duplicate previously written dumps.)

The following example shows a message on the z/OS syslog, indicating whether duplicates of SYSDUMPs are suppressed:

```

IEA995I SYMPTOM DUMP OUTPUT 504
SYSTEM COMPLETION CODE=0C4 REASON CODE=00000004
TIME=11.02.24 SEQ=02327 CPU=0000 ASID=0060
PSW AT TIME OF ERROR 078D2000 8D70A656 ILC 4 INTC 04
ACTIVE LOAD MODULE ADDRESS=0D708F68 OFFSET=000016EE
NAME=SPECIALNAME
        61939683 81936199 85A2F1F0 61A48689 */local/res10/ufi*
        A7F5F161 82899561 82899789 94818995 *x51/bin/bipimain*
DATA AT PSW 0D70A650 - 91345000 00001F22 41209140
GPR 0-3 00000000 0D711B18 0D711B18 00000000
GPR 4-7 0D711300 0D70BD04 00000000 0D711B18
GPR 8-11 0D7121CF 0D7111D0 0D70C316 0D70B317
GPR 12-15 8D70A318 0D7111D0 00000312 00000000
END OF SYMPTOM DUMP
IEA838I SYSDUMP SUPPRESSED AS A DUPLICATE OF: 505
ORIGINAL:DATE 01170 TIME 10:59:40:05 CPU 8A7645349672
MOD/#PATHNAM CSECT/BIPIP PIDS/5655G9700 AB/S00C4
REXN/IMBSIREC FI/9134500000001F2241209140 REGS/0C33E
HRC1/00000004 SUB1/INFRASTRUCTURE#MAIN

```

The *MVS™ Diagnosis: Tools and Service Aids* manual provides details about using z/OS dump analysis and elimination.

In extreme cases, you might instead receive a core dump, which is written to the started task user's directory. The maximum size of a core dump is defined through MAXCORESIZE in the BPXPRMxx parmlib member. The IBM supplied default is 4 MB. To ensure the completeness of a core dump of any IBM Integration Bus for z/OS address space, change

the value to 2 GB. The started task user's directory must then have at least this size.

To make use of these core dumps, copy them to a partitioned data set by using the **TSO/E OGET** command. Allocate the data set as a sequential data set with a logical record length (LRECL) of 4160 and a record format of FBS. Specify a primary allocation of at least 500 cylinders and a secondary allocation of at least 250 cylinders. The data set must be transferred in binary format. For example:

```
oget '/u/user_directory/coredump.pid' 'mvs_dataset_name.pid' bin
```

To ensure that all of the diagnostic information is collected in these extreme cases, specify the following dump options in SYS1.PARMLIB:

- Member IEADMP* SDATA=(LSQA,TRT,CB,ENQ,DM,IO,ERR,SUM)
- Member IEADMR* SDATA=(NUC,SQA,LSQA,SWA,TRT,RGN,LPA,CSA,SUM,GRSQ)

IBM Integration Bus for z/OS abends

Abend code

2C1

Explanation

The IBM Integration Bus for z/OS infrastructure encountered a severe internal error. The reason code helps the IBM Support Center to identify the source of the error. It has the format X'ppmmnnnn', where pp defines the part within the infrastructure where the error occurred, mm defines the specific location of the error, and nnnn defines a recursion index.

You can resolve the following reason codes but you must refer other codes to the IBM Support Center.

ABN=S2C1-0001xxxx Error opening a file system file

The file system component might be full or the broker user ID might not have the correct permissions to access files or directories in the file system component.

ABN=S2C1-0113xxxx Region size too small

Source

IBM Integration Bus for z/OS

System Action

The system might issue a dump.

Programmer Response

None

System Programmer Response

Search the problem-reporting databases for a fix for the problem. If no fix exists, contact the IBM Support Center, providing the reason code and, if present, the dump.

Abend files

When a process does not end normally an abend file is generated.

When a process does not end normally, an abend entry is made in the syslog, or the Windows Event log. If there is more data to be written than is appropriate for the log, a new file is created to contain it, and the log entry tells you the file name. You need to send the file to your IBM® Support Center for analysis.

Abend files are never generated during normal operation; always involve your IBM Support Center when they do occur.

The new file is called:

Windows

`filename.abend`; if you have set the workpath using the **-w** parameter of the **mqsicreatebroker** command, the location is `workpath\Common\errors`. If you have not specified the broker workpath, the workpath can be resolved by issuing the command **echo %MQSI_WORKPATH%** from the installation's Command Console. If it is not possible to write to this directory, the file is put in the `workpath\Common\log` directory, but in either case the message in the Event Log tells you where to find it.

UNIX

`filename.abend` in the `/var/mqsi/common/errors` directory.

z/OS®

`filename.abend` in the `/<component_HFS>/common/errors` directory. where *filename* is a unique dynamically-allocated name, as given in the syslog or Event Log message.

The abend file might provide a stack for the failing thread; this might help you to identify problems in the plug-in code when an abend occurs here. In which case, for z/OS, this stack will be available in a CEEDUMP file in the same directory as the abend file by default.

Each abend file contains a header that includes the following sections:

- Product details (the broker product that is being used)
- Operating System

- Environment (including the installation path and the process ID that contains the failing thread)
- Deployment (including the component and, where applicable, the integration server name and UUIDs)
- Build Information (for IBM internal use)
- Failure Location (including the time of failure and, where applicable, the message flow name)

The content and organization of the abend file and its header are subject to change without notice.

Periodically clear any unwanted abend files from the errors or z/OS log directory. You can do this by moving the files to an archive, or by deleting them when they are no longer needed. This ensures that the workpath does not become full; in extreme conditions, your system performance can be degraded if significant space is being used up with old abend files.

IBM Integration Bus event reports

Event messages are published by a broker in response to certain conditions that occur while the broker is active.

Reports are raised in response to the following circumstances:

- [IBM Integration Bus event reports: configuration changes](#)
- [IBM Integration Bus event reports: operational information](#)

Further information about events is provided in the following topics:

- [IBM Integration Bus event reports: general architecture](#)
- [IBM Integration Bus event reports: notification message schema](#)

The events are published on a series of system-defined topics. The body of the message contains additional information in XML format. Every message is generated in code page 1208.

The following set of events can be reported:

Configuration changes

- An integration server has been created, changed, or deleted
- A message flow has been created, changed, or deleted

Operational information

- A broker has been started or stopped
- A message flow has been started or stopped

Changes to the local configuration of the broker

An event is published to a system topic when an entity is created, changed, or deleted.

Notification of changes to the broker's configuration (create, change, or delete entities) is provided by publishing events on the following system topic:

```
$SYS/Broker/broker_name/Configuration/ExecutionGroup/exec_grp_name
```

where:

broker_name

is the name of the broker issuing this message

exec_grp_name

is the name of the integration server for which the configuration has changed

One such event is published for each configuration request message that is received and processed by an integration server within the broker and can thus contain information that reflects complex configuration changes to multiple entities within the broker.

The body of each publication is the part of the configuration request that causes the event to be triggered. If an integration server is renamed, subsequent publications that report the state of that integration server use the new name.

These events are published non-persistently as non-retained publications.

Only create, change, and delete actions on the message flow are reported.

Configuration change

The following figure shows an example notification when a message flow is created. The number of attributes mentioned in the example can vary.

```
<Broker uuid="1234" label="Broker1" version="1">
  <ExecutionGroup uuid="2345">
    <Create>
      <MessageFlow uuid="3456" label="MessageFlow1">
        <!-- Create the Input and Output Nodes -->
        <ComIbmMqInputNode uuid="4567"
          queueName="InputQueue1" label="InputNode1" />
        <ComIbmMqOutputNode uuid="5678"
          queueName="OutputQueue1"
          label="OutputNode1"/>
        <ComIbmMqOutputNode uuid="6789"
          queueManagerName="QueueManager1"
```

```

        queueName="OutputQueue2"
            label="OutputNode2"/>
<!-- Create the filter -->
<ComIbmFilterNode uuid="7890"
    filterExpression="Company=IBM"
    label="FilterNode1"/>
<!-- Connect them together -->
<Connection sourceNode="4567"
    sourceTerminal="out"
    targetNode="7890" targetTerminal="in"/>
<Connection sourceNode="7890"
    sourceTerminal="true"
    targetNode="5678" targetTerminal="in"/>
<Connection sourceNode="7890"
    sourceTerminal="false"
    targetNode="6789" targetTerminal="in"/>
</MessageFlow>
</Create>
</ExecutionGroup>
</Broker>

```

IBM Integration Bus event reports: operational information

Changes to the processing state of a broker or an individual message flow publish events with predefined system topics.

The following system topics are used:

```
$SYS/Broker/broker_name/Status
```

```
$SYS/Broker/broker_name/Status/ExecutionGroup/exec_grp_name
```

where:

broker_name

is the name of the broker whose execution state has changed

exec_grp_name

is the name of the integration server that contains the message flow whose execution state has changed

The body of each publication is an XML message that gives additional information concerning the state change that caused the event to be triggered, specifically indicating whether the entity has been started or stopped.

For example, starting a message flow generates the following message:

```
<Broker uuid="1234" label="Broker1" version="1">
  <ExecutionGroup uuid="5678">
    <Start>
      <MessageFlow uuid="7812"/>
    </Start>
  </ExecutionGroup>
</Broker>
```

Stopping a broker generates the following message body:

```
<Broker uuid="1234" label="Broker1" version="1">
  <StatusChange state="Stopped"/>
</Broker>
```

Currently, the only states that are notified for both brokers and message flows are Started and Stopped.

These events are nonpersistent, retained publications.

IBM Integration Bus event reports: general architecture

Brokers publish messages on reserved topics after significant events within the broker. By subscribing to these topics, a client can be informed when these events occur.

For each topic, the type of event and message body are explained. The body of these messages is in XML format.

An event publication can contain more than one entry if the topic is the same (for example, if several message flows are created in the same operation).

The general form of the system topics on which events are published is:

```
$SYS/Broker/broker_name/event_type/...
```

where:

broker_name

is the name of the broker issuing or raising this event.

event_type

is the type of the event and is one of:

- Configuration

- Neighbor
- Subscription
- Topic
- Status
- Expiry

This specification of topics helps clients to filter events, based on the broker from which the event originated and the type of event. The clients register subscriptions for these topics to receive the reports.

For specific events, additional information is included in the topic to help filter on the specific object that raised the event. The inclusion of the string `Broker` at the second level of the topic hierarchy allows for future extension to additional subsystems that publish system management events through the broker.

IBM Integration Bus event reports: notification message schema

The structure of all valid notification messages.

This example describes the structure of the messages only. It does not define about how many elements are in the messages or the order in which they appear. The rules for the number of elements are:

- One broker element
- Other elements: zero, one, or more

There are no rules for the order of notification messages.

In this example, `<...>` denotes an XML element, and `???` indicates where individual class names are allowed:

```
<Broker identifier label>
. <ExecutionGroup identifier>
. . <Create>
. . . <MessageFlow message_flow_identifier message_flow_attributes>
. . . . <???Node node_identifier node_attributes>
. . . . <Connection connection_identifier>
. . <Change>
. . . <MessageFlow message_flow_identifier message_flow_attributes>
. . <Delete>
. . . <AllMessageFlows>
. . . <MessageFlow message_flow_identifier>
. . <Start>
. . . <AllMessageFlows>
. . . <MessageFlow message_flow_identifier>
```

```
. . <Stop>
. . . <AllMessageFlows>
. . . <MessageFlow message_flow_identifier>
```

WebSphere MQ facilities

IBM® Integration Bus components depend on WebSphere® MQ resources in many ways. You can therefore gain valuable information from the WebSphere MQ logs and events.

WebSphere MQ logs

The WebSphere MQ product logs can be useful in diagnosing errors that occur in your broker network. For example, if the IBM Integration Toolkit cannot communicate with a broker, the channels that connect them might be wrongly configured, or experiencing network problems.

On distributed systems, operational messages in a user-readable format (such as `queue manager started`), are written to the error logs in the errors subdirectory of the queue manager directory.

FFST files

First Failure Support Technology™ (FFST™) records are normally severe, unrecoverable errors, and indicate either a configuration problem with the system, or a WebSphere MQ internal error. FFST files are named `AMQnnnnn.mm.FDC`, where `nnnnn` is the ID of the process that is reporting the error, and `mm` is a sequence number.

On Windows, records are written to the `install_dir\errors` directory. Operational messages and FFST records are also written to the Event log.

On UNIX and Linux systems, records are written to the `/var/mqm/errors` directory. WebSphere MQ writes one line for each FFST containing the name of the FFST file to the syslog, but no operational messages.

WebSphere MQ events

WebSphere MQ provides information about errors, warnings, and other significant occurrences in queue managers in the form of instrumentation event messages.

You can activate event activity by using the MQSC or PCF interfaces in three areas:

- Queue manager events
- Performance events
- Channel events

When active, these event messages are sent to event queues that can be monitored or triggered. You might find it appropriate to activate WebSphere MQ events when you are investigating the performance, or unexpected behavior, of your broker network.

WebSphere MQ facilities

IBM® Integration Bus components depend on WebSphere® MQ resources in many ways. You can therefore gain valuable information from the WebSphere MQ logs and events.

WebSphere MQ logs

The WebSphere MQ product logs can be useful in diagnosing errors that occur in your broker network. For example, if the IBM Integration Toolkit cannot communicate with a broker, the channels that connect them might be wrongly configured, or experiencing network problems.

On distributed systems, operational messages in a user-readable format (such as `queue manager started`), are written to the error logs in the errors subdirectory of the queue manager directory.

FFST files

First Failure Support Technology™ (FFST™) records are normally severe, unrecoverable errors, and indicate either a configuration problem with the system, or a WebSphere MQ internal error. FFST files are named `AMQnnnnn.mm.FDC`, where `nnnnn` is the ID of the process that is reporting the error, and `mm` is a sequence number.

On Windows, records are written to the `install_dir\errors` directory. Operational messages and FFST records are also written to the Event log.

On UNIX and Linux systems, records are written to the `/var/mqm/errors` directory. WebSphere MQ writes one line for each FFST containing the name of the FFST file to the syslog, but no operational messages.

WebSphere MQ events

WebSphere MQ provides information about errors, warnings, and other significant occurrences in queue managers in the form of instrumentation event messages.

You can activate event activity by using the MQSC or PCF interfaces in three areas:

- Queue manager events
- Performance events
- Channel events

When active, these event messages are sent to event queues that can be monitored or triggered. You might find it appropriate to activate WebSphere MQ events when you are investigating the performance, or unexpected behavior, of your broker network.



ODBC trace

You can use various methods to trace for ODBC activity, depending on the operating system that you are using.

 **Windows** For Windows, use the **Tracing** tab of the ODBC function:

1. Click **Start > Settings > Control Panel > Administrative Tools**.
2. Double-click **Data Sources**.
3. Click the **Tracing** tab.
4. Click **Start Tracing Now**.
5. Click **OK**.

To stop ODBC tracing, on the **Tracing** tab, click **Stop Tracing Now**, then **OK**.

 **Linux**  **UNIX** For Linux and UNIX operating systems using IBM® Integration ODBC Database Extender drivers:

- To initiate trace for ODBC activity, edit the [ODBC] stanza in the `odbcinst.ini` file in the directory pointed to by your ODBC_SYSINI environment variable as follows:
 1. Change `Trace=no` to `Trace=yes`.
 2. Specify a path and file name for `TraceFile`
 3. Ensure that the `TraceFile` entry points to a file system that has enough space to receive the trace output

 **z/OS** For z/OS®, to initiate application trace for ODBC activity:

1. Edit the `BIPDSNAO` file in the component dataset and under the stanza entry [COMMON], change `APPLTRACE=0` to `APPLTRACE=1`

2. Remove the comment from the `COMPDIR` variable declaration and the `APPLTRC DD` from the steps `EGNOENV` and `EGENV`, in the IBM Integration Bus started task JCL.
3. Stop and restart the broker after you have made all the changes to the `BIPDSNAO` file and the started task JCL.
By default, the trace output file is written to `<component_HFS>/output/`, into a file called `db2appltrace..`. Each address space has a unique number, and the eight character integration server label appended to the end of `db2appltrace..`. This unique number, appended to the ODBC file, is the SE number in the integration server address space `JOBLOG`. If the eight character integration server label is not unique across multiple integration servers, look for the value of `SE` in the `JOBLOG` for which you want to view the ODBC trace, and find the file that specifies this value.

DB2® on IBM Integration Bus for z/OS

The following sample ODBC trace files show the layout of a trace file, together with some examples of successful and error returns. The general layout of each group in an ODBC file is that:

- Each line is preceded by a time stamp.
- The first line displays what the call does.
- The second line displays the return.
- The third line displays the result.

The first trace file shows a trace where a call fails because an object does not have the correct authority to perform an action:

```
[2008-09-24 15:49:20.544123] SQLAllocStmt( hDbc=2, phStmt=&1c7f9554 )
[2008-09-24 15:49:20.544156] SQLAllocStmt( phStmt=1 )
[2008-09-24 15:49:20.544163]      ---> SQL_SUCCESS

[2008-09-24 15:49:20.544179] SQLFreeStmt( hStmt=1, fOption=SQL_CLOSE )
[2008-09-24 15:49:20.544189] SQLFreeStmt( )
[2008-09-24 15:49:20.544194]      ---> SQL_SUCCESS

[2008-09-24 15:49:20.544205] SQLPrepare( hStmt=1 )
[2008-09-24 15:49:20.544212] ( pszSqlStr="SELECT TESTTABLE.ID FROM
WMQI77.TESTTABLE TESTTABLE", cbSqlStr=-3 )
[2008-09-24 15:49:20.587083] SQLPrepare( )
[2008-09-24 15:49:20.587101]      ---> SQL_ERROR

[2008-09-24 15:49:20.587157] SQLError( hEnv=0, hDbc=0, hStmt=1,
pszSqlState=&3902af28, pfNativeError=&3902af24, pszErrorMsg=&1b88b0b0,
cbErrorMsgMax=1024, pcbErrorMsg=&3902aefc )
[2008-09-24 15:49:20.587190] SQLError( pszSqlState="42501",
pfNativeError=-551,
pszErrorMsg="{DB2 FOR OS/390}{ODBC DRIVER}{DSN09015}
```

```

DSNT408I SQLCODE = -551, ERROR:  WMQI83 DOES NOT HAVE THE PRIVILEGE TO
PERFORM
        OPERATION SELECT ON OBJECT WMQI77.TESTTABLE
DSNT418I SQLSTATE   = 42501 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXOSC SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -100  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'FFFFFF9C' X'00000000' X'00000000' X'FFFFFFF'
        X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
        ERRLOC=1:13:2", pcbErrorMsg=623 )
[2008-09-24 15:49:20.587666]      ---> SQL_SUCCESS

[2008-09-24 15:49:20.587725] SQLError( hEnv=0, hDbc=0, hStmt=1,
pszSqlState=&3902af28, pfNativeError=&3902af24, pszErrorMsg=&1b88b0b0,
cbErrorMsgMax=1024, pcbErrorMsg=&3902aefc )
[2008-09-24 15:49:20.587752] SQLError( )
[2008-09-24 15:49:20.587757]      ---> SQL_NO_DATA_FOUND

[2008-09-24 15:49:20.588049] SQLFreeStmt( hStmt=1, fOption=SQL_DROP )
[2008-09-24 15:49:20.588075] SQLFreeStmt( )
[2008-09-24 15:49:20.588080]      ---> SQL_SUCCESS

[2008-09-24 15:49:20.593800] SQLTransact( hEnv=1, hDbc=0, fType=SQL_COMMIT
)
[2008-09-24 15:49:20.593887] SQLTransact( )
[2008-09-24 15:49:20.593893]      ---> SQL_SUCCESS

```

The second trace file shows the same trace file with the operation working:

```

[2008-09-24 16:00:25.287052] SQLAllocStmt( hDbc=1, phStmt=&1c7f8e54 )
[2008-09-24 16:00:25.287068] SQLAllocStmt( phStmt=1 )
[2008-09-24 16:00:25.287075]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.287088] SQLFreeStmt( hStmt=1, fOption=SQL_CLOSE )
[2008-09-24 16:00:25.287098] SQLFreeStmt( )
[2008-09-24 16:00:25.287104]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.287114] SQLPrepare( hStmt=1 )
[2008-09-24 16:00:25.287121] ( pszSqlStr="SELECT TESTTABLE.ID FROM
WMQI77.TESTTABLE TESTTABLE", cbSqlStr=-3 )
[2008-09-24 16:00:25.302484] SQLPrepare( )
[2008-09-24 16:00:25.302510]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302539] SQLFreeStmt( hStmt=1,
fOption=SQL_CLOSE )
[2008-09-24 16:00:25.302555] SQLFreeStmt( )
[2008-09-24 16:00:25.302560]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302573] SQLExecute( hStmt=1 )

```

```
[2008-09-24 16:00:25.302622] SQLExecute( )
[2008-09-24 16:00:25.302628]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302660] SQLNumResultCols( hStmt=1,
pcCol=&3902c7fa )
[2008-09-24 16:00:25.302672] SQLNumResultCols( pcCol=1 )
[2008-09-24 16:00:25.302679]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302697] SQLDescribeCol( hStmt=1, iCol=1,
pszColName=&3902cb10, cbColNameMax=200, pcbColName=&3902c804,
pfSQLType=&3902c802, pcbColDef=&3902c858, pibScale=&3902c800,
pfNullable=&3902c7fe )
[2008-09-24 16:00:25.302733] SQLDescribeCol( pszColName="ID",
pcbColName=2, pfSQLType=SQL_CHAR, pcbColDef=10, pibScale=0,
pfNullable=SQL_NULLABLE )
[2008-09-24 16:00:25.302819]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302826] SQLColAttribute( hStmt=1, iCol=1,
fDescType=SQL_DESC_OCTET_LENGTH, rgbDesc=NULL, cbDescMax=0,
pcbDesc=NULL, pfDesc=&3902c864 )
[2008-09-24 16:00:25.302850] SQLColAttribute( pfDesc=10 )
[2008-09-24 16:00:25.302857]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302866] SQLBindCol( hStmt=1, iCol=1,
fCType=SQL_C_CHAR, rgbValue=&1b48829c, cbValueMax=12,
pcbValue=&1b488298 )
[2008-09-24 16:00:25.302888] SQLBindCol( )
[2008-09-24 16:00:25.302894]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302901] SQLSetStmtAttr( hStmt=1,
fAttribute=SQL_ATTR_ROW_BIND_TYPE, pvParam=&10, iStrLen=0 )
[2008-09-24 16:00:25.302917] SQLSetStmtAttr( )
[2008-09-24 16:00:25.302922]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302928] SQLSetStmtAttr( hStmt=1,
fAttribute=Unknown value 9, pvParam=&20, iStrLen=0 )
[2008-09-24 16:00:25.302943] SQLSetStmtAttr( )
[2008-09-24 16:00:25.302949]      ---> SQL_SUCCESS

[2008-09-24 16:00:25.302956] SQLExtendedFetch( hStmt=1,
fFetchType=SQL_FETCH_NEXT, iRow=0, pcRow=&1c7f6894,
rgfRowStatus=&1bca17d0 )
[2008-09-24 16:00:25.317947] ( Row=1, iCol=1, fCType=SQL_C_CHAR,
rgbValue="TABLG      ", pcbValue=10 )
[2008-09-24 16:00:25.317980] ( Row=2, iCol=1, fCType=SQL_C_CHAR,
rgbValue="TABLF      ", pcbValue=10 )
```

```
[2008-09-24 16:00:25.318001] ( Row=3, iCol=1, fCType=SQL_C_CHAR, r
rgbValue="TABLE      ", pcbValue=10 )
[2008-09-24 16:00:25.318022] ( Row=4, iCol=1, fCType=SQL_C_CHAR,
rgbValue="TABLD      ", pcbValue=10 )
[2008-09-24 16:00:25.318044] ( Row=5, iCol=1, fCType=SQL_C_CHAR,
rgbValue="TABLC      ", pcbValue=10 )
[2008-09-24 16:00:25.318065] ( Row=6, iCol=1, fCType=SQL_C_CHAR,
rgbValue="TABLB      ", pcbValue=10 )
[2008-09-24 16:00:25.318087] ( Row=7, iCol=1, fCType=SQL_C_CHAR,
rgbValue="TABLA      ", pcbValue=10 )
[2008-09-24 16:00:25.318109] SQLExtendedFetch( pcRow=7 )
```

Solutions to similar problems

Various sources of information about problems with IBM® licensed programs and their use.

Useful Web sites

Various Web sites are available on the Internet, not all managed by IBM, that you might find useful aids to problem determination. Here are some URLs that you might try:

- [IBM Integration Bus support web page](#)
- [WebSphere® MQ Integrator newsgroup](#)
- [MQSeries®.net](#)

RETAIN

IBM keeps records of all known problems with its licensed programs on its software support database (RETAIN®). IBM Support Center staff continually update this database as new problems that are found. They regularly search the database to see if problems that they are told about are already known.

If you have access to one of IBM's search tools, such as INFORMATION/ACCESS or INFORMATION/SYSTEM, you can look on the RETAIN database yourself. If not, you can contact the IBM Support Center to search for you.

You can search the database using a string of keywords to see if a similar problem exists.

You can use the keyword string (also called the symptom string) that appears in a dump or SYS1.LOGREC record to search the database, or you can build your own keyword string.

If the search is successful, you find a similar problem description and, usually, a fix. If the search is unsuccessful, use these keywords when contacting IBM for additional assistance, or when documenting a possible authorized program analysis report (APAR).

Making initial checks


Before you start problem determination in detail, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save a lot of work by highlighting a simple error, or by narrowing down the range of possibilities.

This section contains a list of questions to consider. As you go through the list, make a note of anything that might be relevant to the problem. Even if your observations do not suggest a cause straight away, they might be useful later if you have to carry out a systematic problem determination exercise.

- [Has IBM Integration Bus run successfully before?](#)
- [Did you log off Windows while IBM Integration Bus components were active?](#)
- [Are the Linux and UNIX environment variables set correctly?](#)
- [Are there any error messages or return codes that explain the problem?](#)
- [Can you reproduce the problem?](#)
- [Has the message flow run successfully before?](#)
- [Have you made any changes since the last successful run?](#)
- [Is there a problem with descriptive text for a command?](#)
- [Is there a problem with a database?](#)
- [Is there a problem with the network?](#)
- [Does the problem affect all users?](#)
- [Have you recently changed a password?](#)
- [Have you applied any service updates?](#)
- [Do you have a component that is running slowly?](#)
- [Additional checks for z/OS users](#)

Are the Linux and UNIX environment variables set correctly?

Use the **mqsiprofile** command to set a command environment.

 On Linux and UNIX systems, the basic settings are made by the **mqsiprofile** command, which is located in the following directory:

```
install_dir/bin
```

Setting up a command environment

After you have installed the product on one of the distributed systems, you must initialize the environment before you can use a runtime component or command.

Also complete this task if you have migrated to IBM® Integration Bus Version 9.0 from an earlier version. A profile is provided to help you set up the environment.

If appropriate, you can extend the initialization performed by this profile; for example, for databases, or for other products that you want to use within the broker.

Ensure that you use this environment each time you run an administrative command, or start a broker.

For information about setting up your command and runtime environment on a Windows system, see [Command environment: Windows systems](#).

For information on setting up your command and runtime environment on Linux and UNIX systems, see [Command environment: Linux and UNIX systems](#).

For information about setting up your command and runtime environment for brokers on Linux and UNIX systems, see [Broker-specific command environment: Linux and UNIX systems](#).

For information about setting up your command and runtime environment for integration servers on Linux and UNIX systems, see [Integration server-specific command environment: Linux and UNIX systems](#).

For information about setting up your command and runtime environment for brokers on Windows systems, see [Broker-specific command environment: Windows systems](#).

For information about setting up your command and runtime environment for integration servers on Windows systems, see [Integration server-specific command environment: Windows systems](#).

Broker-specific command environment: Linux and UNIX systems

Extend or change the Linux or UNIX environment that is used when you run a specific IBM® Integration Bus.

If you want to run your own additional environment settings for a broker, add one or more scripts (with the file extension `.sh`) to the `work_path/config/broker_name/profiles` directory, where `work_path` is the machine-wide IBM Integration Bus working directory, and `broker_name` is the name of your broker.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

Ensure that the broker name, and any integration servers that you create, contain only characters that are valid on your file system. You might also have to create the required directory structure.

All files that are placed into the `work_path/config/broker_name/profiles` directory that have a `.sh` extension, are processed when the broker starts.

Scripts in the `work_path/config/broker_name/profiles` directory are run after the **mqsi** profile script, and any scripts in the `work_path/common/profiles` directory have run.

To diagnose any problems, a log file is written that lists the profile scripts that have been read, and the environment that is used by this broker. The log file location is `work_path/common/log/broker_name.profilelog`.

Integration server profile scripts are not removed automatically when an integration server is deleted. Therefore, if you have created any integration server profile files for this broker, you must delete these profile files manually if they are no longer required.

Broker profile scripts are not removed automatically when a broker is deleted. If you no longer need the profile files that you have created, delete them manually.

Integration server-specific command environment: Linux and UNIX systems

Extend or change the Linux or UNIX environment that is used when you run a specific integration server.

To enable function that becomes available in IBM® Integration Bus fix packs, use the **-f** parameter on the **mqsichangebroker** command. For more information, see [mqsichangebroker command](#).

If you want to run your own additional environment settings, add one or more scripts (with the file extension `.sh`) to the `work_path/config/broker_name/server_name/profiles` directory, where `work_path` is the machine-wide IBM Integration Bus working directory, `broker_name` is the name of your broker, and `server_name` is the name of your integration server.

To verify the machine-wide IBM Integration Bus working directory, enter the following command in a command console:

```
echo %MQSI_WORKPATH%
```

Ensure that the broker name, and any integration servers that you create, contain only characters that are valid on your file system. You might also have to create the required directory structure.

All files that are placed into the `work_path/config/broker_name/server_name/profiles` directory that have a `.sh` extension, are processed when the integration server (`server_name`) on the broker (`broker_name`) starts.

Scripts in

the `work_path/config/broker_name/server_name/profiles` directory are run after the following scripts have run

1. The **mqsiprofile** script.
2. Any scripts in the `work_path/common/profiles` directory.
3. Any scripts in the `work_path/config/broker_name/profiles` directory.

To diagnose any problems, a log file is written that lists the profile scripts that have been read, and the environment that is used by this integration server. The log file location

is `work_path/common/log/broker_name.server_name.profilelog`.

Integration server profile scripts are not removed automatically when an integration server is deleted. If you no longer need the profile files that you have created, delete them manually.

Are there any error messages or return codes that explain the problem?

You can find details of error messages and return codes in several places.

- **BIP messages and reason codes**

IBM® Integration Bus error messages have the prefix BIP. If you receive any messages with this prefix (for example, in the UNIX, Linux, or z/OS® syslog), you can search for them in the information center for an explanation. You can also view the full content of a BIP message by using the **mqsiexplain** command. For more information, see [mqsiexplain command](#).

Windows In the Windows Event log, references to BIP messages are identified by the source name "WebSphere® Broker v6000", where v6000 can be replaced by a number representing the exact service level of your installation, for example 6001.

IBM Integration Bus messages that have a mixture of identifiers such as BIPmsgs, BIPv700, BIPv610, BIPv500, WMQIv210, MQSIv202, and MQSIv201 indicate a mixed installation, which does not work properly.

- **Other messages**

For messages with a different prefix, such as AMQ or CSQ for WebSphere MQ, or SQL for DB2®, see the appropriate messages and codes documentation for a suggested course of action to help resolve the problem.

Messages that are associated with the startup of IBM Integration Bus, or were issued while the system was running before the error occurred, might indicate a system problem that prevented your application from running successfully.

A large or complex IBM Integration Bus broker environment might require some additional configuration of the environment beyond what is recommended in [Installing complementary products](#). The need for such configuration changes is typically indicated by warning or error messages that are logged by the various components, including WebSphere MQ, the databases, and the operating system. These messages are normally accompanied by a suggested user response.

Is there a problem with a database?

If you have database problems, complete a set of initial checks to identify errors.

- Check that the database is started.
- Check that you have correctly completed ODBC configuration:
 - **Linux** **UNIX** On Linux and UNIX systems, check that you have created a copy of the sample ODBC configuration file (`odbc.ini`), and have modified them for your environment, and that you have not added any extra unsupported parameters.

- **Windows** On Windows systems, click **Start > Control Panel > Administrative Tools > Data Sources (ODBC)** to configure the connections you require.

Detailed instructions for setting up ODBC connections on distributed systems are provided in [Enabling ODBC connections to the databases](#).

- Use the [mqsicvp command](#) to help you with ODBC database diagnostics.
- Check that you have correctly completed JDBC configuration. Detailed instructions for setting up JDBC connections are provided in [Enabling JDBC connections to the databases](#).
- Check the number of database connections that are in use on DB2® for AIX®. If you use local mode connections, a maximum of 10 is supported.
- If messages that indicate that `imbdfdb2v6.lil` failed to load, check that you have installed a supported database. Details of database managers and versions are given in [Supported databases](#).

Enabling ODBC connections to the databases

Set up the resources and environment that the broker requires for Open Database Connectivity (ODBC) connections to databases on distributed systems.

You can configure both ODBC and Java™ Database Connectivity (JDBC) connections for access to databases:

- To set up ODBC connections to databases, follow the instructions in this section.
Optionally, after configuring the ODBC connection parameters, run the [mqsicvp command](#) to verify that the broker can connect to the data source, and to provide useful information about the data source and its interface. On Linux and UNIX systems, this command also checks that the ODBC environment is set up correctly.
- To set up JDBC connections to databases, see [Enabling JDBC connections to the databases](#).
- On z/OS® systems, see [Data sources on z/OS](#) for information about enabling connections to databases. You do not have to follow the tasks that are described in this section.
- On Linux and UNIX systems, unixODBC is the supported driver manager and is installed by default as part of the IBM® Integration ODBC Database Extender SupportPac. The SupportPac is automatically installed in silent mode when the installation of the IBM Integration Bus component completes. For more information, see [Installing the IBM Integration ODBC Database Extender](#). DataDirect database drivers continue to be used by the unixODBC driver manager when interfacing with Oracle, Sybase, and SQLServer databases.

Delete the ODBCINI64 environment variable if it exists; it is not required by IBM Integration Bus.

For more information, see [Database connections](#).

The sample `odbc.ini` and `odbcinst.ini` files that are supplied, and the information that is contained in these configuration topics, include all the connection parameters that are supported for connections to your databases. Any additional parameters that are provided by your chosen database drivers are not tested or supported in a broker environment; consider your requirements carefully before specifying other parameters in your tailored ODBC `.ini` files.

To enable connections on distributed systems:

Define the ODBC DSNs according to your platform:

Windows **On Windows:**

Follow the instructions in [Connecting to a database from Windows systems](#).

Linux **UNIX** **On Linux and UNIX systems:**

For all supported databases, follow the instructions in [Connecting to a database from Linux and UNIX systems by using the IBM Integration ODBC Database Extender](#).

You have now configured the ODBC DSNs for your databases.

You have now enabled the broker to make connections to your databases.

Next: If you have been using the instructions in [Working with databases](#), the next task is [Configuring ODBC connections for globally coordinated transactions](#) (optional).

mqsicvp command

Use the **mqsicvp** command to perform verification tests on a broker.

Supported platforms

- Windows
- Linux and UNIX systems





When you start a broker by using the **mqsistart** command, this command is run automatically to verify the component.

z/OS On z/OS®, the same verification procedures are run automatically when you start a broker.

You can run this command against a broker that is running, or is not running. If the broker is not running, the verification tests are performed, but the broker is not started.

Purpose

The **mqsicvp** command completes the following actions:

- The command checks that the broker environment is set up correctly; for example, that the installed level of Java™ is supported.
- The command verifies that the WebSphere® MQ queues are defined and accessible.
-   On Linux and UNIX systems, the command verifies that the ODBC environment (if specified) is configured correctly. If the ODBCINI environment variable is set, the **mqsicvp** command writes warning messages to the syslog in the following circumstances:
 - If the file to which the ODBCINI environment variable points does not exist, or the broker does not have access to read it or write to it
 - If ODBCYSINI is not set
 - If ODBCYSINI is set but the directory to which this variable points does not contain a file named `odbcinst.ini`, or the broker does not have access to read or write to this file
 - If the IE02_PATH environment variable is not set
-   On Linux and UNIX systems, if the ODBC environment check is successful, the command connects to all data sources that are listed in the `odbc.ini` files, and that were associated with the broker by using the **mqsisetdbparms** command.

Use of the command to provide information about user data sources

The data sources that are described are those that you defined when you set up the ODBC database. See [Enabling ODBC connections to the databases](#).

Extra invocations of the **mqsicvp** command provide ODBC test tool function. This function provides useful information about a user data source, or compares two user data sources for equivalence. These invocations of the command are discrete from the basic invocation of the **mqsicvp** command, and are not run at broker startup.

When you use the **mqsicvp** command as an ODBC test tool, the command issues an informational message for a successful connection, providing the name of the data source, database type, and version. If a secondary data source is supplied, the **mqsicvp** command issues a second informational message for a successful connection to that data source, with the same information about the secondary data source, and stating that a comparison will be made.



When the tool is run against one data source, it completes several checks against the ODBC interface to determine which data types and functions

are supported, together with information about the names and sizes of those data types. If any data types or functions are not supported, they are summarized in a final informational message.

When these tests are run against two data sources, they are run against both data sources, and both sets of results are shown. A final informational message states whether the two data sources are equivalent and eligible to be used in the same message processing node; for further information, see [Accessing databases from ESQL](#).

Syntax

When you run the **mqsicvp** command with only the broker name, and no other parameters, the command completes the following checks:

- Checks that the broker environment is set up correctly (for example, the installed level of Java is supported).
- Verifies that the WebSphere MQ queues are defined and accessible.
-   On Linux and UNIX systems only, the command verifies that the ODBC environment (if specified) is configured correctly. If the ODBCINI environment variable is not set, ODBC environment verification is skipped. If the ODBC environment check is completed successfully, the command then tries to connect to all data sources that are listed in the `odbc.ini` files where the **mqsisetdbparms** command was used to associate the data source with the broker.

```
>>-mqsicvp--brokername-----  
-><
```

Syntax

Invocation that uses fully specified data source names

When you run the **mqsicvp** command with parameters shown in the following syntax diagram, the command provides ODBC test tool function. You can use the command to display useful information about a user data source, or compare two user data sources. On Linux and UNIX systems only, invocations of the command that use these parameters also verify that the ODBC environment (if specified) is configured correctly. In these cases, the command checks that the ODBCINI, ODBCYSINI, and IE02_PATH variables are set.

Each data source name (DSN) is fully specified with a user name and password.



This invocation does not rely on an association between the broker and the data source, and returns information about the data source even when there is no association with a broker.

```
>>-mqsicvp-- -n--primaryDataSourceName-----  
-->  
  
>-- -u--primaryDataSourceUserId-----  
-->  
  
>-- -p--primaryDataSourcePassword-----  
-->  
  
>--+-----  
-----+-->  
  
    '- -c--secondaryDataSourceName-- -i--  
secondaryDataSourceUserId-- -a--secondaryDataSourcePassword-  
'  
  
>--+-----+-----  
-><  
  
    '- -v-'
```

Syntax

Invocation where data source names were associated with the broker

This invocation requires an association between the broker and the data source name (DSN). To use this invocation successfully, you must first run the **mqsisetdbparms** command to identify a specific user ID and password for the broker to use when connecting to the data source. The output from this invocation is the same as the output from the invocation that uses fully specified data source names.

  On Linux and UNIX systems only, invocations of the command that use the parameters in the following syntax diagram also



verify that the ODBC environment (if specified) is configured correctly. In these cases, the command checks that the ODBCINI, ODBCYSINI, and IE02_PATH variables are set.

```
>>-mqsicvp--brokername-- -n--primaryDataSource-----  
-->  
  
>--+-----+--+-----+-----  
-><  
  
'- -c--secondaryDataSource- ' '- -v-'
```

Parameters

brokername

(Required - if you are using an invocation that contains *brokername*.) Specify a broker name to verify, or the broker name with which the *primaryDataSource* is associated.

  All names are case-sensitive on Linux and UNIX systems.

-n *primaryDataSource*

(Required if you are using an invocation that contains *primaryDataSource*.) The name of the ODBC connection to verify.

-u *primaryDataSourceUserId*

(Required if you did not previously associate the data source name with the broker.) The user name with which to connect to the *primaryDataSource*.

-p *primaryDataSourcePassword*

(Required if you did not previously associate the data source name with the broker.) The password that is used with the *primaryDataSourceUserId*.

-c *secondaryDataSource*

(Optional) If two data sources are being compared for equivalence, this data source is the second ODBC connection name.

-i *secondaryDataSourceUserId*

(Optional) The user name with which to connect to the *secondaryDataSource*.

-a *secondaryDataSourcePassword*




(Optional) The password that is used with the *secondaryDataSourceUserId*.

-v

(Optional) This parameter causes extra, untranslated, diagnostic information that is related to supported CASTS to be shown by the command.

Authorization

For information about authorizations that are specific to operating systems, see the following topics:

-   [Security requirements for Linux and UNIX platforms](#)
-  [Security requirements for Windows systems](#)

If you enabled broker administration security, you must also set up the authority that is described in [Tasks and authorizations for administration security](#).

Responses

- BIP8040W: No database access (unable to connect).
- BIP8267W: Warning, there might be issues when using this data source. For more information, see the preceding messages.
- BIP8268I: The two data sources that are supplied are compatible, and can be used in the same Compute node.
- BIP8269W: The two data sources that are supplied are not compatible, and must not be used in the same Compute node.
- BIP8270I: Connected to data source <multiple inserts>
- BIP8271I: Connected to second data source <multiple inserts> for comparison.
- BIP8272W: Data source specified is not associated with the broker.
- BIP8273I: The following data types and functions are not natively supported by data source '&1': <multiple inserts>
- BIP8274W: The following data types and functions might cause problems when you use data source '&1' with IBM® Integration Bus: <multiple inserts>
- BIP8288W: Unable to read ODBCINI file '*file_name*'. Check that this file exists, and that the broker user ID has permission to read and write the file.
- BIP8289W: Unable to read ODBCYSINI file in specified directory '*directory_name*'. Check that this

file exists, and that the broker user ID has permission to read and write the file.

- BIP8290I: Verification passed for the ODBC environment.
- BIP8291W: The IE02_PATH environment variable is not set.
- BIP8292I: 'insert1' user data sources were not verified because they do not have **mqsisetdbparms** credentials.
- BIP8294I: ODBC environment verification was skipped because the ODBCINI environment variable is not set.
- BIP8295E: ODBCINI environment variables have not been set, or are invalid.
- BIP8296W: The ODBCSYSINI environment variable is not set.
- BIP8297W: 'environment_variable_name' file 'file_name' is empty.
- BIP8873I: Starting the component verification for broker 'broker_name'.
- BIP8874I: The component verification for 'broker_name' finished successfully.
- BIP8875W: The component verification for 'broker_name' finished, but one or more checks failed.
- BIP8876I: Starting the environment verification for broker 'broker_name'.
- BIP8877W: The environment verification for broker 'broker_name' finished, but one or more checks failed.
- BIP8878I: The environment verification for broker 'broker_name' finished successfully.
- BIP8882I: Starting the WebSphere MQ verification for broker 'broker_name'.
- BIP8883W: The WebSphere MQ verification for broker 'broker_name' finished, but one or more checks failed.
- BIP8884I: The WebSphere MQ verification for broker 'broker_name' finished successfully.
- BIP8885E: Verification failed. Failed to connect to queue manager 'queue_manager_name'.
MQRC: return_code MQCC: completion_code
- BIP8886I: Verification has passed for queue 'queue_name' on queue manager 'queue_manager_name'.
- BIP8887E: Verification has failed for queue 'queue_name' on queue manager 'queue_manager_name' while issuing 'operation'.
MQRC: return_code MQCC: completion_code
- BIP8888E: Verification failed. Failed to disconnect from queue manager 'queue_manager_name'.
MQRC: return_code MQCC: completion_code
- BIP8892E: Verification failed. The installed Java level 'level_installed' does not meet the required Java level 'level_supported'.

- BIP8893E: Verification has failed for environment variable '*variable_name*'. Unable to access file '*file_name*' with user ID '*user_ID*'. Extra information for IBM support: *data1 data2*.
- BIP8894I: Verification has passed for '*broker_name*'.
- BIP8895E: Verification failed. Environment variable '*variable_name*' is incorrect or missing.
- BIP8896E: Verification failed. Unable to access the registry with user ID '*user_ID*'. Extra information for IBM support: *data1 data2*
- BIP8897E: Verification failed. Environment variable '*variable_name*' does not match the broker name '*broker_name*'.
- BIP8900I: APF Authorization check successful for file '*file_name*'.
- BIP8903E: Verification failed. The APF Authorization check failed for file '*file_name*'.
- BIP8904E: Verification failed. Failed to stat file '*file_name1*' with return code '*return_code*' and error number '*error_number*'.

Examples

Run verification checks on the broker named: IB9NODE

```
mqsicvp IB9NODE
```

Data source name (DSN) *MyDB* is associated with the broker IB9NODE by using the **mqsisetdbparms** command:

```
mqsicvp IB9NODE -n MyDB
```

DSN*MyDB* is compared against a secondary DSN *MyDB2* associated with the broker IB9NODE by using the **mqsisetdbparms** command:

```
mqsicvp IB9NODE -n MyDB -c MyDB2
```

The fully qualified DSN*MyDB* is compared against a secondary fully qualified DSN *MyDB2* by using the primary and secondary user IDs and passwords:

```
mqsicvp -n MyDB -u username -p password -c MyDB2 -i  
username2 -a password2
```

Is there a problem with the network?

IBM® Integration Bus uses WebSphere® MQ for inter-component communication. If components are on separate queue managers, they are connected by message channels.

There can be communication problems between any of these:

- Brokers
- The IBM Integration Explorer
- The IBM Integration Toolkit

If any two components are on different queue managers, make sure that the channels between them are working. Use the WebSphere MQ **display chstatus** command to see if messages are flowing.

Use the **ping** command to check that the remote computers are connected to the network, or if you suspect that the problem might be with the network itself. For example, use the command **ping brokername**, where *brokername* is a computer name. If you get a reply, the computer is connected. If you don't get a reply, ask your network administrator to investigate the problem. Further evidence of network problems might be messages building up on the transmission queues.

Do you have a component that is running slowly?

If a particular component, or the system in general, is running slowly, you can take some actions to improve the performance.

Consider the following actions:

- Check whether tracing is on. You might have started IBM® Integration Bus user tracing or service tracing, ODBC tracing, WebSphere® MQ tracing, or native database tracing. If one or more of these traces are active, turn them off.
- Clear out all old abend files from your errors directory. If you do not clear the directory of unwanted files, you might find that your system performance degrades because significant space is used up.
- On Windows, use the workpath **-w** parameter of the [mqsicreatebroker](#) command to create the errors directory in a hard disk partition that does not contain IBM Integration Bus or Windows.
- Increase your system memory.

Message flow statistics and accounting data

You can collect message flow statistics and accounting data in various output formats.

Details of the information that is collected, and the output formats in which it can be recorded, are provided in the following topics:

- [Statistics details](#)
- [Data formats](#)
- [Example output](#)

You can also find information on how to use accounting and statistics data to improve the performance of a message flow in this [developerWorks® article on message flow performance](#).

Message flow accounting and statistics details

You can collect message flow, thread, node, and terminal statistics for message flows.

Message flow statistics

One record is created for each message flow in an integration server. Each record contains the following details:

- Message flow name and UUID
- Integration server name and UUID
- Broker name and UUID
- Start and end times for data collection
- Type of data collected (snapshot or archive)
- Processor and elapsed time spent processing messages
- Processor and elapsed time spent waiting for input
- Number of messages processed
- Minimum, maximum, and average message sizes
- Number of threads available and maximum assigned at any time
- Number of messages committed and backed out
- Accounting origin

Thread statistics

One record is created for each thread assigned to the message flow. Each record contains the following details:

- Thread number (this has no significance and is for identification only)
- Processor and elapsed time spent processing messages
- Processor and elapsed time spent waiting for input
- Number of messages processed
- Minimum, maximum, and average message sizes

Node statistics

One record is created for each node in the message flow. Each record contains the following details:

- Node name
- Node type (for example MQInput)
- Processor time spent processing messages
- Elapsed time spent processing messages
- Number of times that the node is invoked
- Number of messages processed
- Minimum, maximum, and average message sizes

Terminal statistics

One record is created for each terminal on a node. Each record contains the following details:

- Terminal name
- Terminal type (input or output)
- Number of times that a message is propagated to this terminal

For further details about specific output formats, see the following topics:

- [User trace entries for message flow accounting and statistics data](#)
- [XML publication for message flow accounting and statistics data](#)
- [z/OS SMF records for message flow accounting and statistics data](#)

User trace entries for message flow accounting and statistics data

Certain information is written to the user trace log for message flow accounting and statistics data.

The data records are identified by the following message numbers:

- BIP2380I
- BIP2381I
- BIP2382I
- BIP2383I

The inserts for each message are described in the following tables.

This table describes the inserts in message BIP2380I. One message is written for the message flow.

Field	Data type	Details
ProcessID	Numer ic	Process ID
Key	Numer ic	Key that is used to associate related accounting and statistics BIP messages

Field	Data type	Details
Type	Character	Type of output, one of: <ul style="list-style-type: none"> • Archive • Snapshot
Reason	Character	Reason for output, one of: <ul style="list-style-type: none"> • MajorInterval • Snapshot • Shutdown • ReDeploy • StatsSettingsModified
BrokerLabel	Character (maximum 32)	Broker name
BrokerUUID	Character (maximum 32)	Broker universal unique identifier
ExecutionGroupName	Character (maximum 32)	Integration server name
ExecutionGroupUUID	Character (maximum 32)	Integration server universal unique identifier
MessageFlowName	Character (maximum 32)	Message flow name

Field	Data type	Details
	num 32)	
StartDate	Character	Interval start date (YYYY-MM-DD)
StartTime	Character	Interval start time (HH:MM:SS:NNNNNN)
EndDate	Character	Interval end date (YYYY-MM-DD)
EndTime	Character	Interval end time (HH:MM:SS:NNNNNN)
TotalElapsedTime	Numerical	Total elapsed time spent processing input messages (microseconds)
MaximumElapsedTime	Numerical	Maximum elapsed time that is spent processing an input message (microseconds)
MinimumElapsedTime	Numerical	Minimum elapsed time that is spent processing an input message (microseconds)
TotalCPUTime	Numerical	Total processor time spent processing input messages (microseconds)
MaximumCPUTime	Numerical	Maximum processor time that is spent processing an input message (microseconds)

Field	Data type	Details
MinimumCPUTime	Numer ic	Minimum processor time that is spent processing an input message (microseconds)
CPUTimeWaitingForInputMessage	Numer ic	Total processor time spent waiting for input messages (microseconds)
ElapsedTimeWaitingForInputMessage	Numer ic	Total elapsed time that is spent waiting for input messages (microseconds)
TotalInputMessages	Numer ic	Total number of messages processed TotalInputMessages records only those messages that are propagated from input node terminals.
TotalSizeOfInputMessages	Numer ic	Total size of input messages (bytes)
MaximumSizeOfInputMessages	Numer ic	Maximum input message size (bytes)
MinimumSizeOfInputMessages	Numer ic	Minimum input message size (bytes)
NumberOfThreadsInPool	Numer ic	Number of threads in pool
TimesMaximumNumberofThreadsReached	Numer ic	Number of times the maximum number of threads is reached
TotalNumberOfMQErrors	Numer ic	Number of MQGET errors (MQInput node)

Field	Data type	Details
		<p>or web services errors (HTTPInput node)</p> <p>For example, a conversion error occurs when the message is got from the queue.</p>
TotalNumberOfMessagesWithErrors	Numer ic	<p>Number of messages that contain errors</p> <p>These errors include exceptions that are thrown downstream of the input node, and errors that are detected by the input node after it successfully retrieves the message from the queue (for example, a format error).</p> <p>TotalNumberOfMessagesWithErrors can include messages that are not included in TotalInputMessages.</p>
TotalNumberOfErrorsProcessingMessages	Numer ic	Number of errors while processing a message
TotalNumberOfTimeOutsWaitingForRepliesToAggregateMessages	Numer ic	Number of timeouts while processing a message (AggregateReply node only)
TotalNumberOfCommits	Numer ic	Number of transaction commits
TotalNumberOfBackouts	Numer ic	Number of transaction backouts
AccountingOrigin	Charac ter (maxi	Accounting origin

Field	Data type	Details
	num 32)	

The following table describes the inserts in message BIP2381I. One message is written for each thread.

Field	Data type	Details
ProcessID	Numeric	Process ID
Key	Numeric	Key that is used to associate related accounting and statistics BIP messages
Number	Numeric	Relative thread number in pool
TotalNumberOfInputMessages	Numeric	Total number of messages that are processed by a thread
TotalElapsedTime	Numeric	Total elapsed time spent processing input messages (microseconds)
TotalCUPTime	Numeric	Total processor time spent processing input messages (microseconds)

Field	Data type	Details
CPUTimeWaitingForInputMessage	Numeric	Total processor time spent waiting for input messages (microseconds)
ElapsedTimeWaitingForInputMessage	Numeric	Total elapsed time that is spent waiting for input messages (microseconds)
TotalSizeOfInputMessages	Numeric	Total size of input messages (bytes)
MaximumSizeOfInputMessages	Numeric	Maximum size of input messages (bytes)
MinimumSizeOfInputMessages	Numeric	Minimum size of input messages (bytes)

The following table describes the inserts in message BIP2382I. One message is written for each node.

Field	Data type	Details
ProcessID	Numeric	Process ID
Key	Numeric	Key that is used to associate related accounting and statistics BIP messages
Label	Character	Name of node (Label)

Field	Data type	Details
Type	Character	Type of node
TotalElapsedTime	Numeric	Total elapsed time spent processing input messages (microseconds)
MaximumElapsedTime	Numeric	Maximum elapsed time spent processing input messages (microseconds)
MinimumElapsedTime	Numeric	Minimum elapsed time spent processing input messages (microseconds)
TotalCPUTime	Numeric	Total processor time spent processing input messages (microseconds)
MaximumCPUTime	Numeric	Maximum processor time spent processing input messages (microseconds)
MinimumCPUTime	Numeric	Minimum processor time spent processing input messages (microseconds)
CountOfInvocations	Numeric	Total number of messages that are processed by this node
NumberOfInputTerminals	Numeric	Number of input terminals
NumberOfOutputTerminals	Numeric	Number of output terminals

The following table describes the inserts in message BIP2383I. One message is written for each terminal on each node.

Field	Data type	Details
ProcessID	Numeric	Process ID
Key	Numeric	Key that is used to associate related accounting and statistics BIP messages
Label	Character	Name of terminal
Type	Character	Type of terminal, one of: <ul style="list-style-type: none">• Input• Output
CountOfInvocations	Numeric	Total number of invocations

Example of an XML publication for message flow accounting and statistics

This example shows an XML publication that contains message flow accounting and statistics data.

The following example shows what is generated for a snapshot report. The content of this publication message shows that the message flow is called *XMLflow*, and that it is running in an integration server named *defaulton* broker *MQ02BRK*. The message flow contains the following nodes:

- An MQInput node called *INQueue3*
- An MQOutput node called *OUTQueue*
- An MQOutput node called *FAILQueue*

The MQInput node's Out terminal is connected to the OUTQueue node. The MQInput node's Failure terminal is connected to the FAILQueue node.

During the interval for which statistics have been collected, this message flow processed no messages.

A publication that is generated for this data always includes the appropriate folders, even if there is no current data.

The following command has been issued to achieve these results:

```
mqsichangeflowstats MQ02BRK -s -c active -e default -f XMLFlow -n advanced  
-t basic -b basic -o xml
```

Blank lines have been added between folders to improve readability.

The broker takes information about statistics and accounting from the operating system. On some operating systems, such as Windows, UNIX, and Linux, rounding can occur because the system calls that are used to determine the processor times are not sufficiently granular. This rounding might affect the accuracy of the data.

The following example is the subscription message.

The <psc> and <mcd> elements are part of the RFH header.

```
<psc>  
  <Command>Publish</Command>  
  <PubOpt>RetainPub</PubOpt>  
  <Topic>$SYS/Broker/MQ02BRK/StatisticsAccounting/SnapShot/default/XMLflow  
  </Topic>  
</psc>  
  
<mcd>  
  <Msd>xml</Msd>  
</mcd>
```

The following example is the publication that the broker generates:

```
<WMQIStatisticsAccounting RecordType="SnapShot" RecordCode="Snapshot">  
  
<MessageFlow BrokerLabel="MQ02BRK"  
  BrokerUUID="7d951e31-f200-0000-0080-efelb9d849dc"  
  ExecutionGroupName="default"  
  ExecutionGroupUUID="77cf1e31-f200-0000-0080-efelb9d849dc"  
  MessageFlowName="XMLflow" StartDate="2003-01-17"  
  StartTime="14:44:34.581320" EndDate="2003-01-17"  
  EndTime="14:44:44.582926"  
  TotalElapsedTime="0"  
  MaximumElapsedTime="0" MinimumElapsedTime="0" TotalCPUtime="0"  
  MaximumCPUtime="0" MinimumCPUtime="0" CPUtimeWaitingForInputMessage="685"  
  ElapsedTimeWaitingForInputMessage="10001425" TotalInputMessages="0"  
  TotalSizeOfInputMessages="0" MaximumSizeOfInputMessages="0"  
  MinimumSizeOfInputMessages="0" NumberOfThreadsInPool="1"  
  TimesMaximumNumberOfThreadsReached="0" TotalNumberOfMQErrors="0"
```

```

    TotalNumberOfMessagesWithErrors="0"
    TotalNumberOfErrorsProcessingMessages="0"
    TotalNumberOfTimeOutsWaitingForRepliesToAggregateMessages="0"
    TotalNumberOfCommits="0" TotalNumberOfBackouts="0"
    AccoutingOrigin="DEPT1"/>

<Threads Number="1">
<ThreadStatistics Number="5" TotalNumberOfInputMessages="0"
TotalElapsedTime="0" TotalCPUTime="0" CPUTimeWaitingForInputMessage="685"
ElapsedTimeWaitingForInputMessage="10001425" TotalSizeOfInputMessages="0"
MaximumSizeOfInputMessages="0" MinimumSizeOfInputMessages="0"/>
</Threads>

<Nodes Number="3">

    <NodeStatistics Label="FAILQueue" Type="MQOutput" TotalElapsedTime="0"
        MaximumElapsedTime="0" MinimumElapsedTime="0" TotalCPUTime="0"
        MaximumCPUTime="0" MinimumCPUTime="0" CountOfInvocations="0"
        NumberOfInputTerminals="1" NumberOfOutputTerminals="2">
        <TerminalStatistics Label="failure" Type="Output"
CountOfInvocations="0"/>
        <TerminalStatistics Label="in" Type="Input" CountOfInvocations="0"/>
        <TerminalStatistics Label="out" Type="Output" CountOfInvocations="0"/>
    </NodeStatistics>

    <NodeStatistics Label="INQueue3" Type="MQInput" TotalElapsedTime="0"
        MaximumElapsedTime="0" MinimumElapsedTime="0" TotalCPUTime="0"
        MaximumCPUTime="0" MinimumCPUTime="0" CountOfInvocations="0"
        NumberOfInputTerminals="0" NumberOfOutputTerminals="3">
        <TerminalStatistics Label="catch" Type="Output"
CountOfInvocations="0"/>
        <TerminalStatistics Label="failure" Type="Output"
CountOfInvocations="0"/>
        <TerminalStatistics Label="out" Type="Output" CountOfInvocations="0"/>
    </NodeStatistics>

    <NodeStatistics Label="OUTQueue" Type="MQOutput" TotalElapsedTime="0"
        MaximumElapsedTime="0" MinimumElapsedTime="0" TotalCPUTime="0"
        MaximumCPUTime="0" MinimumCPUTime="0" CountOfInvocations="0"
        NumberOfInputTerminals="1" NumberOfOutputTerminals="2">
        <TerminalStatistics Label="failure" Type="Output"
CountOfInvocations="0"/>
        <TerminalStatistics Label="in" Type="Input" CountOfInvocations="0"/>

```

```

    <TerminalStatistics Label="out" Type="Output" CountOfInvocations="0"/>
  </NodeStatistics>

</Nodes>

</WMQIStatisticsAccounting>

```

mqsichangeflowstats command - Windows, Linux and UNIX systems

Use the mqsichangeflowstats command on Windows, Linux and UNIX systems to control the accumulation of statistics about message flow operation.

Syntax

Read syntax diagramSkip visual syntax diagram

```
>>-mqsichangeflowstats-- BrokerName --+ -a +----->
```

```
    '- -s '
```

(1)

```
>--+ -e -IntegrationServerName--+ -f -MessageFlow--+----->
```

```
    |                '- -j-----' |
```

```
    '- -g - -j-----'
```

```
>--+-----+--+-----+----->
```

```
    '- -b -AccountingOrigin-' '- -c -Control-'
```

```
>--+-----+--+-----+----->
```

```
    '- -k -applicationName-' '- -n -NodeData-'
```

```
>--+-----+--+-----+----->
```

```
    '- -o -OutputFormat-' '- -r ' '- -t -ThreadData-'
```

>+-----+-----><

'- -y -libraryName-'

Notes:

See the parameter descriptions for information on valid combinations of the e, f, g, and j parameters.

Parameters

BrokerName

(Required) Specify the label of the broker for which accounting and statistics are to be changed.

-a

(Required) Specify that the command modifies archive accounting and statistics collection.

You must specify either -a or -s. If you do not specify one of these arguments you receive a warning message.

-b AccountingOrigin

(Optional) Specifies that the environment tree path Broker.Accounting.Origin is used to partition the collected statistics into distinct outputs. Possible values are:

none - do not partition statistics according to accounting origin data

basic - partition statistics according to accounting origin data

-c Control

(Optional) Specify the string value that controls the level of the action to be applied to accounting and statistics collection for snapshot or archiving. Possible values are:

active - turn on snapshot or archiving

inactive - turn off snapshot or archiving.

-e IntegrationServerName

(Required) Specify the name for the integration server, for which accounting and statistics options are to be changed.

You must specify either -e or -g. If you do not specify one of these arguments you receive a warning message.

-f MessageFlow

(Required) Specify the label for the message flow, for which accounting and statistics options are to be changed.

You must specify either -f or -j. If you do not specify one of these arguments you receive a warning message.

-g

(Required) Specifies that the command applies to all integration servers that belong to the broker.

You must specify either -e or -g. If you do not specify one of these arguments you receive a warning message.

-j

(Required) Specifies that the command applies to all message flows that belong to the integration server.

You must specify either -f or -j. If you do not specify one of these arguments you receive a warning message.

Note: If you set the -g option for all integration servers, you must use -j instead of -f.

-k applicationName

(Optional) This parameter specifies the name of the application that contains the message flow for which statistics are being collected.

When you set this parameter, you must also set either the -f (message flow) parameter, or the -j (all message flows) parameter. To collect statistics for a named message flow in a specific application, set the -k and -f parameters. To collect statistics for all message flows in an application, set the -k and -j parameters.

You can use this parameter with the -y (library name) parameter to specify that the message flow is contained in a library referenced by the application.

-n NodeData

(Optional) Specify a string value to modify the collection of node statistics data for a message flow. Possible values are:

none - exclude node related data in the statistics

basic - include node related statistics in the statistics

advanced - include node related and terminal related data in the statistics

-o OutputFormat

(Optional) Specify the output destination for the statistics reports. You can specify multiple output destinations by using comma-separated values. Possible values are:

usertrace - this is the default and writes "bip" messages to usertrace, which can be post processed in the normal way using the mqsireadlog and mqsiformatlog commands

xml - the statistics reports are generated as XML documents and published by the broker running the message flow.

The topic on which the data is published has the following structure:

```
$SYS/Broker/<brokerName>/StatisticsAccounting/<recordType>  
/  
/executionGroupLabel/<messageFlowLabel>
```

where recordType is set to Snapshot or Archive, and broker, integration server, and message flow names are specified according to the subscriber's requirements.

json - the statistics reports are generated in JSON format, which enables the statistical information to be viewed in the web user interface. If statistics collection is turned on in the web user interface, the output format is changed to json. The topic on which the data is published has the following structure:

```
$SYS/Broker/<brokerName>/Statistics/JSON/SnapShot/isName/applications/MyApplication  
/  
/libraries/MyLibrary/messageflows/MyFlowName
```

-r

(Optional) This parameter applies only to archive data and specifies that archive data is to be reset.

This results in the clearing out of accounting and statistics data accumulated so far for this interval, and restarts collection from this point. All archive data for all flows in the integration server, or groups, is reset.

The archive interval timer is only reset if the -v option (statistics archive interval) of mqsicreatebroker or mqsichangebroker is non zero.

That is, the interval timer is set only if the internal interval notification mechanism is being used, and not an external method.

-s

(Required) Specify that the command modifies snapshot accounting and statistics collection.

You must specify either -a or -s. If you do not specify one of these arguments you receive a warning message.

-t ThreadData

(Optional) Specify a string value to modify the collection of thread statistics data for a message flow
Possible values are:

none - exclude thread related data from the statistics

basic - include thread related data in the statistics

-y libraryName

(Optional) This parameter specifies the name of the library that contains the message flow for which statistics are being collected.

When you set this parameter, you must also set either the -f (message flow) parameter, or the -j (all message flows) parameter. To collect statistics for a named message flow in a specific library, set the -k and -f parameters. To collect statistics for all message flows in a library, set the -k and -j parameters.

Examples

Turn on snapshot statistics for the message flow "myFlow1" in all integration servers of BrokerA and specify that the statistics are not to be partitioned according to accounting origin data:

```
mqsischange flowstats BrokerA -s -g -j -b none -c active
```

Turn on the collection of snapshot statistics data for all message flows in all integration servers for Broker A, and emit the data in XML and JSON formats :

```
mqsischange flowstats BrokerA -s -g -j -c active -o xml,json
```

Turn off the collection of archive statistics for message flow "MyFlow1" in integration server "EGRP2" for BrokerA, and at the same time modify the granularity of data that is to be collected (when next activated) to include thread related data.

```
mqsischange flowstats BrokerA -a -e EGRP2 -f MyFlow1 -c inactive -t basic
```

Turn off snapshot data for all message flows in all integration servers for Broker A.

```
mqsischange flowstats BrokerA -s -g -j -c inactive
```

Message flow accounting and statistics output formats

The message flow accounting and statistics data can be written in the following four formats: user trace, XML, JSON, and SMF.

You can specify one or more output formats for accounting and statistics data, depending on the way in which you want to view the data:

- XML format is required if you want to view the statistics data in the IBM® Integration Explorer. If statistics collection is turned on through the IBM

Integration Explorer, the data is emitted in XML format, in addition to any other formats that are being emitted.

- JSON format is required if you want to view the statistics data in the web user interface. If statistics collection is turned on through the web user interface, the data is emitted in JSON format, in addition to any other formats that are being emitted.
- User trace log entries can be updated with statistics data. This is the default destination for statistics data if no output format has been specified.
- SMF records can be updated with the statistics data emitted on z/OS®. Accounting and statistics data uses SMF type 117 records.

For information about the available output formats, see the following topics:

- [XML publication](#)
- [JSON publication](#)
- [User trace entries](#)
- [z/OS SMF records](#)

User trace entries for message flow accounting and statistics data

Certain information is written to the user trace log for message flow accounting and statistics data.

The data records are identified by the following message numbers:

- BIP2380I
- BIP2381I
- BIP2382I
- BIP2383I

The inserts for each message are described in the following tables.

This table describes the inserts in message BIP2380I. One message is written for the message flow.

Field	Data type	Details
ProcessID	Numeric	Process ID

Field	Data type	Details
Key	Numer ic	Key that is used to associate related accounting and statistics BIP messages
Type	Charac ter	Type of output, one of: <ul style="list-style-type: none"> • Archive • Snapshot
Reason	Charac ter	Reason for output, one of: <ul style="list-style-type: none"> • MajorInterval • Snapshot • Shutdown • ReDeploy • StatsSettingsM odified
BrokerLabel	Charac ter (maxi mum 32)	Broker name
BrokerUUID	Charac ter (maxi mum 32)	Broker universal unique identifier
ExecutionGroupName	Charac ter (maxi mum 32)	Integration server name
ExecutionGroupUUID	Charac ter (maxi	Integration server universal unique identifier

Field	Data type	Details
	num 32)	
MessageFlowName	Charac ter (maxi mum 32)	Message flow name
StartDate	Charac ter	Interval start date (YYYY-MM-DD)
StartTime	Charac ter	Interval start time (HH:MM:SS:NNNNNN)
EndDate	Charac ter	Interval end date (YYYY-MM-DD)
EndTime	Charac ter	Interval end time (HH:MM:SS:NNNNNN)
TotalElapsedTime	Numer ic	Total elapsed time spent processing input messages (microseconds)
MaximumElapsedTime	Numer ic	Maximum elapsed time that is spent processing an input message (microseconds)
MinimumElapsedTime	Numer ic	Minimum elapsed time that is spent processing an input message (microseconds)
TotalCPUTime	Numer ic	Total processor time spent processing input messages (microseconds)

Field	Data type	Details
MaximumCPUTime	Numer ic	Maximum processor time that is spent processing an input message (microseconds)
MinimumCPUTime	Numer ic	Minimum processor time that is spent processing an input message (microseconds)
CPUTimeWaitingForInputMessage	Numer ic	Total processor time spent waiting for input messages (microseconds)
ElapsedTimeWaitingForInputMessage	Numer ic	Total elapsed time that is spent waiting for input messages (microseconds)
TotalInputMessages	Numer ic	Total number of messages processed TotalInputMessages records only those messages that are propagated from input node terminals.
TotalSizeOfInputMessages	Numer ic	Total size of input messages (bytes)
MaximumSizeOfInputMessages	Numer ic	Maximum input message size (bytes)
MinimumSizeOfInputMessages	Numer ic	Minimum input message size (bytes)
NumberOfThreadsInPool	Numer ic	Number of threads in pool

Field	Data type	Details
TimesMaximumNumberofThreadsReached	Numer ic	Number of times the maximum number of threads is reached
TotalNumberOfMQErrors	Numer ic	<p>Number of MQGET errors (MQInput node) or web services errors (HTTPInput node)</p> <p>For example, a conversion error occurs when the message is got from the queue.</p>
TotalNumberOfMessagesWithErrors	Numer ic	<p>Number of messages that contain errors</p> <p>These errors include exceptions that are thrown downstream of the input node, and errors that are detected by the input node after it successfully retrieves the message from the queue (for example, a format error).</p> <p>TotalNumberOfMessagesWithErrors can include messages that are not included in TotalInputMessages.</p>
TotalNumberOfErrorsProcessingMessages	Numer ic	Number of errors while processing a message
TotalNumberOfTimeOutsWaitingForRepliesToAggregateMessages	Numer ic	<p>Number of timeouts while processing a message (AggregateReply node only)</p>
TotalNumberOfCommits	Numer ic	Number of transaction commits

Field	Data type	Details
TotalNumberOfBackouts	Numeric	Number of transaction backouts
AccountingOrigin	Character (maximum 32)	Accounting origin

The following table describes the inserts in message BIP2381I. One message is written for each thread.

Field	Data type	Details
ProcessID	Numeric	Process ID
Key	Numeric	Key that is used to associate related accounting and statistics BIP messages
Number	Numeric	Relative thread number in pool
TotalNumberOfInputMessages	Numeric	Total number of messages that are processed by a thread
TotalElapsedTime	Numeric	Total elapsed time spent processing input messages (microseconds)

Field	Data type	Details
TotalCUPTime	Numeric	Total processor time spent processing input messages (microseconds)
CPUTimeWaitingForInputMessage	Numeric	Total processor time spent waiting for input messages (microseconds)
ElapsedTimeWaitingForInputMessage	Numeric	Total elapsed time that is spent waiting for input messages (microseconds)
TotalSizeOfInputMessages	Numeric	Total size of input messages (bytes)
MaximumSizeOfInputMessages	Numeric	Maximum size of input messages (bytes)
MinimumSizeOfInputMessages	Numeric	Minimum size of input messages (bytes)

The following table describes the inserts in message BIP2382I. One message is written for each node.

Field	Data type	Details
ProcessID	Numeric	Process ID
Key	Numeric	Key that is used to associate related accounting and statistics BIP messages

Field	Data type	Details
Label	Character	Name of node (Label)
Type	Character	Type of node
TotalElapsedTime	Numeric	Total elapsed time spent processing input messages (microseconds)
MaximumElapsedTime	Numeric	Maximum elapsed time spent processing input messages (microseconds)
MinimumElapsedTime	Numeric	Minimum elapsed time spent processing input messages (microseconds)
TotalCPUTime	Numeric	Total processor time spent processing input messages (microseconds)
MaximumCPUTime	Numeric	Maximum processor time spent processing input messages (microseconds)
MinimumCPUTime	Numeric	Minimum processor time spent processing input messages (microseconds)
CountOfInvocations	Numeric	Total number of messages that are processed by this node
NumberOfInputTerminals	Numeric	Number of input terminals

Field	Data type	Details
NumberOfOutputTerminals	Numeric	Number of output terminals

The following table describes the inserts in message BIP2383I. One message is written for each terminal on each node.

Field	Data type	Details
ProcessID	Numeric	Process ID
Key	Numeric	Key that is used to associate related accounting and statistics BIP messages
Label	Character	Name of terminal
Type	Character	Type of terminal, one of: <ul style="list-style-type: none"> • Input • Output
CountOfInvocations	Numeric	Total number of invocations

Resource statistics data: FTEAgent

Learn about the data that is returned for the FTEAgent resource type when you activate resource statistics collection.

You can view these statistics in the IBM® Integration Explorer, or you can write a program that subscribes to a publication (single XML message) that returns this data. For an example of the publication message, see [Example XML output](#).

The FTE agent is an embedded FTE component that allows IBM Integration Bus to send and receive files by using WebSphere® MQ File Transfer Edition. A one-to-one mapping exists between an integration

server and its FTE agent. Therefore, the resource manager statistics provide an accurate picture of the WebSphere MQ File Transfer Edition activity of that integration server.

The following table describes the measurements that are returned for the agent.

Measurements	Description
inboundTransfers	The number of transfers received by the agent.
outboundTransfers	The number of transfers sent by the agent.
inboundBytes	The number of bytes received by the agent.
outboundBytes	The number of bytes sent by the agent.

Subscribing to statistics reports

You can subscribe to topics that return statistics about the operation of your message flows and resource managers.

Message flow performance

If you enable message flow accounting and statistics collection for a broker, you can subscribe to the messages that the broker publishes on the following topic:

```
$SYS/Broker/brokerName/StatisticsAccounting/recordType/integrationServerLabel/messageFlowLabel
```

where *broker_name* is the name of the broker, *recordType* is the type of record (SnapShot or Archive), *integrationServerLabel* is the name of the integration server that you created on that broker, and *messageFlowLabel* is the name of the message flow that you deployed to the integration server.

These messages contain statistics reports and are published at a regular interval, which you control by setting the statsInterval property of the broker. Each publication is a JMS TextMessage that contains the statistics report in XML format.

Note: If you need to revert to using a JMS BytesMessage format, this can be achieved by setting the environment variable **MQSI_STATS_MQSTR=false**.

Resource performance

If you enable resource statistics collection for one or more integration servers on a broker, you can subscribe to the messages that the broker publishes at 20-second intervals on the following topic:

```
$SYS/Broker/brokerName/ResourceStatistics/integrationServerLabel
```

For more information about how to interpret the resource statistics that are included in the publication, see [Viewing resource statistics data in the IBM Integration Explorer](#).

Using wildcards in subscriptions

You can use wild cards when you subscribe to statistics reports. For example, to receive message flow statistics reports for all brokers and all integration servers, subscribe to the following topic:

```
$SYS/Broker/+/StatisticsAccounting/#
```

To receive integration server resource statistics reports for all brokers and all integration servers, subscribe to the following topic:

```
$SYS/Broker/+/ResourceStatistics/#
```

For further details about how you can use wildcards, see [Special characters in topics](#).

Subscribers receive statistics reports only from those brokers that are enabled to produce statistics.

Viewing resource statistics data in the IBM Integration Explorer

Use the IBM® Integration Explorer to view resource statistics data for your integration servers in the Broker Resources and Broker Resources Graph views.

Before you start:

- Read the concept topic, [Resource statistics](#).
- Start resource statistics collection: [Starting resource statistics collection in the IBM Integration Explorer](#)

You can also view resource statistics collection by subscribing to the topic on which statistics are published. For further details, see [Subscribing to statistics reports](#).

To view resource statistics in the IBM Integration Explorer, complete the following steps.

1. In the **WebSphere MQ Explorer - Navigator** view, expand the Integration Nodes folder.
2. To open the Resource Statistics and Resource Statistics Graph views, click **Window > Show View > Resource Statistics**. These two views are displayed together. If you close one of the views, the other view is also closed.
3. Use the information in these views to review the use of resources for which statistics are available.

The following examples demonstrate the types of question that can be answered by collecting resource statistics. This list is not exhaustive, and does include all resource types. For a full list of resource types, and the type of information that is collected for each one, see [Resource statistics data](#)

JVM statistics

How much memory is the JVM using?

Many tools that are specific to an operating system give you the total memory that is used by the integration server, but they do not show you how that memory is divided between Java™ processing and other processing in the integration server. By looking at the field `CommittedMemoryInMB`, you can see how much memory is currently allocated to the JVM. Then look at the field `MaxMemoryInMB` to see the maximum amount of memory that can be allocated.

How often is garbage collection done? Is it affecting the performance of the integration server?

To see how often the JVM is doing garbage collection, check the `CumulativeNumberOfGCCollections` field to see if the rate of collections is increasing. Garbage collection is a normal process, and is therefore expected to some degree. However, excessive garbage collection can affect performance.

To see if current garbage collection is excessive, monitor the `CumulativeGCTimeInSeconds` value. If this value is increasing by more than 2 seconds in each 20-second statistics interval, try increasing the JVM maximum heap size for your integration server by using the **`mqsichangeproperties`** command. You might also want to inspect all the Java user-defined nodes and JavaCompute nodes that are included in your deployed message flows, to ensure that they do not create and delete many objects that could be reused; frequent deletions can contribute to excessive garbage collection.

Do I need to change the minimum or maximum heap sizes?

- If the `CumulativeGCTimeInSeconds` value is increasing by more than 2 seconds in each 20 second statistics interval, increase the maximum heap size to reduce this increase.
- If the `UsedMemoryInMB` value is never close to the `InitialMemoryInMB` value, you might have allocated more memory for the heap than is required. Therefore, reduce the JVM minimum heap size value for the integration server to a value that is closer to the `UsedMemoryInMB` value.

Change these values gradually, and check the results to find the optimum settings for your environment.

Parsers

Are message parsers using more memory than expected?

A message flow parses input messages and can create many output messages. These messages may have large bit streams or large message trees. The parsers created to perform this message processing might consume a large amount of memory. Use the Parsers statistics to determine if message flow parsers are using more memory than expected. If so, consider deploying such flows into separate integration servers or improving ESQL or Java plugin API processing to efficiently handle large messages or transformations.

Is message parsing or writing failing frequently for a particular message flow?

If a message flow receives or attempts to write an invalid message, it is likely that this will be rejected by a parser. Use the message parsers statistics to see if a message flow is rejecting a large amount of input or output messages compared with successful processing.

Outbound sockets

Are the nodes reusing outbound sockets?

Creating outbound sockets can be an expensive operation, and the number of sockets available on a computer is a finite resource. Therefore, increasing socket reuse can enhance performance. If the workload is continuous and consistent, the `TotalSockets` value indicates an initial period of activity, which then reduces when the integration server starts to reuse sockets.

A steady increase in the `TotalSockets` value over time is expected because sockets are closed after a period of inactivity, or when they have been used many times.

If the `TotalSockets` value increases significantly over time, this trend might indicate that outbound sockets are not being reused. If your message flows include `HTTPRequest` nodes, check that you have set the `keepalive` property *Enable HTTP/1.1 keepalive*. Check also whether the endpoint that is called uses keepalive sockets.

Which endpoints are most used?

The values `TotalMessages` indicates how busy each endpoint is. The value in the summary record tells you how much activity occurred across the whole integration server.

How large are sent and received messages?

The values of the `SentMessageSize_*` and `ReceivedMessageSize_*` fields give a profile of the message sizes flowing to and from each endpoint.

JDBC connection pools

Do I need to change the size of the connection pool?

If the statistics show that the count of callers waiting for connections is high, and the wait time is increasing, consider increasing the size of the pool using the `MaxConnectionPoolSize` property for the `JDBCProvider` configurable service. Alternatively, try reducing the number of additional instances configured for the message flow.

TCPIPClientNodes

Are the nodes reusing outbound sockets?

Creating outbound sockets can be an expensive operation, and the number of sockets available on a computer is a finite resource. Therefore, increasing socket reuse can enhance performance. If the workload is continuous and consistent, the `TotalSockets` value indicates an initial period of activity, which then reduces when the integration server starts to reuse sockets.

A steady increase in the `TotalSockets` value over time is expected because sockets are closed after a period of inactivity, or when they have been used many times.

If the `TotalSockets` value increases significantly over time, this trend might indicate that outbound sockets are not being reused. If your message flows include `HTTPRequest` nodes, check that you have set the keepalive property *Enable HTTP/1.1 keepalive*. Check also whether the endpoint that is called uses keepalive sockets.

How the does the information that I see in IBM Integration Explorer relate to my TCP/IP flows?

An entry is displayed for each configurable service, not for each flow.

Starting resource statistics collection in the IBM Integration Explorer

Use the IBM® Integration Explorer to start collecting resource statistics data for your integration servers. You can then view the data in the Broker Resources and Broker Resources Graph views.

Before you start:

- Ensure that you have created a message flow; for more information, see [Creating a message flow](#).
- Ensure that you have deployed a broker archive (BAR) file; for more information, see [Deploying a broker archive file](#).
- Read the concept topic, [Resource statistics](#).

Use the resource statistics data to monitor the performance and resource usage in your integration servers. You can start collecting data for active integration servers at any time.

If you prefer, you can start resource statistics collection by using the **mqsichangeresourcestats** command; for more information, see [Starting resource statistics collection](#).

To start resource statistics collection in the IBM Integration Explorer, complete the following steps.

1. In the WebSphere MQ Explorer - Navigator view, expand the Integration Nodes folder, then expand the folder of the integration node with which you are working.
2. Right-click the integration server for which you want to collect statistics. If you want statistics for several integration servers on this integration node, select more than one by using standard operating system interfaces. For example, on Windows systems, hold down the **Ctrl** key and select each integration server before you right-click to open the menu.
3. Click **Statistics > Start Resource Statistics**. A message is sent to the integration node to start collecting data for the selected integration servers. The property *Resource Statistics Active* is updated in the properties view of each affected integration server to indicate that data collection is now active.
A warning message is displayed in the integration server properties window to warn you that performance might be affected by your action.
4. To view statistics, click **Window > Show View > Resource Statistics** to open the Broker Resources and Broker Resources Graph views. If you are displaying statistics for this integration server for the first time, the views might be empty until the first data is received. Update messages are sent every 20 seconds, and the views refresh automatically.
The Broker Resources and Broker Resources Graph views are displayed together. If you close one of the views, the other view is also closed.

Stopping resource statistics collection in the IBM Integration Explorer

Use the IBM® Integration Explorer to stop collecting resource statistics data for your integration servers.

Before you start:

- Read the concept topic about [resource statistics](#).
- [Start resource statistics collection](#)

You can stop collecting data for active integration servers at any time.

If you prefer, you can stop collecting resource statistics by using the **mqsichangeresourcestats** command; see [Stopping resource statistics collection](#).

To stop collecting resource statistics in the IBM Integration Explorer:

1. In the **WebSphere MQ Explorer - Navigator** view, expand the Integration Nodes folder.
2. Right-click the integration node or integration server for which you want to stop statistics collection. If you want to stop statistics collection for several integration servers, you can select more than one by using standard operating system interfaces; for example, on Windows systems, hold down the Ctrl key and select each integration server before you right-click to open the menu.
3. Click **Statistics > Stop Resource Statistics**. A message is sent to the integration node to stop collecting resource data for the selected integration servers. The status of data collection is updated in the properties view for each affected integration server.

If you click **Window > Show View > Resource Statistics** to open the Broker Resources and Broker Resources Graph views when statistics collection is not active, the data that is displayed represents the last update message received by the IBM Integration Explorer. If you have never started statistics collection for this integration server, the views are displayed but contain no data.

Viewing resource statistics data in the WebSphere Message Broker Explorer

Use the WebSphere® Message Broker Explorer to view resource statistics data for your execution groups in the Broker Resources and Broker Resources Graph views.

Before you start:

- Read the concept topic, [Resource statistics](#).

- Start resource statistics collection: [Starting resource statistics collection in the WebSphere Message Broker Explorer](#)

You can also view resource statistics collection by subscribing to the topic on which statistics are published. For further details, see [Subscribing to statistics reports](#).

To view resource statistics in the WebSphere Message Broker Explorer, complete the following steps.

1. In the **WebSphere MQ Explorer - Navigator** view, expand the Brokers folder.
2. To open the Resource Statistics and Resource Statistics Graph views, click **Window > Show View > Resource Statistics**. These two views are displayed together. If you close one of the views, the other view is also closed.
3. Use the information in these views to review the use of resources for which statistics are available.
The following examples demonstrate the types of question that can be answered by collecting resource statistics. This list is not exhaustive, and does include all resource types. For a full list of resource types, and the type of information that is collected for each one, see [Resource statistics data](#)

JVM statistics

How much memory is the JVM using?

Many tools that are specific to an operating system give you the total memory that is used by the execution group, but they do not show you how that memory is divided between Java™ processing and other processing in the execution group. By looking at the field `CommittedMemoryInMB`, you can see how much memory is currently allocated to the JVM. Then look at the field `MaxMemoryInMB` to see the maximum amount of memory that can be allocated.

How often is garbage collection done? Is it affecting the performance of the execution group?

To see how often the JVM is doing garbage collection, check the `CumulativeNumberOfGCCollections` field to see if the rate of collections is increasing. Garbage collection is a normal process, and is therefore expected to some degree. However, excessive garbage collection can affect performance.

To see if current garbage collection is excessive, monitor the `CumulativeGCTimeInSeconds` value. If this value is increasing by more than 2 seconds in each 20-second statistics interval, try increasing the JVM maximum heap size for your execution group by using the **`mqsichangeproperties`** command. You might also want to inspect all the Java user-defined nodes and JavaCompute nodes

that are included in your deployed message flows, to ensure that they do not create and delete many objects that could be reused; frequent deletions can contribute to excessive garbage collection.

Do I need to change the minimum or maximum heap sizes?

- If the `CumulativeGCTimeInSeconds` value is increasing by more than 2 seconds in each 20 second statistics interval, increase the maximum heap size to reduce this increase.
- If the `UsedMemoryInMB` value is never close to the `InitialMemoryInMB` value, you might have allocated more memory for the heap than is required. Therefore, reduce the JVM minimum heap size value for the execution group to a value that is closer to the `UsedMemoryInMB` value.

Change these values gradually, and check the results to find the optimum settings for your environment.

Parsers

Are message parsers using more memory than expected?

A message flow parses input messages and can create many output messages. These messages may have large bit streams or large message trees. The parsers created to perform this message processing might consume a large amount of memory. Use the Parsers statistics to determine if message flow parsers are using more memory than expected. If so, consider deploying such flows into separate execution groups or improving ESQL or Java plugin API processing to efficiently handle large messages or transformations.

Is message parsing or writing failing frequently for a particular message flow?

If a message flow receives or attempts to write an invalid message, it is likely that this will be rejected by a parser. Use the message parsers statistics to see if a message flow is rejecting a large amount of input or output messages compared with successful processing.

Outbound sockets

Are the nodes reusing outbound sockets?

Creating outbound sockets can be an expensive operation, and the number of sockets available on a computer is a finite resource. Therefore, increasing socket reuse can enhance performance. If the workload is continuous and consistent, the `TotalSockets` value indicates an initial period of activity, which then reduces when the execution group starts to reuse sockets.

A steady increase in the `TotalSockets` value over time is expected because sockets are closed after a period of inactivity, or when they have been used many times.

If the `TotalSockets` value increases significantly over time, this trend might indicate that outbound sockets are not being reused. If your message flows include HTTPRequest nodes, check that you have set the keepalive property *Enable HTTP/1.1 keepalive*. Check also whether the endpoint that is called uses keepalive sockets.

Which endpoints are most used?

The values `TotalMessages` indicates how busy each endpoint is. The value in the summary record tells you how much activity occurred across the whole execution group.

How large are sent and received messages?

The values of the `SentMessageSize_*` and `ReceivedMessageSize_*` fields give a profile of the message sizes flowing to and from each endpoint.

JDBC connection pools

Do I need to change the size of the connection pool?

If the statistics show that the count of callers waiting for connections is high, and the wait time is increasing, consider increasing the size of the pool using the `MaxConnectionPoolSize` property for the JDBCProvider configurable service. Alternatively, try reducing the number of additional instances configured for the message flow.

TCP/IPClientNodes

Are the nodes reusing outbound sockets?

Creating outbound sockets can be an expensive operation, and the number of sockets available on a computer is a finite resource. Therefore, increasing socket reuse can enhance performance. If the workload is continuous and consistent, the `TotalSockets` value indicates an initial period of activity, which then reduces when the execution group starts to reuse sockets.

A steady increase in the `TotalSockets` value over time is expected because sockets are closed after a period of inactivity, or when they have been used many times.

If the `TotalSockets` value increases significantly over time, this trend might indicate that outbound sockets are not being reused. If your message flows include HTTPRequest nodes, check that you have set the keepalive property *Enable HTTP/1.1 keepalive*. Check also whether the endpoint that is called uses keepalive sockets.

How does the information that I see in WebSphere Message Broker Explorer relate to my TCP/IP flows?

An entry is displayed for each configurable service, not for each flow.

Starting resource statistics collection in the WebSphere Message Broker Explorer

Use the WebSphere® Message Broker Explorer to start collecting resource statistics data for your execution groups. You can then view the data in the Broker Resources and Broker Resources Graph views.

Before you start:

- Ensure that you have created a message flow; for more information, see [Creating a message flow](#).
- Ensure that you have deployed a broker archive (BAR) file; for more information, see [Deploying a broker archive file](#).
- Read the concept topic, [Resource statistics](#).

Use the resource statistics data to monitor the performance and resource usage in your execution groups. You can start collecting data for active execution groups at any time.

If you prefer, you can start resource statistics collection by using the **mqsichangeresourcestats** command; for more information, see [Starting resource statistics collection](#).

To start resource statistics collection in the WebSphere Message Broker Explorer, complete the following steps.

1. In the WebSphere MQ Explorer - Navigator view, expand the Brokers folder, then expand the folder of the broker with which you are working.
2. Right-click the execution group for which you want to collect statistics. If you want statistics for several execution groups on this broker, select more than one by using standard operating system interfaces. For example, on Windows systems, hold down the **Ctrl** key and select each execution group before you right-click to open the menu.
3. Click **Statistics > Start Resource Statistics**. A message is sent to the broker to start collecting data for the selected execution groups. The property *Resource Statistics Active* is updated in the properties view of each affected execution group to indicate that data collection is now active.
A warning message is displayed in the execution group properties window to warn you that performance might be affected by your action.
4. To view statistics, click **Window > Show View > Resource Statistics** to open the Broker Resources and Broker Resources Graph views. If you are displaying statistics for this execution group for the first time, the views

might be empty until the first data is received. Update messages are sent every 20 seconds, and the views refresh automatically. The Broker Resources and Broker Resources Graph views are displayed together. If you close one of the views, the other view is also closed.

Stopping resource statistics collection

Use the **mqsichangeresourcestats** command to stop collecting resource statistics.

Before you start:

- Read the concept topic about [resource statistics](#).
- [Start integration server resource statistics collection](#).

You can stop collecting data for an active integration server at any time.

If you prefer, you can stop collecting resource statistics by using the IBM® Integration Explorer; see [Stopping resource statistics collection in the IBM Integration Explorer](#).

To stop resource statistics collection:

1. If your broker is running on Linux, UNIX, or Windows systems, set up the correct command environment. For details of how to complete this task, see [Setting up a command environment](#).
2. Decide whether you want to stop collection for a specific integration server, or for all integration servers on the broker. The way in which you started collection is not important; you can stop collection for one integration server, even if you started it for all integration servers. You can also stop collection for all integration servers, even if you started only one, or each one separately.
3. Run the **mqsichangeresourcestats** command with the appropriate parameters.

For example, to stop collecting resource statistics for the default integration server for BrokerA, enter:

```
mqsichangeresourcestats BrokerA -c inactive -e default
```

See the [mqsichangeresourcestats command](#) for further examples.

Subscribing to statistics reports

You can subscribe to topics that return statistics about the operation of your message flows and resource managers.

Message flow performance

If you enable message flow accounting and statistics collection for a broker, you can subscribe to the messages that the broker publishes on the following topic:

```
$SYS/Broker/brokerName/StatisticsAccounting/recordType/executionGroupLabel/messageFlowLabel
```

where *broker_name* is the name of the broker, *recordType* is the type of record (SnapShot or Archive), *executionGroupLabel* is the name of the execution group that you created on that broker, and *messageFlowLabel* is the name of the message flow that you deployed to the execution group.

These messages contain statistics reports and are published at a regular interval, which you control by setting the `statsInterval` property of the broker. Each publication is a JMS TextMessage that contains the statistics report in XML format.

Note: If you need to revert to using a JMS BytesMessage format, this can be achieved by setting the environment variable **MQSI_STATS_MQSTR=false**.

Resource performance

If you enable resource statistics collection for one or more execution groups on a broker, you can subscribe to the messages that the broker publishes at 20-second intervals on the following topic:

```
$SYS/Broker/brokerName/ResourceStatistics/executionGroupLabel
```

For more information about how to interpret the resource statistics that are included in the publication, see [Viewing resource statistics data in the WebSphere Message Broker Explorer](#).

Using wildcards in subscriptions

You can use wild cards when you subscribe to statistics reports. For example, to receive message flow statistics reports for all brokers and all execution groups, subscribe to the following topic:

```
$SYS/Broker/+/StatisticsAccounting/
```

To receive execution group resource statistics reports for all brokers and all execution groups, subscribe to the following topic:

»

```
$SYS/Broker/+/ResourceStatistics/
```

« For further details about how you can use wildcards, see [Special characters in topics](#).

Subscribers receive statistics reports only from those brokers that are enabled to produce statistics.

Try a Dynamic Guide to help with troubleshooting Message Broker:

https://www.ibm.com/developerworks/community/blogs/c7e1448b-9651-456c-9924-f78bec90d2c2/entry/try_a_dynamic_guide_to_help_with_troubleshooting_message_broker?lang=en

https://www.ibm.com/developerworks/community/blogs/c7e1448b-9651-456c-9924-f78bec90d2c2/resource/pdf/flowchart/PD_Flowchart.html?lang=en#/home

Recovering after the broker fails:

Check what recovery procedures are available, according to what has failed.

Try to get to the root of the problem first, using the diagnosis techniques described throughout the [Troubleshooting and support](#) section of the information center. If your problem cannot be resolved using these techniques, contact your IBM® Support Center. Use the procedure in this section only as a last resort.

If you are able to recover, and can start the broker again, the broker attempts to recover and re-establish all sessions with CMP applications that were active at the time of failure. CMP applications include the WebSphere® Message Broker Explorer, the WebSphere Message Broker Toolkit, and applications that you have written to this API.

If you cannot correct the current problem by using problem determination, complete the following sequence of operations to re-create the broker:

1. Stop the broker by using the **mqsistop** command.
2. Stop the broker queue manager by using the **endmqm** command.
3. Delete the broker by using the **mqsdeletebroker** command.
4. Re-create the broker by using the **mqsicreatebroker** command.
5. Start the broker by using the **mqsistart** command.
6. Redeploy all resources to the broker.

Recovering after the broker fails

Check what recovery procedures are available, according to what has failed.

Try to get to the root of the problem first, using the diagnosis techniques described throughout the [Troubleshooting and support](#) section of the information center. If your problem cannot be resolved using these techniques, contact your IBM® Support Center. Use the procedure in this section only as a last resort.

If you are able to recover, and can start the broker again, the broker attempts to recover and re-establish all sessions with CMP applications that were active at the time of failure. CMP applications include the WebSphere® Message Broker Explorer, the WebSphere Message Broker Toolkit, and applications that you have written to this API.

If you cannot correct the current problem by using problem determination, complete the following sequence of operations to re-create the broker:

1. Stop the broker by using the **mqsistop** command.
2. Stop the broker queue manager by using the **endmqm** command.
3. Delete the broker by using the **mqsdeletebroker** command.
4. Re-create the broker by using the **mqsicreatebroker** command.
5. Start the broker by using the **mqsistart** command.
6. Redeploy all resources to the broker.

Recovering after the broker queue manager fails

Check what recovery procedures are available, according to what has failed.

Try to get to the root of the problem first, by using the diagnosis techniques described throughout the [Troubleshooting and support](#) section of the information center. If your problem cannot be resolved by using these techniques, contact your IBM® Support Center. Use the procedure in this section only as a last resort.

If the broker's queue manager fails and cannot be corrected by using problem determination, or by the IBM Support Center, perform the following sequence of operations to re-create the queue manager:

1. Ensure that no WebSphere® Message Broker Toolkit users are deploying to the broker. You must wait until any such actions have completed.
2. Stop the broker by using the **mqsistop** command.
3. Delete the broker by using the **mqsdeletebroker** command, with the **-q** parameter to remove the queue manager.

4. Re-create the broker by using the **mqsicreatebroker** command.
The **mqsicreatebroker** command creates the queue manager and default queues automatically.
5. Re-create any specific queues that are needed for your message flows.
6. Start your brokers by using the **mqsistart** command.
7. Redeploy all resources to the broker to ensure that its configuration is consistent.

IBM Integration Bus and WebSphere Message Broker Frequently asked questions (FAQ)

Question

The following is a list of answers to frequently asked questions (FAQ) about IBM Integration Bus (IIB) and WebSphere Message Broker (WMB) for new and experienced users.

Answer

1. What are the software requirements for IIB/WMB?

The complete product software requirements and pre-requisites are available on the [System Requirements Page](#).

2. Where can I find more information about IIB/WMB?

- IIB/WMB product documentation can be found in the Library which provides product Information Centers for each of the current releases. See the [IBM Integration Bus Library](#) page or the [WMB Product Library page](#). Each Information Center can be viewed online or through a downloadable version.
-
- There are also a number of IBM Redbooks available which provide detailed information about IIB/WMB. See the [IBM Redbooks](#) page.
-
- IBM developerWorks also has a number of papers published that provide detailed information about specific product functionality. See the [IBM developerWorks](#) page

3. What are the main components of IIB/WMB?

The 3 main components in IIB/WMB are Runtime, Development and Administration.

- **Runtime** - The broker runtime is the main processing component to which development artifacts are deployed.
- **Development** - The Toolkit is used for development of message flows and other development artifacts.
- **Administration** - Message Broker Explorer is an administration tool used to perform deploys, create brokers or new execution groups, and enable flow monitoring. WMB V8 provides a WebUI tool to view the broker runtime or resources deployed in a browser.

4. Does IIB/WMB require a database to function?

From Message Broker V7 onwards, there is no database pre-requisite, the product does not require a

database for any internal function. You may have database nodes or use databases for other function and so Message Broker does provide functionality to interface with application databases.

5. How do I install IIB/WMB?

IIB/WMB installation comprises of installing a Broker component, the Toolkit and Message Broker explorer. On Windows all three components can be installed using Windows Launchpad that is provided with the product. On Unix platforms, installation can be performed through an installation wizard or in silent mode. On z/OS, the installation is performed using SMP/E. Detailed information about installation is provided in the online Information Center

- [IIB Installing and Uninstalling](#)
- [WMB Installing and Uninstalling.](#)

6. Getting started with IIB/WMB. Is there a default configuration of IIB/WMB that I can use?

Yes. IIB/WMB provides a default configuration. This can be started through Toolkit Welcome screen. The default configuration wizard creates a broker and all the resources required.

7. Where can I find training and education on setting up and using the IIB/WMB Product?

The IBM Education Assistant site has a number of education modules across multiple Message Broker releases. See the [IBM Education Assistant: WebSphere Message Broker page](#)

IBM Education also offers courses on IIB/WMB. See this [list of IIB/WMB courses being offered](#).

8. Where can I find samples for IIB/WMB?

A comprehensive set of samples are provided with IIB/WMB. These samples can be found on the Toolkit Welcome page. The samples cover basic concepts as well as advanced areas like WebServices File processing, Monitoring etc. See the [complete list of samples available](#).

9. Are there IBM forums or user groups available for IIB/WMB users?

The following forums and user groups are available for Message Broker.

1. [mqseries.net forum](#) - This forum is active with participation by the Message Broker user community.
2. [IBM developerWorks forum](#)

10. How do I create a broker?

A broker can be created using Message Broker explorer or from the command line. Specific information on creating a broker is available in the online Information Center under the topic Configuring brokers for test and production > [Configuring brokers](#)
[Creating a broker](#)

11. What are the major processes for a running broker?

Once a broker is started, the following processes get started and they can be seen in Task Manager (on Windows) and/or ps listing (on UNIX's). On z/OS these processes run under USS.

- bipservice - This is a bootstrap process that starts the broker runtime
- bipbroker - This main administration that manages the broker runtime.
- biphttplistener - This is the http server process that provides HTTP and SOAP functionality in Message Broker
- DataFlowEngine - This is the Execution Group process which runs message flow threads. Each message flow runs in this process as a thread.

12. Are there any monitoring tools available in IIB/WMB?

Business level monitoring can be configured in IIB/WMB message flows that enables the different nodes in the message flow to emit events when processing messages. These events can be further fed to other applications such as WebSphere Business Monitor for transaction monitoring, transaction auditing, and business process monitoring. More information can be found in the online Information Center under the topic Performance and monitoring > [Business-level monitoring](#)

13. What are the High availability options available in IIB/WMB?

Message Broker can be set up to work with the following High Availability options:

- Multi-instance mode using multi-instance queue manager
- Existing High Availability manager/Windows cluster such as HACMP or MSCS cluster

Please see the following page for more information on [High availability](#).

14. What are the backup and recovery procedures for IIB/WMB?

The details may be found in the online Information Center under the topic Administering existing brokers > [Backing up resources](#)

15. Are there any existing message flow patterns/templates that I can use to design my message flows?

Built-in patterns are supplied with the IIB/WMB Toolkit to create resources that are used to solve specific business problems. You can also create user-defined patterns by using the pattern creation templates provided with IIB/WMB Toolkit.

Patterns are available to a pattern user in the Patterns Explorer view in the Broker Application Development perspective of the IIB/WMB Toolkit.

16. How do I migrate from a previous version to the newer version?

You can migrate in two different ways:

- Import all the development artifacts into the Message Broker Toolkit which will migrate them as necessary and deploy them to the new version runtime.
- Migrate only the runtime by using the `mqsimigratecomponents` command
For example, `mqsimigratecomponents <broker> -s <source version> -t <target version>`

The details may be found in the online Information Center under the topic Reference > Configuration and administration > Commands > Runtime commands > [mqsimigratecomponents command](#)

17. Where can I find the latest Fix Pack for IIB/WMB?

A Fix Pack may be downloaded from the following page [Recommended fixes for WebSphere Message Broker](#)

18. When is the next Fix Pack available?

See the [WebSphere Message Broker planned Fix Pack release schedule](#)

19. Where are the security features provided in IIB/WMB?

IIB/WMB provides a number of security features. Detailed information about all the security features available in IIB/WMB can be found in the online Information Center under the topic [Security overview](#)

20. Can I receive updates about IIB/WMB?

Yes. To subscribe to the quarterly IIB/WMB Newsletter, send an email to wsehelp@us.ibm.com with the subject line: **subscribe IBM Integration Bus and WebSphere Message Broker**
More information on how to subscribe is available at [Knowledge Collection: IBM Support Newsletters](#).

To see the most requested documents for the past quarter as well as those identified as valuable in helping answer your questions related to IIB/WMB see: [Featured documents for IBM WebSphere Message Broker](#)

Backing up resources

Back up your broker components, and the working files associated with brokers, the IBM® Integration Explorer, and the IBM Integration Toolkit, so that you can restore these resources if required.

Back up your broker components regularly to ensure that you can return to a known operational state if necessary. In addition to configuration and operation state, the broker maintains additional resources in its work path, and you can request that these resources are also backed up.

Back up the workspaces and connection files that you have created in the IBM Integration Explorer.

Back up the workspaces you have created in the IBM Integration Toolkit; these resources contain your application development resources; for example message flows and message model schema files. If you use a development repository to store application resources, such as Rational® ClearCase®, see the documentation associated with that repository to check how you can back up this data.

The following topics tell you how to back up and restore brokers and the IBM Integration Toolkit workspace:

- [Backing up the broker](#)
- [Restoring the broker](#)
- [Backing up the IBM Integration Explorer and IBM Integration Toolkit workspace](#)

On distributed systems, you can use the backed up components to restore the broker only on an identical operating environment. The operating system must be at the same level, and the broker and queue manager names must be identical.

Backing up the broker

Back up the broker configuration and all associated resources.




Before you start: [Create the broker](#).


You can back up the broker and its resources to preserve the current state of the broker configuration. You can use the backup file that is created to restore a broker in an identical operating environment: the operating system must be at the same level, and the broker and queue manager names must be identical.

You can run this command for a broker that is active. However, you must not take a backup while the broker is processing configuration changes and deployments; the backup file created might contain

incomplete information. If the file contains partial records, you cannot use it to restore the broker at a later time.

To ensure that the backup is complete and correct, take a backup either when the broker is not processing a configuration change (such as a deployment or change property) or when the broker is stopped.

1. If you want to back up an active broker, check that no configuration change requests are in progress. For example, if you are changing broker properties, or have initiated a deployment, wait for these actions to complete before you back up the broker. Active message flows are unaffected by the backup process.
If you prefer, you can stop the broker before you take a backup by using the **mqsistop** command.
2. Back up the broker. Specify the broker name and the location in which the backup file is created. You can also optionally specify the name of the backup file, and the name of a file to which a detailed trace is written.
 -    Run the **mqsibackupbroker** command, specifying the broker name and the directory to which the backup file is written.
For example, to back up a broker on Windows, enter the following command:

```
mqsibackupbroker IB9NODE -d c:\MQSI\BACKUP
```
 -  Customize and submit the JCL member `BIPBUBK`.
3. When the command has completed successfully, you can continue to use the broker. If you stopped the broker, restart it by using the **mqsistart** command.

The current broker configuration is saved in the backup file. Keep the file safe so that you can restore the broker at a later date if required.

Restoring the broker

Restore a broker configuration that you backed up previously.

Before you start: [Back up the broker.](#)

You can restore a broker on a computer that has an identical configuration by using the backup file that you created. The operating system must be at the same level, and the broker and queue manager names must be identical.

1. If you have deleted the broker and it no longer exists, or if you are restoring it on a different computer, create it by using the **mqsicreatebroker** command. Use the same name and parameters that you used for the broker that you backed up, including the name of the queue manager.
2. If the broker is running, stop it by using the **mqsistop** command. If you intend to restore common configuration information from other brokers


that you have configured on this computer, you must also stop all the brokers that share this common information. For example, you can restore profile information from common files.

3. Restore the broker. Specify the broker name and the name and location of the backup file. If you want to restore common configuration information, or if you want a trace of the actions taken, specify the appropriate parameters for your platform.

-    Run the **mqsirestorebroker** command.

For example, to restore a broker on Windows, enter the following command:

```
mqsirestorebroker WBRK_BROKER -d c:\MQSI\BACKUP -a mybroker.zip
```

-  Customize and submit the JCL member BIPRSBK.



4. When the command has completed successfully, start the broker by using the **mqsistart** command.

Next: The broker configuration has been restored; you can continue your work with this broker.

Backing up the IBM Integration Explorer and IBM Integration Toolkitworkspace



The IBM® Integration Explorer and IBM Integration Toolkit workspaces contain your personal settings and data, such as message flow and message set resources. You can have multiple workspaces in different locations, and you can also have references to projects that are in other locations, therefore consider all these locations when you back up your resources.

The default workspace directory for the IBM Integration Explorer depends on the platform on which it is running:

-  On Windows Server 2008, the default workspace directory is created at C:\Users\user_ID\IBM\MQ Explorer\<server project>\.
-  On Linux, the default workspace directory is created at /home/user_ID/IBM/MQ Explorer\<server project>/.

where *user_ID* is the user name with which you are logged on. Back up files in these locations, and in all other locations in which you have saved workspace files.

The default workspace directory for the IBM Integration Toolkit depends on the platform on which it is running:

-  On Windows Server 2008: C:\Users\user_ID\IBM\IntegrationToolkit90\workspace.
-  On Linux: /home/user_ID/IBM/IntegrationToolkit90/workspac e.

where *user_ID* is the user name with which you are logged on. Back up files in these locations, and in all other locations in which you have saved workspace files.

The IBM Integration Toolkit workspace directory contains a directory called `.metadata`, which contains your personal settings and preferences for the IBM Integration Toolkit. If the `.metadata` directory gets corrupted, you lose these settings, and the IBM Integration Toolkit reverts to the default layout and preferences. If you have not backed up the `.metadata` directory, you must manually set all preferences again, and import all projects, such as message flow projects, that were displayed in the Application Development view. To back up the `.metadata` directory, take a copy of the directory.

The IBM Integration Toolkit workspace also contains a directory for each project (for example, a message flow project) that you have created in the IBM Integration Toolkit. These directories contain your data, which you must back up.

Use one of the following methods to back up the data in your workspace; the instructions are the same for both the IBM Integration Explorer and the IBM Integration Toolkit.

- Export your working projects. You can export the projects directly as a compressed file. For further information, see [Exporting](#) in the Eclipse Workbench User Guide.
- Copy the project directories from the workspace directory to another location.
- IBM Integration Explorer only: Export all `.broker` connection files to save the details of connections to all your brokers.
- IBM Integration Toolkit only: Take copies of all your BAR files to back up their contents. Include all the associated source files when you build your broker archive (BAR) files; you can then save all content by saving only the BAR file. To add resources to a BAR file that is ready for deployment, select **Include source files**, which adds the message flow and message set source files, and the compiled files.

If you want to restore the resources, copy the directories back into your workspace directory and import the projects. For instructions, see [Importing](#) in the Eclipse Workbench User Guide.

Index :

Tab navigation

- [Collecting documentation for Execution Group problems](#)
- [Technotes](#)

- [Webcast Replays](#)
- [developerWorks Articles](#)
- [Blogs](#)
- [Publications](#)
- [Knowledge Collections for IBM Integration Bus and WebSphere Message Broker](#)

Collecting documentation for Execution Group problems

1. [Collect troubleshooting data for an Execution Group problem in WebSphere Message Broker for UNIX and Linux. \(1299777\)](#)

You are experiencing an Execution Group problem in WebSphere Message Broker for UNIX and Linux.

2. [Collect troubleshooting data for an Execution Group problem in WebSphere Message Broker for Windows. \(1299297\)](#)

You are experiencing an Execution Group problem in WebSphere Message Broker for Windows.

3. [Collect troubleshooting data for an Execution Group problem in WebSphere Message Broker for z/OS. \(1302950\)](#)

You are experiencing an Execution Group problem in WebSphere Message Broker for z/OS.

4. [Procedure to take a user or service trace of message flow at an execution group level \(1177321\)](#)

Use this procedure to capture a user or service level trace of a message flow at an execution group level.

5. [Questions on capturing garbage collection data in WebSphere Message Broker \(1444198\)](#)

Use this procedure to capture garbage collection data for execution groups in WebSphere Message Broker.

[Back to top](#)

1. [Execution Group abends showing WTX libraries following WMB upgrade\(1575522\)](#)

After you upgrade to WebSphere Message Broker (WMB) 7.0.0.3, message flows containing WTX nodes cause their Execution Groups to abend.

2. [How the WMB AggregateReply node builds compound messages\(1621833\)](#)

How are compound messages built by the AggregateReply node in WebSphere Message Broker (WMB)?

3. [Instructions for manually deleting an execution group \(1641696\)](#)

How can I manually delete an execution group that is currently stopped?

4. ['Execution Group' view does not display deployed message flows in Message Broker Explorer \(1444279\)](#)

WMB v7 deploy seems to have completed successfully, but the message flow is not listed under the Execution Group when viewed in the Message Broker Explorer.

5. [Java exceptions during message processing \(1609265\)](#)

When processing messages through a WMB message flow containing Java™ nodes, you may encounter an error saying 'Exception in thread "main" java.lang.UnsatisfiedLinkError: awt'.

6. [How does 'mqsisstopmsgflow' command work in a WMB Execution Group? \(1586129\)](#)

Does the WMB 'mqsisstopmsgflow' command stop message flows sequentially or concurrently?

7. [MbXSDInternalError on BAR file deploy \(1588946\)](#)

You are unable to deploy a BAR file to your execution group in WebSphere Message Broker. Errors BIP2087E, BIP5024E, and BIP4395E can be seen in the deploy logs within the Toolkit and the Windows Application Event logs following the failed deploy.

8. [BIP2853E and BIP2865E Problems creating security objects for new execution group \(1500564\)](#)

You are attempting to create a new Execution Group on WMB 7.0 on z/OS. WMB

Administration Security is enabled, and you are receiving warnings and errors.

9. [Files created in \\$TEMP on broker startup \(1431072\)](#)

What are the 0 byte files created in \$TEMP/.com_ibm_tools_attach directory upon broker startup?

10. [WebSphere Message Broker FileInput node message handling \(1503465\)](#)

What is the maximum file size that a WebSphere Message Broker FileInput node can process?

If you did not find what you were looking for in these technotes then you can [View all technotes related to WebSphere Message Broker Execution Group.](#)

[Back to top](#)

Webcast Replays

1. [Ask the Experts Replay: WebSphere Message Broker - Message Flow Design Strategies And Usual Pitfalls](#)

This WebSphere Support Technical Exchange Ask the Experts session addresses general WebSphere Message Broker V7 message flow design strategies and usual pitfalls that one can get into in the areas of design, implementation, configuration and testing.

2. [Webcast replay: WebSphere Message Broker V7 - Administration and Troubleshooting Made Easy](#)

WebSphere Message Broker V7.0 brings in various simplicity features and fewer dependencies. This WebSphere Support Technical Exchange talks about the key changes in broker architecture as a result of this, and examines how the broker environment is a lot easier to manage using the new administration tool - MB Explorer.

3. [Webcast replay: WebSphere Message Broker - Troubleshooting and Common Problems\(7006694\)](#)

This WebSphere Support Technical Exchange presents several methods of troubleshooting functionality available in WebSphere Message Broker and will discuss how to resolve common problems..

4. [Webcast replay: Troubleshooting WebSphere Message Broker - Hints and Tips \(7007638\)](#)

This WebSphere Support Technical Exchange discusses troubleshooting functionality available in WebSphere Message Broker.

5. [Webcast Replay: WebSphere Message Broker Top Ten Problems: A Level 2 Perspective \(7007462\)](#)

This WebSphere Support Technical Exchange is designed to address the top ten problems reported to IBM support, so you can avoid these pitfalls.

[Back to top](#)

developerWorks Articles

1. [Introduction to the WebSphere Message Broker global cache](#)

This article describes the global cache, shows you how to use it, and answers some frequently asked questions about it.

2. [Automated SNMP Alerts in WebSphere Message Broker](#)

This article shows you how to enhance message flow monitoring to automatically send Simple Network Management Protocol (SNMP) alerts to an administrative console in order to streamline your organization's error notification and error response processes.

3. [Enhancing flexibility of Websphere Message Broker applications by deploying application resources as source files](#)

This article shows you how to use Toolkit Fix Pack 1 to construct and configure message flows using deployable subflows, ESQL modules, and maps without compiling a flow into a compiled message flow (CMF) format, and how to modify and assemble Message Broker artifacts on a server without a Toolkit.

4. [Developing a WS-ReliableMessaging solution using WebSphere Message Broker V8](#)

This article shows you how to configure WS-RM in a solution where a JAX-WS web service client and a Microsoft .NET client send SOAP/HTTP messages to Message Broker.

5. [Authoring message flow patterns using WebSphere Message Broker APIs](#)

This article shows you how to author WebSphere Message Broker flow patterns using the Message Broker Java API and Java programming.

[Other IBM Integration Bus or WebSphere Message Broker articles on developerWorks](#)

[Back to top](#)

Blogs

1. [Execution Group Specific ENVFILES](#)

EG specific ENVFILES allow one or more EGs to have their own unique runtime environment.

2. [Changes in broker startup trace in WMB V8](#)

This article describes changes in the startup trace that were introduced in WMB V8.

3. [How to handle a hung flow](#)

This article describes how to redeploy a corrected flow to replace the old one.